# A UNIFIED MODELING FRAMEWORK FOR INTERNET OF THINGS (IOT) SYSTEMS

KHURRUM MUSTAFA ABBASI
01-281152-002

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Software Engineering)

Department of Software Engineering

BAHRIA UNIVERSITY ISLAMABAD

JULY 2023

# APPROVAL FOR EXAMINATION

Scholar's Name: **Khurrum Mustafa Abbasi** Registration No. **34233**

Program of Study: **PhD Software Engineering**

Thesis Title: **A Unified Modeling Framework for Internet of Things (IoT) Systems**

It is to certify that the above scholar's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for examination. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index _____15______% that is within the permissible limit set by the HEC for the PhD degree thesis. I have also found the thesis in a format recognized by the BU for the PhD thesis.

**Principal Supervisor's Signature: ____________________________________________**

**Date: _________________________**

**Name: Dr. Tamim Ahmed Khan**

# AUTHOR'S DECLARATION

I, **Khurrum Mustafa Abbasi** hereby state that my PhD thesis titled "A Unified Modeling Framework for Internet of Things (IoT) Systems" is my own work and has not been submitted previously by me for taking any degree from this university **Bahria University Islamabad** or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw/cancel my PhD degree.

Name of scholar: **Khurrum Mustafa Abbasi**
Date: July 21, 2023

# PLAGIARISM UNDERTAKING

I, solemnly declare that research work presented in the thesis titled "**A Unified Modeling Framework for Internet of Things (IoT) Systems**" is solely my research work with no significant contribution from any other person. Small contribution / help wherever taken has been duly acknowledged and that complete thesis has been written by me. I understand the zero-tolerance policy of the HEC and Bahria University towards plagiarism. Therefore, I as an Author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred / cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of PhD degree, the university reserves the right to withdraw / revoke my PhD degree and that HEC and the University has the right to publish my name on the HEC / University website on which names of scholars are placed who submitted plagiarized thesis.

Scholar / Author's Sign: __________________

Name of the Scholar: Khurrum Mustafa Abbasi

# DEDICATION

This thesis is dedicated to my family. My father "Ghulam Mustafa Khan" who always provided me moral and financial support. My mother always supported and praised for me. My siblings always encouraged me, my wife who supported me in all the times after marriage and the one who is newly added to our family, that is my daughter "Ayesha".

## ACKNOWLEDGMENTS

I would like to express my thanks to Almighty Allah without whose blessings I have never been able to achieve this goal in life. He always showered countless blessings on me. A Supervisor and role model, Prof. Dr. Tamim Ahmed Khan, for his guidance, investing plenty of time and efforts in the completion of this thesis. My Co-Supervisor, Dr. Irfan ul Haq, for always encouraging and providing novel ideas for solving different problems related to this research. Dr. Awais Majeed, the Head of Department supported during all the processes and, taught multiple courses as well.

I am always thankful to my grandfather Nazar Muhammad Khan (passed away in 2009) and maternal grandfather Abdul Hameed Khan (passed away in 2008), who are no longer among us. Both played a key part in teaching me how to deal with tough times and put maximum efforts to achieve goals in life. My grandmother Farida Begum for always praising for my success.

I acknowledge Mr. Atique-ur-Rehman Abbasi (Cousin of mine and a friend), who supported me in content improvement from language aspect. My relatives' uncles, aunts and cousins always encouraged me to achieve this goal. Finally, the Open-Source community, for free availability of different software.

# ABSTRACT

The Internet of Things (IoT) is an emerging field, and the growth of IoT devices is a rapidly increasing trend. These devices are expected to combine both heterogeneity and smartness, enabling them to provide services for both humans and other devices. Due to this, it is crucial to have an appropriate modeling framework in place for modeling the integration of these devices. Modeling IoT systems requires the consideration of several aspects, including the selection of discrete or continuous mathematical models, computational simulation, or a combination of these. The selection of modeling approaches or frameworks is also crucial in this process. In this research, provide a unified modeling framework by integrating multiple frameworks and approaches. Our research began with exploring, deducing, and inducting the research problem. We then built a hypothesis that a unified framework can be developed for modeling complex IoT systems. To achieve this, we first formulated sub-frameworks based on the architectural components of IoT systems. These sub-frameworks were then integrated into a unified framework, providing a way to model the behavior of service-oriented internet-based devices and systems in complex scenarios. The unified framework is based on three distinct sub-frameworks for modeling IoT systems. The first sub-framework is aimed at modeling IoT systems from the Software Engineering viewpoint. The second sub-framework is focused on modeling IoT systems that have fuzzy values, such as values that fall between 0 and 1, giving rise to fuzzy logic. The third sub-framework is aimed at modeling IoT systems that have ambient entities in their composition. An ambient entity is an entity that must possess the properties of mobility, inclusion, and narrowness, such as a bus or an airplane. We used our framework to model case studies and compared our framework with the exiting. This study is expected to make a significant contribution to the modeling paradigm employed in understanding, analyzing, and designing IoT systems. With the proposed unified framework, it is possible to model the behavior of IoT systems in a comprehensive manner, considering the different perspectives and aspects that are involved in the modeling process. This will enable the development of more robust and reliable IoT systems, which can better serve the needs of both humans and other devices.

# TABLE OF CONTENTS

**CHAPTER** **TITLE** **PAGE**

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| SDN | Software Defined Networks |
| 2D | Two Dimensional |
| 3D | Three Dimensional |
| ABM | Agent-based Modeling |
| AC | Air Conditioner |
| AGG | Attributed Graph Grammar |
| AOM | Aspect-oriented Modeling |
| AOP | Aspect-oriented Programming |
| BDI | Belief Desire Intention |
| BRTS | Bus Rapid Transit System |
| CBM | Contract-based Modeling |
| CP | Collecting Person |
| CV | Collecting Vehicle |
| DES | Discrete Event Simulation |
| FLM | Fuzzy-logic Modeling |
| GIS | Global Information System |
| HEC | Higher Education Commission |
| IIRA | Industrial Internet of Things Reference Architecture |
| IoT | Internet of Things |
| IoT-A, | Internet of Things Architecture |
| LED | Light Emitting Device |
| LHS | Left Hand Side |
| ML | Machine Learning |
| MOU | Memorandum of Understanding |
| NBM | Network-based Modeling |
| OBM | Object-based Modeling |
| OHD | Over Head Display |
| OOM | Object-oriented Modeling |
| OOP | Object-oriented Programming |
| OWL | Web Ontology Language |
| RAMI | Reference Architecture Model Industry |
| RHS | Right Hand Side |
| SDLC | Software Development Life cycle |
| SE | Software Engineering |

| | |
|---|---|
| SOAO | Service-oriented Architecture Ontology |
| SOAP | Simple Object Access Protocol |
| SOAML | Service Oriented Architecture Modeling Language |
| SOM | Service-oriented Modeling |
| SORA | Service-oriented Reference Architecture |
| SORM | Service-oriented Reference Model |
| SoTAP | Service-oriented Things Access Protocol |
| TSDL | Things Service Description Language |
| TV | Television |
| UDDI | Universal Data Discovery and Integration |
| WS | Web Service |
| WSA | Web Service Architecture |
| WSDL | Web Service Description Language |
| WSMO | Web Service Modeling Ontology |
| XML | Extensible Markup Language |
| XSD | XML Schema Definition |
| XSLT | Extensible Stylesheet Language Transformations |

# LIST OF SYMBOLS

| Symbol | Description |
|---|---|
| ρ | Set of properties |
| $\lambda$ | Location |
| $\iota$ | Identification |
| $\zeta$ | Autonomicity |
| $\mu$ | Inclusion |
| $\delta$ | Mobility |
| $\varepsilon$ | Granularity |
| $\eta$ | Cognition |
| $\kappa$ | Flexibility |
| $A$ ( Alpha ) | Meta agent |
| $B$ ( Beta ) | Cognitive agent |
| Γ ( Gamma ) | Mobile agent |
| $E$ ( Epsilon ) | Static agent |
| $Z$ ( Zeta ) | Proto agent |
| $H$ ( Eta ) | Connecting agent |
| $O\rho$ | Optional Properties |
| $M\rho$ | Mandatory Properties |
| & ( ampersand ) | And |
| ^ ( caret ) | And |
| ∨ ( reversed caret ) | Or |
| ⇒ | Implies |
| ⇔ | Equivalent |

| | |
|---|---|
| $\forall$ | For all |
| $\exists$ | There exists |
| $\nexists$ | There does not exist |
| $\in$ | Belongs to |
| $\notin$ | Does not belongs to |
| $\mid$ | Such that |
| $\emptyset$ | Null set or empty set |
| $\cup$ | Union of two sets |
| $\cap$ | Intersection of two sets |

# Chapter 1

# INTRODUCTION

The term Internet of Things (IoT), first introduced by Kevin Ashton in 1998, refers to an emerging paradigm that consists in new Internet-based information service architecture [1]. The Internet of Things (IoT) is a rapidly evolving paradigm that encompasses a new Internet-based information service architecture. The term "things" in IoT refers to objects and devices that are interconnected to form much larger systems, enabling new forms of ubiquitous and pervasive computing. IoT has introduced a new dimension to technology, shifting the focus from human interaction to machine-to-machine communication over the internet. Modeling approaches for IoT are used to better understand the behavior of these systems and to improve their design, performance, and security.

Modeling approaches for IoT can include agent-based modeling, event-based modeling, network-based modeling, data-driven modeling, and hybrid modeling. These approaches can be used to simulate the behavior of IoT systems and to evaluate different design options. For example, agent-based modeling can be used to simulate the interactions between devices in an IoT system, while network-based modeling can be used to evaluate the performance of different communication protocols. However, the widespread adoption of IoT also brings new challenges, such as data privacy and security, interoperability, scalability, and management. Security concerns are of paramount importance in IoT as it involves the collection and transmission of sensitive personal and business data. Ensuring the security of IoT systems is crucial to protect against cyber-attacks and data breaches. Modeling approaches can also be used to evaluate the security of IoT systems and to identify vulnerabilities that need to be addressed. The contract-based reasoning approach in the design and development of IoT systems is a powerful technique that enforces several key principles. These principles, including component substitutivity and reuse, incremental development through successive refinements, and independent Implement-ability, provide a solid foundation for the component-based development process. Besides being a solution for system design, contract-based reasoning offers diverse opportunities: mapping and tracing requirements to components, tracking the evolution of requirements during development, reviewing models, virtual integration of components [2], and, most importantly, compositional verification. Instead of reasoning with implementations during formal verification of requirements, one can use contracts and split the verification in two steps: (1) verify that each component satisfies its contract and (2) verify that the network of contracts correctly assembles and satisfies the requirement. Even though the number of relations that need to be verified in order for contract-based reasoning to work is multiplied (linearly with the number of components), in general they involve more abstract specifications and thus they are less prone to combinatorial explosion, which makes them more tractable by automatic verification tools.

This approach offers a formal framework for demonstrating the satisfaction of requirements and provides several benefits to the system design process. Contract-based reasoning enables the mapping and tracing of requirements to components, facilitates the tracking of requirement evolution throughout the development cycle, enables model reviews, supports the virtual integration of components, and most importantly, provides compositional verification. Rather than verifying requirements through the implementation, contract-based reasoning splits the verification process into two steps. First, it verifies that each component satisfies its own contract, and then it verifies that the network of contracts correctly assembles and satisfies the overall requirement. The verification process is multiplied linearly with the number of components, but these relations typically involve more abstract specifications and are therefore less prone to combinatorial explosion, making the process more manageable for automatic verification tools.

The use of models in software engineering is a crucial aspect of the development process, serving various purposes. The first type of modeling, known as domain modeling, focuses on providing a clear understanding of the context and major concepts of a specific problem. Domain models aim to represent the properties of a domain, such as business processes, by capturing real-world information. Specification modeling, on the other hand, is used to formally represent requirements in a clear and concise manner. This type of modeling is used to describe the desired features and functionalities of the software system. Design modeling, also known as implementation modeling, takes the specifications provided by the specification modeling and converts them into a working solution. This type of modeling provides a control flow and is used to represent the behavior of different parts of the system, allocate responsibilities, and represent the modularization, interaction, and composition of the system. Different perspectives of modeling exist in software engineering, each with its own unique approach and goal. The selection of the appropriate model and perspective depends on the specific requirements and goals of the software development project.

The development of IoT systems involves multiple phases that require careful consideration of various aspects such as infrastructure design, security, embedded systems architecture, service provisioning and management, integration with emerging technologies such as Fog computing, Software Defined Networking (SDN), Blockchain, and the modeling and management of IoT applications. To assist in this process, researchers have proposed various reference architectures that outline the different components involved in IoT systems and their relationships. One of the well-known reference architectures for generic IoT systems is IoT-A, but there are also domain-specific reference architectures, such as the Industrial IoT Reference Architecture (IIRA) and the Reference Architectural Model Industrie 4.0 (RAMI 4.0). These reference architectures provide a holistic view of the components involved in IoT systems, from low-level hardware components to higher-level software components, and can help organizations in designing and deploying robust IoT solutions. In addition, the reference architectures also consider the various constraints and requirements specific to IoT systems, such as low-power requirements, real-time communication, and large-scale deployment. Figure 1.1 *Functional Model of Reference Architecture of IoT* [225] shows functional model of the IoT-A reference architecture for Internet of Things.

## 1.1 Motivation

The field of IoT is currently a highly active area of research. Despite the increasing number of IoT devices and the potential benefits they offer, there is still a lack of a comprehensive IoT model that can address the challenges and complexities of IoT systems. One of the key challenges is the need for an environment that can seamlessly integrate different IoT devices and enable them to communicate and share services with each other.

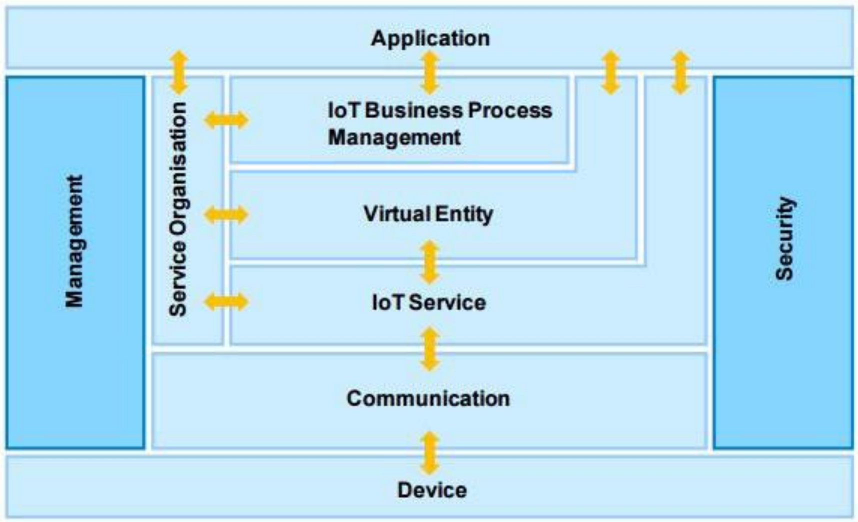


Figure 1.1 Functional Model of Reference Architecture of IoT

To address these challenges, there is a need for a system that can ensure trust and security between IoT devices. This can be achieved by establishing contracts between devices, which can be easily managed and monitored. The system also can identify devices and maintain records of their account status and credibility information. This information can be used to make informed decisions and enforce consequences for any device that violates the established contracts.

To illustrate the potential of such a system, consider the hypothetical scenario of universities in the twin cities of Rawalpindi and Islamabad in Pakistan. There are 26 HEC recognized universities in these cities, with 21 in Islamabad and 5 in Rawalpindi. In this scenario, all universities are considered "smart" and equipped with advanced systems, such as an emergency response system, smart parking, smart waste management, smart pollution control, and smart classrooms.

For example, consider the scenario of an international conference to be held at the Air

University in Islamabad. Given the expected influx of 200 additional vehicles for the event, the university can only accommodate 120 additional vehicles. To deal with this, the university looks towards its neighboring universities, such as Bahria University and National Defense University (NDU), for help. Bahria University has a free space for 230 vehicles, which the Air University hires to accommodate the surplus during the event. The host university also needs additional garbage collectors to manage the waste generated by the influx of visitors and accompanying refreshments and meals. NDU is asked to provide the required garbage collectors.

Managing such services, especially when they are provided on a regular basis, can be difficult and require a full department for hiring, managing, and paying for the services. However, if all universities are connected over the internet with unique identifications for each device, the problem can be solved by using agents. All universities can have prime agents (the supreme agent in a group) that have different groups, such as waste management, parking, and security, under it. The leader of each group can publish its requirements through the prime agent, and the appropriate response can be calculated and selected based on the responses from other agents in the same region. There can be a negotiation process, and once both parties reach a consensus, an agreement can be signed between them and monitored by the interior agents of both parties.

In addition to managing the services, the health of the agents is also important. There will be fitness control agents who will check the health of other agents and approve their fitness. If an agent is found to be damaged or unfit, it will be referred to a recovering agent. If the problem is too grave for the recovering agent to handle, the owner of the agent will be informed. The process of adding a new university is also straightforward. The prime agent of the new university will request for its addition, and the request will be published by NDU to one hop. Based on the decisions by the respective members, the university will be added to the system.

Agent-based modeling (ABM) is a type of computer simulation that uses autonomous agents to represent individuals or groups in a system. These agents interact with one another and their environment, and their actions can affect the overall behavior of the system. ABM is often used in fields such as economics, sociology, and ecology to study complex systems that are difficult to model using traditional approaches. ABM can be combined with other modeling approaches, such as systems dynamics and network analysis, to create more comprehensive models. For example, in economics, ABM can be used to model the behavior of individual firms, while systems dynamics can be used to model the overall market. This combination allows for a more detailed understanding of how individual agents' decisions affect the market. Similarly, in ecology, ABM can be used to model the behavior of individual animals, while network analysis can be used to model the interactions between different species. This combination allows for a more detailed understanding of how changes in one species can affect the entire ecosystem.

Agent-based modeling (ABM) can be used to study the behavior of Internet of Things (IoT) systems, which consist of interconnected devices that can communicate and share information with one another. ABM can also be used to study the security and privacy of IoT systems, by simulating the behavior of malicious agents and the interactions between them and the other agents in the system. In short, ABM can be used to study the behavior of IoT systems, by modeling the interactions between the different devices and their environment and can also be used to analyze the security and privacy of IoT systems by simulating the behavior of malicious agents.

Agent-based modeling (ABM) can be used in combination with other modeling approaches to create more comprehensive models of complex systems. Here are a few examples of how ABM can be used in combination with other modeling approaches:

1. System Dynamics: ABM can be used to model the behavior of individual agents, while systems dynamics can be used to model the overall system. This combination allows for a more detailed understanding of how individual agents' decisions affect the system. For example, in economics, ABM can be used to model the behavior of individual firms, while systems dynamics can be used to model the overall market.
2. Network Analysis: ABM can be used to model the behavior of individual agents, while network analysis can be used to model the interactions between different agents. This combination allows for a more detailed understanding of how changes in one agent can affect the entire system. For example, in ecology, ABM can be used to model the behavior of individual animals, while network analysis can be used to model the interactions between different species.
3. Geographic Information Systems (GIS): ABM can be combined with GIS to model the interactions between agents and their environment. This allows for a more detailed understanding of how the spatial distribution of agents affects their behavior. For example, ABM can be used to model the behavior of individuals in a city, while GIS can be used to model the spatial distribution of resources and infrastructure.
4. Machine Learning: ABM can be combined with machine learning to improve the realism of the model by learning from real-world data. For example, ABM can be used to model the

behavior of individuals in a city, while machine learning can be used to infer the decision-making rules of the agents from data

The motivation behind this dissertation is the lack of a single, unified modeling framework for IoT systems. The proposed system, with its use of agents, addresses many of the challenges and complexities of IoT systems and provides a promising solution for creating a unified modeling framework for IoT.

## 1.2 Problem Description

The Internet of Things (IoT) is a rapidly growing technology that has the potential to revolutionize the way we live, work, and interact with the world around us. IoT systems are complex and dynamic, and they require a multi-disciplinary approach to design and development. The development of IoT systems involves many different aspects, including the collection of data from sensors, the manipulation of data, the decision-making process for actuators, and the integration of other components into the system. To ensure the success of IoT systems, it is important to have a deep understanding of these different aspects and to model them appropriately.

### 1.2.1 Research Gap

Proper understanding of IoT systems and their different aspects is crucial for the success of the system. Modeling different aspects of the system helps in determining the feasibility of the system, which is an important step in the development process. If the aspects of the system are not properly understood and modeled, there is a risk of system failure. This is because the system may not function as intended due to incorrect assumptions, incorrect data collection and manipulation, or incorrect decisions made about actuators.

One of the key benefits of modeling is that it allows for the identification of potential weaknesses and limitations in the system. For example, if the data collection process is not properly modeled, the system may not be able to collect all the necessary data, which can lead to incorrect decisions being made. Similarly, if the decision-making process is not properly modeled, the system may not be able to make the right decisions, which can result in the system not functioning as intended. Moreover, modeling also helps in identifying the interdependencies between different components of the system. This is important because it allows the software engineer to ensure that all components of the system work together seamlessly, which is necessary for the system to function as intended. Despite the rapid growth of IoT, there is still a lack of a unified framework for modeling IoT systems. This is a major challenge for software engineers who are responsible for designing and developing IoT systems. Many researchers have used different modeling approaches to represent specific aspects of IoT systems, but there is no single framework that provides comprehensive guidance on how to model IoT systems from a software engineering perspective. Although many researchers used different modeling approaches to represent or describe certain specific aspects of IoT systems, in general there is no single unified framework for modeling IoT systems [3], [4]. The absence of a unified framework leads to ambiguity in the usage of terms and concepts associated with modeling IoT systems, which can increase the risk of system failure.

There are various examples in literature where different modeling approaches have been used but there is lack of framework for modeling IoT systems from the perspective of software engineer. We are unable to find a framework that provides concrete guidance on how to model IoT systems that have fuzzy agents with in them. There does not exist framework that guides on modeling ambient that are a part of an IoT system. To address this challenge, there is a need for a unified modeling framework that provides concrete guidelines on how to develop different types of models for IoT systems. This framework should provide guidance on how to model IoT systems from different perspectives, how to model for different purposes, how to model human-involved IoT systems, how to model ambient-involved IoT systems, and how to combine different models into a single model. Such a framework would also help to remove ambiguities in the usage of terms associated with modeling IoT systems. Several researchers have proposed different frameworks for modeling IoT systems, but none of them have been widely adopted. In conclusion, there is a pressing need for a unified modeling framework that provides comprehensive guidance on how to model IoT systems. Such a framework would help to ensure the success of IoT systems by reducing the risk of system failure and by removing ambiguities in the usage of terms associated with modeling IoT systems. Further research is needed to develop such a framework and to validate its effectiveness in real-world IoT systems.

### 1.2.2 Problem Statement

IoT systems are complex in nature having different levels such as application level, virtual entity and service level etc. Modeling different constituents at different levels require different viewpoints.

There is a lack of unified framework for modeling complex Internet of things (IoT) systems that considers different components of such systems and provides rules according to well-known reference architectures.

We provide a framework to model complex IoT systems.

### 1.2.3 Research Questions

The overall goal of this research is to provide a unified framework for modeling complex IoT Systems. For this purpose, we have used a combination of multiple modeling approaches at different levels and for different purposes. Our research is based on following research questions: RQ1: Viewpoint Specific Modeling of IoT systems: How can we use different modeling approaches in combination for the modeling IoT systems (IoTS) with respect to specific viewpoints and for certain layers?

This question is related to the problem statement of device-to-device service provision. Different types of services, such as cloud platforms for other devices, gateways for other devices, and data collection from other devices by application service providers, can be created in such a system. Service-oriented devices are based on contracts. Additionally, IoT systems have various layers, including the device layer, communication layer, service layer, virtual entity layer, and application layer, and each layer has different types of entities, objects, or agents. Therefore, a combination of modeling approaches and different viewpoints are required to model such IoT systems. Different modeling approaches are used in combination for modeling IoT systems because each approach provides a different perspective on the system and focuses on different aspects. For example, some approaches might focus on modeling the data and information flow within the system, while others might focus on modeling the interactions between different components or the decision-making process. By using multiple approaches, it allows for a more comprehensive understanding of the IoT system and helps to identify potential problems or areas for improvement. Additionally, each approach may be better suited for modeling different layers of the system, such as the data layer, the network layer, or the application layer. Using a combination of approaches ensures that the IoT system is modeled from multiple perspectives and at different levels of abstraction, providing a more complete and accurate representation of the system.

RQ2: Modeling systems/ subsystems from software engineering viewpoint: How IoT systems can be modeled for the software engineering viewpoint?

This question is related to RQ1, as software is a key component of IoT systems. The viewpoint of the system for software engineers requires the use of multiple modeling approaches in combination, along with rules and procedures for their efficient use. IoT systems should be modeled from the software engineering viewpoint because it is essential to have a systematic approach to designing, developing, and testing these systems. Software engineering principles provide a set of guidelines and best practices that help to ensure the reliability and efficiency of software systems. In the case of IoT systems, these principles can help to ensure that the different components of the system are properly integrated, that the system is scalable, and that it can be maintained and updated over time. Additionally, modeling IoT systems from the software engineering viewpoint can help to identify potential design problems early on, which can reduce the risk of system failure. Furthermore, it provides a framework for testing and evaluating the performance of the system, which can help to improve the overall quality of the system.

RQ3: Modeling systems/ subsystems that have subsystems/ components of type “Fuzzy”: How IoT systems that have fuzzy subsystems within them can be modeled?

This question is related to RQ1, as fuzzy logic deals with the intermediate values between "true" and "false" and is used for effective decision making and cognition. IoT systems also include fuzzy devices and information. Modeling such systems requires consideration of fuzzy logic modeling in-combination with other modeling approaches. Rules and procedures for such modeling are required for Fuzzy IoT systems. Modeling IoT systems that have fuzzy subsystems is important because fuzzy subsystems introduce uncertainty and imprecision into the system. Fuzzy subsystems are often used in IoT systems to model human behavior or environmental conditions that are difficult to predict with certainty. Modeling these fuzzy subsystems accurately and consistently helps to ensure the overall reliability and robustness of the IoT system. Furthermore, modeling fuzzy subsystems allows for the simulation and testing of different scenarios and conditions, which can help to identify and resolve potential problems before the system is deployed.

RQ4: Modeling systems/ subsystems that have subsystems of type “Ambient”: How IoT systems that have ambient subsystems with in them can be modeled?

Ambient entities can incorporate other entities and have properties of mobility and limitations in movement. Smart transport systems and vehicles are examples of IoT systems that include ambient subsystems. Therefore, modeling such systems requires the consideration of ambient-oriented modeling in conjunction with other modeling methods. A framework that outlines the rules and procedures for combining these modeling approaches is needed to effectively model these systems. Modeling IoT systems that have ambient subsystems, such as a bus, is important for several reasons: Contextual Awareness: Modeling the ambient subsystems helps in understanding the context in which the IoT system operates. For example, modeling the bus in an IoT system provides information about the movement of the bus and the environment it operates in.

Interactions: Modeling the ambient subsystems helps in understanding the interactions between different components of the IoT system, including the ambient subsystems. In the case of the bus, this information can be used to understand how passengers, other vehicles, and the environment interact with the bus.

Optimization: Modeling the ambient subsystems can help in optimizing the IoT system. For example, modeling the bus can help in optimizing its movement, energy consumption, and resource utilization.

Decision Making: Modeling the ambient subsystems can provide valuable information for decision making in the IoT system. For example, modeling the bus can provide information about the best routes, stops, and schedules, which can be used to improve the overall performance of the system.

RQ5: Unified modeling framework: How can we develop a unified framework for IoT systems?

The research statement and RQ1 emphasize the need for a combination of different modeling approaches for effective modeling of IoT systems. RQ2 to RQ4 highlight various viewpoint-specific frameworks. However, a unified framework that outlines the rules and procedures for utilizing these different modeling approaches and viewpoint frameworks is necessary for a comprehensive modeling of IoT systems. A unified modeling framework for IoT systems is important because it provides a consistent and standardized approach to modeling the various components and aspects of these systems. This framework can help ensure that all necessary components and aspects are considered and incorporated into the model. Additionally, a unified modeling framework can provide concrete guidelines on how to develop models from different perspectives, how to model for human-involved IoT systems, how to model for ambient-involved IoT systems, and how to combine different models into a single model. This can help reduce ambiguities and improve the overall quality and reliability of the IoT system. Furthermore, a unified modeling framework can also help identify potential gaps in the current models and suggest ways to fill those gaps. This can lead to a more comprehensive and robust model that better represents the complexity of IoT systems.

## 1.3 Aims

The thesis has been formulized aiming in providing a unified framework for modeling of complex service oriented IoT systems. The framework provides guidelines for the researchers and engineers to develop their viewpoint models for the purpose of system analysis and design. The framework covers three aspects of an IoT systems i.e., Application and Software, Fuzzy components and Ambient.

## 1.4 Contribution

The research is adding knowledge in the domain of modeling service oriented IoT systems by providing a unified framework that provides guidelines and procedures for modeling along with the demonstration of the framework by modeling relevant case studies. Some of the key original contributions in the modeling and simulation of service oriented IoT systems are summarized as follows:

A. The first contribution of this research is a unified framework for modeling complex service-oriented IoT systems. The framework is using three other frameworks. The first framework is for modeling software engineering aspect or viewpoint of service oriented IoT systems. The second framework is for modeling Fuzzy information involved service oriented IoT systems. And the third framework is for modeling ambient involved service oriented IoT systems. The proposed unified framework has different viewpoint specific levels. We provided procedure to decompose systems into subsystems and compose systems from subsystems at different levels for the purpose of defining granularity and adding details.

B. The framework for software engineering allows using discrete and continuous time modeling and simulation approaches in combination for IoT systems. The proposed framework demonstrates on how to model Ad hoc and general systems IoT systems for software engineering purpose. It also considers the procedure for modularization and composition of the software for IoT systems. We presented procedures for creating discrete and continuous as well as Ad hoc, flexible, and general models for service oriented IoT systems.

C. We proposed a framework for modeling IoT systems that involve fuzzy information, and this framework is based on concepts of different modeling approaches. These modeling approaches include Agent-based Modeling, Network-based Modeling, Fuzzy-logic Modeling and Aspect-oriented Modeling.

D. We proposed a framework for integrated use of agent-based and ambient-oriented modeling for the purpose of modeling IoT systems that contains ambient in their composition. This framework provides us a way to model complex systems having agents that contain other agents and have the ability to move within a limited location. The framework also provides a way to represent different agents of different levels based on their dependencies. The framework also provides a set of rules and procedures to add details like message sending or receiving by certain agents in link with the representation of dependencies.

## 1.5 Research Methodology

The methodology used in this thesis is design science research, which aims to create new and useful artifacts in order to solve a specific problem. In this research, the problem addressed is the need for an appropriate modeling framework to model the

integration of IoT devices. The research started with an exploration of the problem, followed by the deduction and induction of the research question. The hypothesis was formed that a unified modeling framework could be developed for the modeling of complex IoT systems. The proposed framework was built by first formulating sub-frameworks based on the architectural components of IoT systems. These sub-frameworks were then integrated into a unified framework, providing a way to model the behavior of service-oriented internet-based devices and systems in complex scenarios. The unified framework was based on three distinct sub-frameworks for modeling IoT systems, each of which was capable of modeling the system from a different perspective. The first sub-framework focused on modeling IoT systems from a software engineering viewpoint, using concepts from different modeling approaches such as agent-based, aspect-oriented, contract-based, and service-oriented modeling. To further support our research, we provided a detailed literature review which included a deep insight into various modeling approaches such as agent-based modeling, ambient-oriented modeling, aspect-oriented modeling, fuzzy-logic modeling, network-based modeling, object-based modeling, and service-oriented modeling. Additionally, we examined the previous use of these approaches in combination and considered their potential for use in the modeling of IoT systems.

Furthermore, we validated our proposed framework by using it in a series of case studies and discussed the results in detail. The case studies allowed us to demonstrate the practical application of the framework and its ability to model the behavior of IoT systems in real-world scenarios. The findings of this research are expected to provide a useful tool for researchers, practitioners, and professionals working in the field of IoT. The proposed unified modeling framework provides a way to model the behavior of service-oriented internet-based devices and systems in complex scenarios, considering multiple perspectives and modeling approaches. This research adds value to the existing body of knowledge in the area of IoT by providing a comprehensive and integrated framework for modeling the behavior of IoT systems. The results of this study are expected to be useful for practitioners, researchers, and professionals working in the field of IoT, providing them with a new tool to help them better understand, analyze, and design IoT systems. The second sub-framework dealt with modeling IoT systems that have fuzzy values, utilizing concepts from agent-based, aspect-oriented, network-based, and fuzzy logic modeling. The third sub-framework was designed to model IoT systems with ambient entities, using a combination of agent-based and ambient-oriented modeling concepts. It is expected that this study will contribute to the modeling paradigm used in the understanding, analysis, and design of IoT systems, providing a unified framework that considers multiple perspectives and modeling approaches.

## 1.6 Thesis Organization

In this section, the first sub-section is providing an insight to the case studies that we used in this thesis. The second sub-section provides the sequence of the chapters included in this thesis.

### 1.6.1 Overview of Case Studies

We used the case study of universities located in twin cities of Pakistan (Rawalpindi and Islamabad) that are HEC recognized. The case study involves smart devices and smart systems such as smart parking and smart trash. We also used the case study of smart room that includes air conditioner, fans and heaters. The case study is based on smart temperature adjustment and switching of different devices. Another case study we used is of rapid bus transit system modeling.

### 1.6.2 Outline of Thesis

We have five chapters of this thesis. The rest of the chapters are as following:

Chapter-2 This chapter provides an overview of the field and the modeling approaches that exist in the area of service-oriented IoT systems. The various approaches, such as agent-based, ambient-oriented, aspect-oriented, contract-based, fuzzy-logic, network-based, and object-based modeling are discussed. The chapter evaluates the suitability of these approaches and provides a summary of the related work in the field of service oriented IoT systems.

Chapter-3 This chapter presents a unified framework for modeling service-oriented IoT systems. The framework is based on agents and provides a software engineering viewpoint for modeling IoT systems. The purpose, target and affordability of the framework are outlined, as well as the steps involved in its inception and elaboration. The use of contracts and the implementation of the framework are also discussed. Furthermore, this chapter provides a sub-framework that uses agent-based modeling in-combination with ambient-oriented modeling. It also provides a framework for modeling ambient systems.

Chapter-4 This chapter presents a series of case studies that demonstrate the use of the unified framework. The case studies validate the framework and provide a practical example of how it can be applied in real-world scenarios. The results of the case studies are analyzed and discussed. The framework and its results are evaluated using the case studies as a basis for comparison. The strengths and weaknesses of the framework are identified, as well as its limitations and future directions for improvement. The chapter also compares the framework to other existing approaches and provides a summary of its evaluation.

Chapter-5 The conclusion of the thesis summarizes the main findings and contributions of the research. It provides a summary of the framework and its practical application, as well as its strengths and weaknesses. The conclusion also suggests future directions for improvement and discusses the implications of the research for the field of service oriented IoT systems.

# Chapter 2

# BACKGROUND AND RELATED WORK

Agent-based Modeling Agent-based modeling (ABM) can be used to study the behavior of Internet of Things (IoT) systems, which consist of interconnected devices that can communicate and share information with one another. In an IoT system, each device can be represented as an agent, and ABM can be used to simulate the interactions between these agents and their environment. For example, ABM can be used to model the behavior of a smart home system, in which different devices such as thermostats, lights, and security cameras are connected to the internet and can communicate with one another. The agents in this system would represent the different devices, and the interactions between them would be based on the rules and protocols that govern the communication between the devices. Another example is the use of ABM to model the behavior of a smart city system, where different agents could represent traffic lights, transportation vehicles, and other infrastructure components that are connected to the internet. These agents can interact with one another, for example, by sharing data on traffic flow and adjusting traffic lights accordingly.

In AOM, an entity is represented as an "ambient," which own the properties such as mobility, inclusion and narrowness. The ambient are autonomous, and their behavior is determined by their internal states and the rules that govern their interactions with other ambient. AOM is often used in fields such as ambient intelligence, smart cities, and ubiquitous computing, to study the interactions between individuals and technology-rich environments. For example, AOM can be used to model the behavior of individuals in a smart city, where the ambient would include resources such as transportation.

Fuzzy logic modeling is a mathematical approach to modeling uncertainty and imprecision in real-world systems. It can be used in a variety of applications, including in the Internet of Things

(IoT). Fuzzy logic is based on the idea that there are many things in the world that cannot be described precisely in terms of "true" or "false," but rather as degrees of truth. The degree of truth is represented by a value between 0 and 1. In IoT systems, fuzzy logic modeling can be used to model complex relationships between inputs and outputs. For example, in a smart home system, fuzzy logic can be used to model the relationship between temperature, humidity, and air conditioning. If the temperature is high and the humidity is low, the air conditioning system might be set to run at full capacity. However, if the temperature is only slightly high and the humidity is very high, the air conditioning system might only run at a lower capacity. Fuzzy logic can also be used in decision-making processes in IoT systems. For example, in a smart traffic system, fuzzy logic can be used to determine the best route for a vehicle based on traffic conditions, road conditions, and other factors. The system might use fuzzy logic to determine the degree of congestion on each road and to weigh the importance of each factor in the decision-making process. In addition, fuzzy logic can be used in predictive maintenance in IoT systems. For example, in a manufacturing plant, fuzzy logic can be used to model the relationship between machine performance, vibration data, and the likelihood of machine failure. The system can then use this information to predict when a machine is likely to fail and to schedule maintenance accordingly.

Network-based modeling is a modeling approach that involves the representation and simulation of complex systems as interconnected networks. It is used to study and analyze the behavior of systems in which the components or entities are connected and interact with each other through a network. Network-based models provide a way to represent and analyze the interactions between components, allowing for the exploration of complex behaviors and patterns that emerge from these interactions. For example, network-based modeling can be used in social network analysis to study the interactions between individuals in a social network, such as Facebook. The individuals can be represented as nodes in the network, and their interactions as edges connecting the nodes. This allows for the exploration of patterns of behavior, such as the formation of communities, the spread of information, and the influence of individuals on each other. Another example is the use of network-based modeling in transportation systems, where the nodes represent transportation hubs, and the edges represent the connections between them. This type of model can be used to analyze the flow of people and goods through the transportation network and to identify bottlenecks and optimize routes.

Network-based modeling is a technique used to model the behavior of Internet of Things (IoT) systems. This modeling approach represents IoT systems as interconnected nodes (e.g., sensors, devices, and actuators) that communicate and exchange data over a network. The network structure, communication patterns, and data flow between the nodes can be modeled and analyzed to understand the behavior of the system. In the context of IoT, network-based modeling can be used to simulate the behavior of IoT systems, predict their performance, and identify potential issues before they occur. For example, network-based modeling can be used to optimize the communication protocols used by IoT devices, to determine the optimal placement of sensors, or to analyze the network's robustness against failures. Another use case of network-based modeling for IoT is in the design and implementation of smart cities, where large numbers of connected devices and systems need to be integrated and optimized. Network-based modeling can be used to simulate the behavior of the city's transportation systems, energy networks, and communication infrastructure, and to evaluate different scenarios to determine the best approaches for improving efficiency, reducing costs, and increasing sustainability.

Service-oriented modeling (SOM) is a software design approach that focuses on modeling the functionality of a system as a set of reusable and interoperable services. SOM provides a way to represent the interactions between system components as a series of well-defined and loosely coupled services. This approach enables the modeling of complex systems in terms of smaller, simpler building blocks that can be developed, tested, and deployed independently. In the context of Internet of Things (IoT) systems, service-oriented modeling is particularly useful as it allows the modeling and representation of IoT systems as a set of interconnected services, where each service represents a specific functional capability. This approach enables the development of modular and scalable IoT systems that can be easily integrated with other systems and services. Additionally, SOM provides a way to manage the complexity of IoT systems by breaking them down into smaller, more manageable components. Contract-based modeling is a software engineering approach for modeling and design of distributed systems. It refers to the use of formal contracts to specify the interactions between system components, including their expected behaviors and obligations. The contracts specify the conditions under which a service provider can deliver a service and the conditions under which a service consumer can use the service. In the context of IoT systems, contract-based modeling is used to describe the relationships between IoT devices and the services they provide. For example, consider a smart home system with multiple connected devices, such as a thermostat, smart lights, and a smart lock. A contract-based model of this system would specify the obligations and expectations of each device in terms of the services it provides and the data it generates. The contract would specify the conditions under which the thermostat should control the temperature, the conditions under which the smart lights should turn on or off, and the conditions under which the smart lock should lock or unlock the doors. By using contract-based modeling, the system can ensure that each component of the system operates as expected, and that the interactions between components are well-defined and consistent. In practice, contract-based modeling is often implemented using formal languages, such as the Service Component Architecture (SCA) or the Web Services Description Language (WSDL). These languages provide a way to specify the contracts in a machine-readable format, al lowing for automated testing and verification of the contracts. The use of contract-based modeling can help to ensure the reliability and predictability of IoT systems, and to support the development of large-scale, complex IoT systems.

This chapter provides the understanding of different approaches used in this research based on previous research. It also provides an insight to the relation of different terminologies and technologies used in this research.

## 2.1 Overview

In this chapter we provide the background and related work of this research. The chapter provides an insight to different modeling approaches. It also contains the application of different approaches for IoT systems. We separated different aspects of IoT systems and analyzed the application of approaches against those aspects as well. The chapter also includes the use of different modeling approaches in combination for IoT systems. In the end of this chapter, we provided the summary and research gap.

## 2.2 Modeling Approaches

This section is aimed at providing an insight to different modeling approaches. These modeling approaches include agent-based modeling, ambient-oriented modeling, aspect-oriented modeling, network-based modeling, object-based modeling, fuzzy-logic modeling, contract-based modeling and service-oriented modeling.

### 2.2.1 Agent-based Modeling

Agent-based modeling (ABM) is employed for different purposes in different ways, for instance, modeling complex adaptive systems for the purpose of prediction of the behavior of a system in response of any particular actions. Owing to the simulation friendliness of ABM, it has found higher degree of acceptability for predicting the behavior of certain agents in large-scale and crowd involving emergencies [5]. One of the strengths of ABM is its models with flexibility and simulation friendliness that enhance the analytical capabilities for simulations of business processes. Currently, there exist multiple frameworks for ABM having their pros and cons however, the analysis depends on the framework that has been used. In the arena of software engineering, numerous tools for modeling are based on the techniques of component-based software development [6].

ABM can be employed for the analysis of complex scenarios, such as world politics, where we can organize the actors involved as agents i.e., primary actors and meta-agents i.e., about other actors [7]. One can also use it for macro-economic research and development. ABM is conducive for modeling the systems that comprise of heterogeneous agents that are interacting in the system [8]. It has overridden mathematical modeling which used to comprise of heterogeneous agents and complex environment [9]. The behavior of the people in an uncertain condition or emergency can also be modeled effectively using ABM and simulations such as in case of earthquakes or tsunami and floods [10].

A type of agents known as Belief, Desire, and Intention (BDI) is well known for service-oriented systems and in software engineering domain. The belief is first constructed. The agents have a Belief for any action, based on defined belief the agent have some desires. Belief is updated every time the systems obtain some information from environment and the process of updating is known as belief revision. The belief can also be merged based on information [11].

### 2.2.2 Ambient-oriented Modeling

Among the main properties of ambient are Inclusion (container), Narrowness (limitation), and Mobility (not static). Narrowness implies that all ambient will always have movement in limited location. The property of Inclusion entails that all ambient should be capable of including other ones involved in system. The property of Mobility of all ambient denotes that ambient should not be static and it should change location, whereby when movement of an ambient takes place then it should take with it the sub-ambient that it is containing. For the purpose of developing a model, the key attribute to be assigned to the ambient is an identifier that is used for identification. Unlike Agent-based modeling where there are rules for model, in ambient-oriented modeling there are processes of each ambient and hence, these processes can be tracked in certain area being specified [12].

By putting intelligence to the former, IoT is creating a link between the objects from physical world and the objects in virtual world. Given the enormous diversity and huge number of physical objects to be connected through internet, the resultant systems and environment will be of highly complex nature. Such a combination of heterogeneous objects implicates ambient-oriented systems and when these systems are taken in the viewpoint of a software engineer then these are taken as ubiquitous systems. In [13], for the purpose of managing software development of ambient-oriented systems a scheme has been proposed which is based on formalism of Discrete-events Specification.

### 2.2.3 Aspect-oriented Modeling

In the early stages of its inception, aspect-orientation was used for the purpose of modularization along with at the level of programming. This approach separates the concerns i.e. crosscutting concerns and non-crosscutting concerns. It is now used for the purpose of software engineering at the development phases. On the account of its effectiveness, AOM is used for different purposes, have different notations, for different goals and also have different maturity levels [14]. For software engineering purpose aspect is a modular unit and it distinguish the crosscutting and non-crosscutting concerns. Explicit interfaces of aspects are required to describe the manner in which aspects interact with other aspects and also the other modules in the system and the interaction can either be heterogeneous or homogeneous [15].

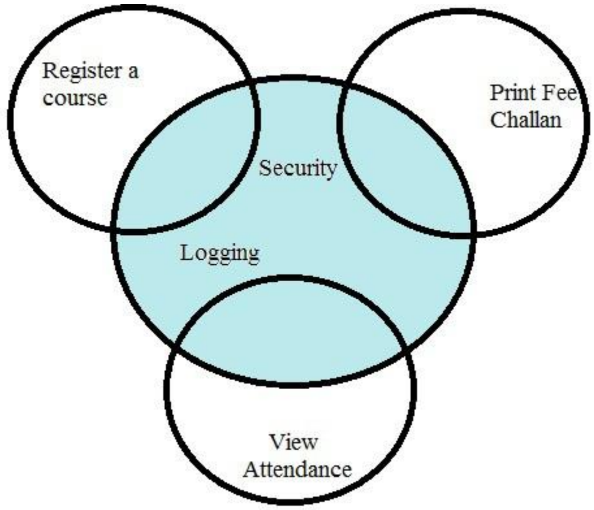


Figure 2.1 : Separation of Crosscutting and Non-crosscutting Concerns

In AOM, aspects tend to differentiate numerous modules involved in a system. Hence, we can define aspect as requirement implemented, in part, in multiple classes [16]. AOM has the denomination of a complete and comprehensive methodology that fosters every phase of software development life cycle through model driven development and architecture [17]. It can be employed for QoS modeling by using the Graphical notations in combination with the formal notations [18]. For a software AOM is a potent approach for modeling the security concerns. It has the potential to forestall certain attacks against the system, therefore, warranting its use as part of security engineering process [19]. Aspect-oriented modeling might play an important role in the development IoT systems that are distributed as it can be used for design of different components, interaction between distributed components and the integration of components in a system [20].

In AOM crosscutting concern is required to perform the core concerns. Figure 2.1 *: Separation of Crosscutting and Non-crosscutting* shows how a crosscutting concern is different from a non-crosscutting one. Internet of Things has much to offer in the form of valuable applications for various fields of life, for instance agriculture, smart city, transportation, and industry. There is a vulnerability of a number of issues such as: human intervention errors of unreliable outcomes due to missing data. There are instances of the use of AOM in combination with multi-agents approach to confront the issues related to smart healthcare applications such as missing data [21]. AOM lessens the complexity of models while modeling a system by dividing the system into different modules. It also improves the adaptability, reusability, robustness, and maintainability of software-based systems. Business processes are also effectively modeled using AOM [22]. The chief working principle of aspect-oriented modeling is differentiating the concerns that improves the quality of models. Well known Aspect-oriented modeling approaches in software development are [23]

1. Aspect-oriented Programming (AOP) Xerox PARC : The model of Xerox PARC is based on points-cuts and join-points.
2. Subject-oriented Programming (SOP): SOP is based on the division of the system sub-systems that are named as subjects and these subjects are then composed into system using composition rules.
3. Adaptive Programming: Adaptive Programming allows the object to interact with its immediate friends.
4. Composition Filters: The composition filters are used by distinguishing between the filters and the objects that are classes.

Preserving the ethos of the separation of concerns that are crosscutting and noncrosscutting, AOM might be used in combination with formal methods [24] as well as to address the security concerns in software at a very early stage [25], [26]. AOP structures include but are not limited to join points, advice, point-cuts and declaration. The concern is addressed by using the advice along with Join points and advice are used along with point cuts for the purpose of regrouping the concerns to compose the system [27].

### 2.2.4 Contract-based Modeling

Contract-based modeling is based on contracts. A contract has some assumptions and some guarantees. In some of the contracts the consumer has only the rights without obligations to be satisfied. This type of contracts is customarily termed as options. There are two well-known standard options i.e., American options and European options. Apart from standard options there are other types of options such as multi-assets options, multi-exercise options and exotic options [28]. A contract having firm commitment is known as forward contract. The customer side for internet storage consumer has higher risk in forward contract whereas the service provider has virtually no risk at all. The case with spot prices, however, is the other way around where the service provider has higher risk comparative to customer who has no risk at all. Due to the distribution of the risk among both the consumer and the provider options and vis-forward contracts are ranked comparatively better [29].

There are four types of contract models. The first one is obligation-free contracts. The second type of contract models is user-centric contract. The third type of contract models is provider-centric contract. The fourth type of contract model is customize-able contract. As the name implies, both the parties are free of restrictions in obligation-free contracts. In user-centric contract, the service provider is under obligation and responsible deliver agreed service and maintain quality of service. In provider-centric contract, the user is bound by the restriction to comply with certain rules within the given boundaries. The customizable model offers a mix of the aforesaid models with the provision to make desirable changes [30].

In software engineering the contracts are used for protecting a component of a software from the other component. Such contracts control the workflow of the system and monitor the execution of different components in the light of contract agreed by programmer. OOPS developers routinely use contract systems [31]. The programmers use to write the contracts for first order functions and also for higher-order modules for verifying the workflows and validation of the values obtained at different modules. Nevertheless, the understanding and interpretation of the first-order contracts are relatively easy. On the other hand, the high-order contracts are open to interpretation and may have different views about satisfaction. Currently, the contracts may be represented by various notions and also there are different languages for satisfaction of contracts [32]. The contracts that are composed of features' behavior of a program are known as Program contracts and the other type of contract that where the changed and unchanged features are considered are termed as changed contracts [33].

Predictive contract mechanisms are meant to be used for protecting against the inconsistency problem arise because of network latency. This latency is usually created by state changes communications [34]. The behavioral description for the web services can be made with the help of contracts. Successful transaction that occurs between the service consumer and the provider of the service is ensured statically by using contracts. Harnessing the contracts theory, the user of the service and the provider of the service can be formalized in a fashion well-suited to both parties and we can replace a service by other service safely [35].

During the course of the development of novel complex systems in an environment that is distributed, the interoperability of different modules of the system is safeguarded using contracts. So, we use the contracts to test the overall properties of the system and also for certain functionalities in a system [36]. Interaction of the components within a system that is brought together by contract existed well before a decade ago [37]. For instance, E-commerce applications are also centered on contracts that are mutually pledged by both the provider of the service and client. The rights and obligations are clearly mentioned on these contracts which guarantee the client's access. Generally, the contracts are by no means static, rather they are quite dynamic because new contracts surface and older one's fade, while there are some contracts which renew after their expiry. For the purpose of access control, it is advisable to use certified policy for the purpose of avoiding problems likely to emerge because of the dynamic nature of certain contracts [38].

The article [39], presents the use of contract-based modeling for the reasoning of software requirements that are based on timed safety. In this paper contracts are established as a pivotal asset for the systems' proper design and architecture. Building upon previous studies, the earlier mentioned article expounds that the contract will be composed of two facets that are assumptions which may be termed as pre-conditions and guarantees which may be termed as post-conditions. In the arena of software engineering, contracts can be grouped into four different types that are synchronization contracts, behavioral contacts, syntactical contracts and QoS contracts.

### 2.2.5 Fuzzy-logic Modeling

Electronics in general and computing machines in particular work on the basis of Boolean values, implying either true or false. However, things are not always such dichotomous in the real-world. Fuzzy logic (FL) designates Likert scale values such as minimum, average and maximum. It employs a degree of membership for different membership functions that is used to mark out the truth level of membership function or certain parameters [40]. Three typical steps are involved in the course of the development of a fuzzy inference system. The first step involves fuzzification of variables in which linguistic terms are used instead of crisp values. In the second step, the rules expressing knowledge are formatted. The third step called defuzzification is a direct opposite of the first one where the fuzzy values are again converted [41]. However, an additional phase of aggregation can also be considered [42]. FL is quite useful for data processing and mining where there are fuzzy values. It is also used in complex adaptive systems and optimization problems [43].

With the integrated use of multiple modeling approaches and different identification methods, FLM offers an effective, flexible, and transparent interface modeling tool for complex systems including non-linear. The interpretation resulting from fuzzy modeling is like to human thinking and marking out reality. Fuzzy Logic systems are principally knowledge-based systems which frequently symbolic processing in combination with numerical. Fuzzy logic is efficient for non-linear mapping of systems with beavering universal values. In FLM the quantitative data and qualitative information may be complimentarily used in combination. Fuzzy logic rule has two parts the first one is antecedent and the second one is consequent. Fuzzy partition is a methodology in which fuzzy reference sets are collected and generated for data representation and particular space is called reference point. The number of linguistic terms in a partition helps determine the level to which the details about the model and its granularity. The information termed as knowledge base is a database and rule base of a fuzzy system. Fuzzy Takagi-Sugeno models, relational fuzzy logic models, State-space fuzzy logic models and Input-output fuzzy logic models are among the most commonly employed models [44]. FL modeling is accomplished as: foremost, the antecedents are aggregated within the rules and are connected by “AND” or “OR” logic operation. The implication relation takes place in second step of modeling where, “IF-THEN” logic operation is applied. The rules are aggregated in third step where, “ALSO” logic is used for connection. The output is used to obtain the input in the fourth step of modeling using fuzzy inference system. And finally, the output being obtained is defuzzied to use for Boolean logic operations [45].

Among the various uses of fuzzy logic, one is energy-aware routing protocols that work on the basis of manipulating the data and after data manipulation the data is processed for the selection of best route for the purpose of efficient data transfer [46]. FL can be applied in IoT-based risk watching systems for analyzing the cold-related professional safety risks and evaluating them [47]. Another use of FL for IoT is local networks clustering and cluster head elections [48]. Also, FL has the potential to replicate human interpretations thereby offering maximum states output making it able to be efficiently used of complex IoT systems such as condition the room with respect to external atmosphere [49]. FL can also be applied to model the qualitative aspects of Likert scale values of day-to-day traveling choices for travel time perception [50]. Among the most useful applications of IoT are systems for fire monitoring, firefighting, and safety management. The concept of Fuzzy logic was first of all presented by Lutfi Zadeh to manipulate the data that contained fuzzy values and Likert scale knowledge. FL can be taken as a conversion of Likert scale values reasoning into formalized math values with rigorous and stringent mathematical symbols. FL can be applied in various applications of IoT that are based on information in natural language-based knowledge [51].

### 2.2.6 Network-based Modeling

The concept of Network-based modeling (NBM) stems from the graph theory and some more information is attached with the nodes and edges of graph [52]. This information is helpful in identifying the relations between different entities placed as objects and nodes in the form of symmetric relations and asymmetric relations. NBM is also helpful in characterizing the application resulting into the availability of deeper insights into the data structures and patterns [53]. The uses of NBM are spread over multiple domains ranging from social, biological sciences, engineering and technology which include but are not limited to public-opinion transmission[54], biomedical systems and processes[55][56], Microbiology [57], financial systems [58], Genetics [59], sensors network, wireless network, smart grids, traffic networks [50], transportation systems [60], design and manufacturing [61], geomagnetic fluctuations [62], Strategic relationships [63], supply chain, negotiation methods for suppliers[64] and big-data mining [65].

The introduction of web 2.0 has transformed the much-articulated idea of "world as a global village" into a reality. NBM can come handy in the study of relations different networks such as social media by applying parameters such as network-tree density and degree of nodes [66]. People with different mindsets belonging to separate groups and localities respond to specific situations in different ways. Social media data can be analyzed to predict the response of specific group of people against stated situations using Network-based modeling [67].

In order to construct a network-based model for the purpose of a system's analyzation, one has to break-down the complete-system into sub-systems as required for the level of granularity to be achieved. Thereafter, the relations among the sub-systems are modeled. There are some situations where weighted values play an important role in achieving the completeness and consistency of model by strengthening the relations [68]. NBM can be useful in creating high level models and to understand structural undercurrents of certain product or systems. NBM may also ease the understanding of the engineering of certain products and the processes involved. NBM can be helpful in uncovering the understanding of a system's environment in which it is meant to function in a specific manner [69]. Still, relational networks matching the attributes of Internet of Things are hard to find [70].

### 2.2.7 Object-based Modeling

The terms 'object-oriented modeling (OOM)' and 'Object-based modeling (OBM)' represent different points-of-view. Object-oriented modeling has its primary underpinning in software engineering. This technique uses a collection of objects to construct the object. The databased are modeled using this approach and, it is used for application modeling, modeling environment for language and unified data transformation. It comprises of three different phases; the foremost is the abstract level or analyzation phase where external details are focused. Further details, still limited, are available in the second phase. The focus in the third phase, called the implementation phase, is on the construction and functionality of the system [71]. OOM can be used to analyze the behavior of a system and also its physical deployment. Object-oriented modeling is helpful when an individual or a group working on a project is interested in detecting the flaws during early stage or when the procedures are meant to be understood [72]. In SE standpoint by the use of programming languages, OOM may integrated effectively. OOM permits models for relations and also provides processing tools. However, the troubles which one face while using OOM are the lack of proper validation mechanism and deficiency of standards. OOM is also ineffective for retrieval of information as it is comparatively difficult [73].

OBM considers physical objects as an extension to OOM. Every module in OBM is handled as discrete type entity. There are various possibilities of describing a system using this type of modeling. The architect of the system can adjust the number of attributes that are used for describing certain object by describing the level of abstraction of the model. Expressions that are Rule-based are utilized for the description relationships that exist between different objects in model. The system's global dynamics are accomplished by the integration of activities that belong to all objects. OOM can be utilized for modeling of existent systems as well as proposed systems that are complex. One can also use it for examining both the simple and complex systems problems. This modeling approach is utilized so far in multiple scientific domains [74]. Smart OOM can be utilized for modeling of Internet of Things systems by considering smart objects as basic units of modeling. Meta-models that provide the information about other models can be used at different levels for SDLC of Internet of Things systems [75]. As the world of internet connects different objects, so Internet of Things requires OOM for privacy [76].

The Integrated OOM templates, modularization and algorithms for automation can help automate assembly modeling system. Modularization will be instrumental in increasing flexibility, robustness, reliability, and expandability of the system. The concern of reusability of modules and independently changing the modules will be addressed by the object-oriented templates. For the sake of assembly planning,

automated algorithms can help retrieve the relational assembly metrics [77]. Among the other uses of object-oriented modeling are Internet of Things early warning system for flood due to snowmelt and knowledge base applications. Knowledge base can be used to store and organize the information belonging to objects or entities [78].

### 2.2.8 Service-oriented Modeling

Multi-agent and Service-oriented concept, when combined, offer flexible and intelligent systems development. IoT enables an intelligent connection of the day-to-day use objects and devices. When devices are intelligently connected to subsystems, it necessitates Interoperable and Open Standards, Simulation Visualization, Proper Communication Approaches, Networked Devices, Secure Infrastructure and Validation Methods [79]. The implementation of Service-oriented architectures in industry are taking over application-oriented approaches and products. An emergent challenge while developing IoT based systems is the modeling and analysis of the reliability of service-oriented architectures [80]. Service-oriented architecture aids the development of business software and applications for the purpose of solving complex problems by splitting the functionalities of the software under consideration and the information sources as different micro-services. These micro-services can be used alone as a service or in other case can be used in aggregation for provision of a service. Therefore, it results into a more flexible system which is composed of more independent components that are being aggregated [81]. The limited attributes of resources such as mode-processing abilities, bandwidth and server capabilities make it difficult to meet the service performance requirements. Congestion control needs to be watched for as well as properly modeling and analysis. Nevertheless, accounting for diverse objects during the course of modeling the control of congestion for accessing the service and analyzing it require great degree of cognizance and expertise [82].

A combination of Model Driven Architecture and Service Oriented Architecture can be instrumental in addressing the tricky issues currently facing the enterprise information systems [83]. [84] proposed the OWL-S model with certain extensions. The OWL-S is based on process model, grounding, and service profile. There are four attributes being used by OWL-S for specification i.e., the first one is inputs, the second one is outputs, the third one is preconditions and the last one is effects. One may use the context preconditions to specify the context requirements by specifying them as an input. Micro-service Architecture is among the latest service-based architectural styles for software applications and the systems that are distributed. Micro-services are self-standing and autonomous on the operation level and implementation along with the added advantages of service identification, composition, and provision. In this, each micro-service is responsible for one business or technological capability [85]. SO Architecture Modeling Language (SOAML), SO Reference Model (SORM), SO Reference Architecture (SORA), SO Architecture Ontology (SOAO), WS Architecture (WSA) and WS Modeling Ontology (WSMO) are among the renowned models, ontologies and meta-models for software applications and services [86].

## 2.3 Suitability of Modeling Approaches

From the literature and background study we derived the suitable use of different modeling approaches. Table 2.1 shows that which modeling approach is suitable for specific certain type of problem. The agent-based modeling approach is suitable for modeling flexible models, simulations, continuous time complex systems and systems with heterogeneous entities. Aspect-oriented modeling is used in software engineering. It promotes ease of reusability by modularizing the crosscutting functionalities and separating concerns. Network-based modeling uses the concept of graph transformation. It may be used to identify symmetric and asymmetric relations among entities. Fuzzy logic-based modeling is used for non-linear systems modeling. It is also appropriate for modeling systems that contain data related to human behaviors. Service-oriented modeling is suitable for modeling modules as services and micro-services. In service-oriented modeling one has to consider the provision of functionality by the service as key element. Object-based modeling is suitable for modeling physical devices interacting with the system or for smaller modules of software which combine and interact with others to provide the desired functionality of a system. Ambient-oriented modeling is suitable for the purpose of modeling where the entities involved in model are working as containers and are changing location within a limited spatial location. Contract-based modeling is suitable for representing components, modules or devices in graphical form along with the dependencies of functionalities representing with pre and post conditions.

Table 2.1: Suitability of Modeling Approaches

| Approach | Suitability of Application |
| --- | --- |
| ABM | Where simulation is required, for flexible models and also, easy for molding system with heterogeneous entities and for modeling continuous time systems |
| AsOM | Normally used in software engineering. Promotes ease of reusability and up gradation of system by modularizing the crosscutting functionalities |
| NBM | Uses Graph Transformation and may be used to identify, symmetric and asymmetric relations among objects/nodes |
| FBM | Most appropriate for non-linear systems modeling and where there is data related to human behaviors |
| SOM | For modeling modules as services and micro-services. The focus of each service or micro-service is on the provision of the functionality |
| OBM | For basic physical devices interacting with the system or for smaller modules of software which combine and interact with others to provide the desired functionality of a system |
| AOM | For the purpose of modeling where the entities involved in model are working as containers and are changing location within a limited spatial location |
| CBM | Modeling Representing components, modules, or devices in graphical form with pre and post conditions |

## 2.4 Service-oriented Internet of Things Systems

Networking in IoT has led the existing business models to unsettle. The peak in research moved from information technology towards self-service, mass customization and cloud manufacturing [87]. IoT is inviting attention to be evolving, ubiquitous and pervasive environment. Continuous research efforts are underway to implement IoT environments. The pressing need of the hour is to design environments for Internet of Things anticipatory to study whether they are compatible with the business goals or not [88]. With the use of IoT, the devices become capable of interacting and communicating with each other in fully decoupled way. However, among the key issues are preserving the overall quality and privacy of data exchanged during the interaction. These issues may be tackled using different access control mechanisms [89]. Contracts for context can be designed to preserve context's quality the systems that are independent and decoupled. These contracts have the advantage of run-time customizability for the addition of new clauses in the contract and refinement or removal of the previously existing clauses [90].

Internet of Things environment is an assortment of electronic devices that are inter-connected by the means of internet having separate goals and intentions whereby each group of objects has their peculiar operating system, and each group works on rules set in specific standards and protocols adopted for the purpose of communication and interaction. Hence, IoT envisages a seamless interconnection of heterogeneous devices to provide services for agriculture, business, management, logistics and other similar social applications. Depending on their goals, the coupling among the devices connected may be tight or loose. For IoT, one may use an integration of agent-based computing and cloud computing for the purpose of modeling systems having loosely coupled devices [91]. Desperateness and Heterogeneity devices is an issue to deal with for IoT operating systems. One of the chief hindrances in communication among devices, however, is that different devices present the data in different ways. Constraint of limitless of micro-controllers and the limitation storage are other hindrance to cope with heterogeneity [92]. Internet of Things systems can be taken as multi-agent systems in which every object is smart and is self-sufficient. This will lead to the provision of workable solution for management, programming and development of smart agent-based systems having object [93]. Recently Internet of Agents (IoA) as a term has come to fore from the combination of agent-oriented technology and IoT. The agents have the distinct character of intelligence and autonomy. The combination of both these technologies has been harnessed by different domains such as smart city, smart industry and smart applications for health. Heterogeneity is a key issue that Internet of Things is facing while it uses agents. The use of software agents in Internet of Things application may be a mitigating factor to cope with the heterogeneity. Semantic contracts may be used by the Internet of agents for the design and the development ecosystem that is intelligent or where for evolutionary process the participation of the end-user is considered as mandatory [94].

IoT has enabled a smart connection among machines as well as between machine(s) and the human thereby making the creation of smart services possible. In view of the large number of devices brought together by IoT, its services have grown more and more complex. Unfortunately, no specific tools or techniques are in place to analyze such services [95]. Each day, the canvas of IoT is rapidly enlarging with the continual increase in attachment novel systems, devices and services on daily basis. Though there is a total absence of systematic modeling and simulation process, yet for such systems one can use service-oriented modeling and agent-based modeling paradigm in combination [96]. ABM allows an in-depth investigation of components interaction and is, therefore, effective for modeling complex systems [97]. The devices in IoT are alive to sensing environment, coordinate, cooperate and interact with one another. These actions and the devices can be categorized as context, agent, service, object and model [98].

An important aspect of IoT is scalability and, therefore hard to take in account when one is modeling Internet of Things system by using conventional approaches as it is not convenient at node level to maintain the details [99]. Internet of Things devices might be semi-automated or fully automated, thus keeping the input required by human at the minimal level. The resultant distributed and open systems allow the addition of new devices. The technology being named as Blockchain is helpful in enhancing the security of system simultaneous to keeping it open. Blockchain has the potential for use in Internet of Things in order to keep the devices connected with each other without having an intermediary [100]. It can also be used in IoT for the management of access control for devices. The significant properties that Blockchain holds are data transparency, decentralized control, audibility, replicated and distributed data, security and decentralized consensus [101]. The smart-contracts that are used along with Blockchain self-enforce the context of the contract prior to transaction [102].

The privacy of end user which we term as consumer data is a vital issue in Internet of Things. The consumer's data can be accessed by anyone and from anywhere without his knowledge. Consumer License contracts are usually used for implementation of the informed consents and also comes with multiple limitations. These limitations can be overcome by defining and monitoring the usage control policies [103]. Services named as contract-monitoring can be used for monitoring the data flow control as per contacts' specifications [104].

When the physical objects are integrated in a system by wrapping it with service-oriented then they are called Servegoods. The IoT results from the addition of sensors accompanied by automaticity in servegoods interconnected through internet. The artificial intelligence along with decision making in real-time play an important role in IoT [112]. Sometimes, the state of an agent in multi-agent systems depends on state change in the other agent. In such a scenario, one agent will maintain the track of the state change in other and such kind of learning is called coordinated learning. Coordinated learning may hold an important position in large-scale internet of things where the agents lack the ability of self-adaption [113].

Since IoT devices cater for the services in real-world, integration between objects and digital world as well as accessibility at large scale of heterogeneous objects call for a machine processable and structured approach. In [114], the authors proposed a modeling approach based on semantic that is useful for different levels of IoT framework. Model may be integrated into Internet of Things framework by the use of mechanisms for association that are automated. Different tasks are interconnected with each other to perform certain functionality. This interconnection of task makes workflows which can be represented using task models. This model effectively designs the interface based on some specific requirements of user [115]. IoT technology has succeeded in attracting the enterprise systems that are applicable in supply-chain systems and production logistic. Due to the involvement of multiple people in the process of development of such systems, so far, a coherent and unified framework for modeling of such systems is absent. To overcome the complexity of modeling in such systems, one can use the top-down and layered modeling [110].

The three-layer architecture of IoT contains application layer, network layer and sensing layer. The application layer concerns itself with the functionalities that belong to the consumer, business process modeling and the workflows as it is the topmost layer. The next layer which is termed as network layer. It is composed of arrangements of service entity which is capable of managing access-control of service, the virtual entity which is a digital twin of any component or service and the information which utilizing the information being available connects these virtual entities that have been designed for certain services and for the notification of events to the applications and service the resources module is being used. The third layer is the sensing layer which is composed of devices for the purpose of information collection from the real-world. For a service consensus decision at edge nodes can be difficult because of the limitation of information. The second thing for consensus decision hindrance may be the overloading of demand for the purpose of deployment of SOIoT [116]. The IoT based services may be categorized on the base of their characteristics as Operation Service, Informational Service, Networking Service, Management Service and Security Service [3].

The traditional E-commerce business models have also been affected by the Internet of things. The IoT based business models ranging from the shipment of goods to retail stores and inventories have become an order of the day in the world of business. Since these models are in their infancy, much remains to be done and the research in this vein is already underway. Smart-contracts along with Blockchain have automatized the payment system and are likely to play a key role in business models that are based on Internet of Things [117]. Currently, IoT confronts development and requirement issues where the latter can be tackled by Agent-based modeling [105]. At present, no standard modeling methodology exists for IoT. One can use agent-based modeling in combination with network-based modeling to model certain scenarios of IoT [4]. The ABM distributed and adaptive parallel simulation approaches should be used in combination for complex IoT systems modeling. Moreover, for the purpose of maximizing holistic and realistic analysis, multi-level modeling is inevitable [118].

In the open systems, the operating conditions are continuously affected by the continual arrival and exit of new devices. Open systems, therefore, operate in an environment that is trustless and combines heterogeneous participants with the conflicting interests, thus enhancing possibility of mismatching specifications. Normally, systems like these, the participants are of two types; the one who provide the service and the other who consume the service. Different participants can belong to different teams and are designed for different purposes. The entry and exit of the participants may take place at different times with different organizations forming coalitions to achieve their goals [119].

The modeling considerations in developing IoT services are generally different from other software services. Ordinary software services do not need modeling consideration for the process of routing because there exists a centralized UDDI. Contrarily, IoT brings together heterogeneous and decentralized devices. Therefore, routing is a key parameter in service modeling for Internet of Things. The focus in the modeling of ordinary services is on the functionalities being offered service; but when there are IoT services in consideration then the devices are the key player and have vital consideration. The inability of objects/devices to provide more than one service at a time is yet another point of consideration [120]. An approach IoT based service modeling that is novel, has been described in [111]. In the two-step modeling of this approach, meta-modeling is accomplished in the first step. In the second step the operational modeling has been used. In the modeling stage abstract level representations for the descriptive purpose have been provided. For the purpose of formalization of services different notations are used in the operational modeling stage. This assists in the upcoming phases. ABM can be coupled with SOA to account for cyber-physical systems to deal with the complexity. The aggregated use of both these approaches is materialized in three steps or levels: intra-model simulation, inter-model simulation, and the modeling and simulation of individual agent [121].

Agents are autonomous but they are neither totally free nor totally dependent. Agents have dependence on environment which provides the condition for its existence [122]. In multi-agent systems there are several types of agents which interact with one another to achieve goals by generating sequence of actions [123]. Agents in multi-agent systems are heterogeneous and are grouped according to their behaviors and are modeled at multiple levels [124]. Recently a type of modeling named as multi-scale modeling evolved to represents complex systems at different scales [125]. Multi-scale modeling is used in engineering and material science by combining the emerging methods with existing ones leading to the predictive approach of modeling [126]. A term used in multi-scale modeling is scale bridging which is used to couple different models and keeping the relations at different scales [127]. Multi-scale modeling approach uses the computing concept of divide and conquer for enhancing the details of models. With the progress in computational facilities now the simulations of complex multi-scale models are used [128]. Its applications include avionics, automobile, materials, medical, electronics, chemical and pharmaceutical industry [129].

The comparison of System Dynamics, Discrete Event simulations and Agent-based modeling has been provided [130]. System dynamics (SD) is one of the earliest major and traditional modeling approaches being introduced in 1950s. It is used for modeling systems at abstract level and provides an aggregate view of the system. Another well-known approach Discrete Event Simulations (DES) was introduced in 1960s. It uses entities as basic element along with block charts and resources that describe resource sharing and entity flow. It also provides the representation of global systems behavior. Agent-based modeling (ABM) is quite recent and is used in a variety of disciplines due to stochastic models. The global system behavior is not defined in ABM. The difference between ABM and the other two is that ABM allows to model entities that have stochastic behavior whereas other two don’t allows this. ABM starts from bottom and moves up whereas the other start from top. DES and ABM both allow to model systems that contain heterogeneous entities at individual level whereas SD focuses on aggregate level [131]. In

DES the entities do not show individual behavior whereas in ABM entities interact with one another and show individual behavior. In Anylogic simulation tool, some agents in a model can be deterministic as well as other as stochastic in same model [132]. In DES there are queues where there is no concept of queues in ABM. Agents in ABM are active whereas entities in DES are passive. [133]. In operational research there exists an SD model against every ABM model. So, DES, SD and ABM are used in combination for more detailed models [134].

Modeling is used for different purposes including education, analyzing the natural systems and for engineering purposes. Agent-based modeling is well known due to its flexibility and simulation friendliness. By simulation friendliness we mean it is easy to simulate the model due to availability of relevant tools and tutorials. A large number of tools for ABM have been developed [6]. It is also used for the prediction of the behavior of certain actors to the system [5]. For complex scenarios like world politics where human behavior is important, it may be used differentiating the agents and meta-agents [7]. In the domain of economics, it helps to model heterogeneous interacting agents [8]. Due to its ease compared to the mathematical modeling, it is preferred for heterogeneous agents playing role in certain systems [9]. Different models have also been developed for analyzing the behavior of people in emergency situations and for crowd management in disasters [10].

New paradigms in computing are emerging frequently. However, the demands in the systems modeling also change with the emergence of new technologies such as modeling vehicles with respect to mobile ad hoc networks and services [135]. The vehicles traveling in long distance contain different agents within themselves as well, such as a vehicle containing traceable goods and human. So, what specific framework can be used to show the movement of vehicle and the agents it contains? Ambient-oriented modeling is based on the concepts of ambient calculus, ambient intelligence, and context aware systems. It provides a mathematical representation of systems [136]. There are three important properties of an ambient i.e., an ambient is moveable, an ambient may include other ambient and the movement of an ambient should take place in a limited space [12]. Internet of Things connects the physical objects with internet. This gives rise to new workflows. In the development of software, discrete time models play a vital role. Discrete time models may be developed using ambient-oriented modeling and may be helpful in Internet of Things application development [13]. [137] have described the use of ambient-oriented modeling for the virtualization of spatial aspects of physical things. [138] have provided AmbiNet which is an environment for ambient-oriented modeling.

Agent-based modeling has been used to model different aspects of air-traffic systems. Iyigunlu et. al. [139] used agent-based modeling for the investigation of boarding times for two different airplanes. Three new methods of boarding were also introduced with the help of simulations and modeling. Molina et. al. [140] analyzed different agent-based approaches for air traffic management and proposed their agent model. Bongiorno et. al. [141] also, presented agent-based model for Air Traffic Management. In this model the interactions between air traffic controllers and aircraft were focused. Delcea et. al. [142] considered different boarding strategies for developing a configurable agent-based model for identification of the best strategy. A well-known open-source tool named as NetLogo was used for the purpose of simulating the models. [143] provided an agent-based model for analyzing and forecasting air transport.

Agent-based modeling has also been used to model different aspects of electric vehicles. [144] analyzed different schemes for the cost-effective distribution of electric vehicle charging stations by using agent-based modeling. Yang et. al. [145] used agent-based modeling to propose an integrated dynamic method for detecting electric vehicles evolution patterns. Bischoff et. al. [146] used agent-based modeling for determining the impact of electrification of long-distance vehicles. [147] provided an ecosystem model for electric vehicles by using agent-based modeling for the purpose of analyzing different parameters like operational costs and workplace charging. [148] used multi-agent-based modeling for electric vehicles integration. In this article authors presented a platform which uses a combination of different simulation environments. [149] developed a framework for modeling electric vehicles. [150] used agent-based modeling for the comparison of four urban adoption policy scenarios of electric vehicles against a baseline. [151] used cognitive agent-based modeling in combination with artificial neural networks for demand of electric vehicles based on social influence.

Liu et. al. [152] discussed shared autonomous vehicles mode requests and used agent-based modeling for determining the preference of vehicles against certain distance slots. [153] used agent-based modeling for autonomous vehicles to provide safe traveling and avoid collisions. The model was simulated in Java Agent Development Framework. [154] used agent-based modeling for emergency evacuation and safety during every day move at trains and subway stations.

Bus Rapid Transit Systems remarkably facilitate people having average and low income by providing rapid source of public transportation. In developing countries which can't afford inter-city trains may use BRTS at a lower cost. However, easy access to the BRTS station is also an important parameter [155]. Modeling and simulation may be helpful in analyzing different aspects to increase the efficiency by reducing the fuel consumption and improving other relevant parameters [156]. There are also, feeder vans that operate as a shuttle service for customers to access the BRTS station [157]. Keeping the buses on schedule is also a difficult job in case of BRTS because of using common roads despite of separate tracks. Departure frequency along with signal priority may be helpful in keeping the buses on schedule as provided in mathematical model by [158]. Crowd management is also an important aspect to deal with in BRTS. Anticipative crowd managing procedures may be used to prevent congestion [159]. Agent-based modeling may be used to model BRTS by using the concept of multi-agent systems for the purpose of scheduling [160]. The use of different modeling approaches for IoT systems in different perspectives has been shown in Table 2.2.

Table 2.2: Techniques Previously Used for IoT Modeling

| Technique | Security | Software | Network | Services | Data | Contracts |
|---|---|---|---|---|---|---|
| ABM | | [105] | | | | [103] |
| AsOM | [20][19] | [20] [106] | | [107][108] | | |
| NBM | | | [70] | | | |
| FLM | [109] | | [46][48] | [47] | [43][42][49] | |
| OBM | [76] | [110][13] | | | [78] | |
| SOM | | [111] | | [83][111] | | |
| AOM | | [12] | | | | |

Ambient-oriented modeling till now has not been widely used for research purposes. However, it has been used for containers having the ability to move and have narrowness. In Table. 2.3, we may analyze the use of this approach for Relation representation and Representation of messages. But it is lacking in simulation against models created by using this approach.

Agent-based modeling has been used for analyzing the systems by using simulations. Abstraction of a model is considering a specific aspect of something or some process under study and representing it in simpler form [164]. According to [168], different levels of abstractions can be applied for detailed modeling of a system. The column "Abstraction" shows that there are different abstraction levels in agent-based models. The abstraction level of the model depends on the granularity of the model. Adaptive abstraction may be used for autonomously shifting from one level to the other.

Table 2.3: Use of Agent-based and Ambient-oriented Modeling for Transport Means and Coverages

| Approach | Strengths | Planning Horizon | Abstraction | Software |
|---|---|---|---|---|
| **Agent-based Modeling** | **Simulation Friendliness and Flexibility** [139], [141], [142], [144], [145], [146], [148], [153], [154], [156], [152], [149] | It can be used with open planning horizon [161]. However, it has also been used for planning horizon over a specific period of time as in [162], [163]. | It supports adaptive abstraction, cross-level interaction, abstraction of real environment and coupling of heterogeneous models. [164], [165] [166], [167] [168] | Netlogo is used by [163]. There are more than eighty agent-based modeling and simulation tools as discussed in [169]. |
| **Ambient-oriented Modeling** | **Representat ion of Inclusion Relations & Messages in models** [12], [136], [137], [138] | The models that are developed using this approach haven't involved any data as in [170] [171] .[136] | It allows the representation of ambient with different levels of inclusions [138] | AMBINET: An environment for ambient-oriented modeling [138]. However, this framework is not so much mature and is not marketed as are tools for ABM. |

An abstraction level for the crosslevel interaction can also be selected in agent-based models. Cross-level interaction occurs where one agent is at less detailed level whereas the other agent is at more detailed level. So, the interaction between different levels of granularity needs identification of agents with respect to their levels. The "Planning horizon" clearly depicts that the agent-based modeling has been used to build models over different planning horizons. The ABM models have been developed for open planning horizons means that without taking specific period in consideration. Agent-based modeling approach has also been used to model systems considering specific period of time. Whereas on the other hand the ambient-oriented modeling has been used for few models. We are unable to find any model in ambient-oriented modeling that considers planning horizon. There are many simulation tools for agent-based modeling, and some are open source as well such as Netlogo. One may easily find help for agent-based models. There is lack of specific simulation tool for ambient-oriented modeling. We found only one software tool for ambient-oriented modeling i.e. AMBINET. However, this tool has been presented in article [138] and we are unable to find it on internet. To the best of our knowledge, in agent-based modeling and simulations there is no formalized way to represent relations of agents where one agent is included in other. Also, agent-based modeling doesn't provide a formalized way for the representation of messages among agents at different levels of abstraction. Hence, there is need for a framework which provide solution to above mentioned lacking of agent-based modeling and simulations.

Boolean logic operates on two values that are 0 and 1. The values falling inbetween the two are not considered. In cognition the value 0 means False while the value 1 means True. Fuzzy logic uses the in-between values for decision making. It is a combination of quantitative and qualitative modeling. In Fuzzy logic model the input and output are quantitative whereas there is a set of qualitative linguistic rules. Rules are based on the elements in the environment. Fuzzy logic model consists of fuzzy set, membership functions, inference, fuzzification and defuzzification. It provides relatively wider choices to define uncertainty. The rules and the rule base are consistent and redundant in Fuzzy logic. It is also appropriate for modeling non-linear systems. It starts with rough approximation and leads to exact solutions. Unlike mathematical models, it provides simple models [172]. In Fuzzy logic the inbetween intervals of values are defined by a set of degrees of membership functions. Fuzzy Inference is the process of using fuzzy logic and has three steps. These three steps are fuzzification, inference and defuzzification [173].

Fuzzy logic has been used for a variety of systems. It has been used for smart irrigation to save power consumption [174]. In smart irrigation there are three input parameters i.e., humidity, soil moisture and temperature. These inputs are provided to Fuzzy logic controller and the Fuzzy logic controller produces outputs to the water supplying motor. In internet-of-things fuzzy logic may be used to improve the Quality of Service [175]. It has also been used to monitor older adults by using values from various sensors attached to wearables [176]. Fuzzy logic has been used for calculating trust in Trust Models which are used for web, ecommerce, and cloud computing [177]. It has also been used for estimating trustworthiness of Internet of Things devices [178]. For Internet of Things networks, Fuzzy logic has been used for authentication mechanism and enhancing security [179]. Fuzzy logic-based data controller to calculate the Rating of Allocation in IoT has been designed [180]. This data controller is useful in using Blockchain based security for Internet of Things in trustless environment.

While in Internet of Things machines, objects and people communicate with each other. This communication needs trust among different stakeholders either devices or human. Fuzzy logic-based model has been proposed to evaluate trust level as an output for each node and the node owning best trust level is selected for collection of information [181]. This helps the user in selecting the best trustworthy node on Internet of Things network. Fuzzy logic has also been used to measure the e-loyalty in Internet of Things based healthcare services [182]. Fuzzy logic has also been used for intelligent connectivity of Internet of Things based systems for smart irrigation in agriculture [183].

Agentification of Internet of things needs to consider multiple aspects and fuzzy logic technique may be used together with agent-based object modeling process [94]. The data obtained from IoT devices can be used for automatic employee performance measurement. Fuzzy logic provides more accurate results for employee appraisal measurement as it considers the values between 0 and 1 [184]. Fuzzy logic is highly effective when there is subjective data taken from human agents [185].

The smartness of grids saves handsome number of resources and power. These grids are capable of auto decision making according to certain conditions. Fuzzy logic can be used in smart grids for improving the decision-making process [186]. Fuzzy logic can be used for decision making of air conditioners in houses to control power consumption [187]. The data from different sensors attached to smart transformers can be manipulated using fuzzy logic controller for measuring health index of transformers [188]. In smart systems Fuzzy logic can be used for power saving as it is simple and consumes comparatively less energy in decision making [189].

Rule based systems are used for decision making process and these systems are composed of set of rules. These systems are useful for using human experts' knowledge in automation. Fuzzy logic rule-based systems can be mathematically formulized to validate completeness and consistency [190]. Fuzzy logic deals with approximate reasoning and handle partial truth i.e., values between 0 and 1. The rules provided by Fuzzy logic for reasoning are based on fuzzy measures [191]. Fuzzy logic can be used to improve the decision-making capability of health caring devices by monitoring the normal and abnormal activities of user [42]. Fuzzy vectorial space approach can be used despite of fuzzy rules for integration of emotions with knowledge to take decisions [192].

Wireless sensor networks play an important role in deployment of Internet of Things systems. Energy consumption in limited battery devices is an important factor to be monitored. One of the aspects of managing power consumption is load balancing. Fuzzy logic is useful in load balancing of limited battery devices [193]. Security challenges in wireless sensor network are also important to deal with. Fuzzy logic can be applied to model trust and reputation of Internet of Things environment [194]. The decisions for data perception can be made using Fuzzy logic to avoid traffic congestion in Internet of Things [195]. Real-time contextual information from user is used for access control in Internet of Things. Fuzzy logic can be used for risk-based models [196].

Fuzzy logic is a technique of artificial intelligence and should be considered in future of Internet of Things [197]. Fuzzy logic is a topic of great attention due to its vide range of applicability. It can be used for Internet of Things where there is data related to experts' knowledge. Hence, it can be applied for Quality of Service and Trust in Internet of Things [198]. Intelligent decisions for internet of things agents can be taken using fuzzy logic [199]. The traditional traffic management system should be replaced by Internet of Things based intelligent traffic administration system [200]. Fuzzy logic may be used in combination with semantic representation for modeling of coherent Internet of Things systems and software [201]. Fuzzy logic-based reasoning can be used for internet of things systems architecture and design [202].

In Internet of Things data is transmitted from peer to peer on request by users. These requests can be managed using fuzzy logic-based modeling [203]. The autonomous interaction of devices and services is one of the main objectives of Internet of Things. Fuzzy logic is useful for subjective values and liker scale measurements to be handled for automation [204]. Imprecise and incomplete data in Internet of Things can be modeled by using Fuzzy reasoner [205]. In Internet of Things based smart health care system fuzzy logic can be integrated with ontology to reason about diabetes related decisions to maintain lifestyle of patients [206]. When there is expert knowledge then it is usually not precise and generally fuzzy, so it's better to use fuzzy logic. In Internet of Things, there is usually noise in data from sensors which creates certain degree of fuzziness. The output of fuzzy logic-based systems is smoother than traditional systems. So, fuzzy logic is better to use for internet of things [207]. Fuzzy logic can be used for cognitive internet of things for better decision making [208]. Fuzzy logic can be used in combination with fog-based computing to offer quick decision making in Internet of Things [209]. In web and cyber physical systems reputation and karma are important factors for trust estimation. Fuzzy logic is used for estimation of reputation and trust and making decisions accordingly [210].

Internet of things business modeling is very important. Business models are based on building blocks which help to enhance the business; a framework for IoT business model is required [211]. The three elements of IoT are interaction, Communication and Networking of interconnected objects [212]. There are three types of IoT infrastructure and business models i.e., Telecoms, Internet, and Industry [213]. Multimedia objects are very important part of internet of things. The feature of multimedia objects has also been neglected by the research community for the internet of things [214]. These objects deal with cloud services. Abstracting of heterogeneity is required to represent the functionality of the objects [215].

There are two approaches for the payment of services in Internet of things [216]. There is a need of cooperative business model in Internet of Things [217]. Internet of things business market need to have a business model in which they can cooperatively interact. Interaction of different service providers will boost the trend. Also, there is a need of mechanism to define the trustworthiness of applications [218]. Trustworthiness is an important feature. Hence there is also a need of trustworthiness of environment in which IoT applications run. Some features that internet of things model should support are Heterogeneity, Scalability, Localization, Self-organization, Energy optimization and security as well as privacy [212]. The distributed model of IoT should have openness, security, reliability, viability, data management, interoperability, and scalability [219].

Agent Base Modeling (ABM) is a technique which uses set of agents to analyze the behavior of interacting objects in model [220]. This approach may be used to model interacting and decision-making objects [221]. The major benefits of agent-based modeling include flexibility and simulation friendliness. Simulations of agent-based models have been used to predict the behavior of certain agents in large scale emergency situations [5]. Hence, in disaster situations and emergency conditions it provides an effective way to analyze the emerging behavior of people [10]. Components of a system may be represented as agents to model complex systems [6].

Complex scenarios can be modeled using agent-based modeling by distinguishing the actors in the categories of agents and meta-agents [7]. For macro-economic research agent-based modeling may be used to model heterogeneous interacting agents in an efficient way [8]. Due to its user friendliness and providing a way to model continuous time systems in an efficient way it has replaced the mathematical modeling at a large scale. Social sciences prefer agents-based modeling over mathematical modeling due to its easiness and effectiveness [9].

Contract is an agreement between two parties. According to [30], there are four basic types of contract models. The first type of contract models is Obligation free contracts. The second type of contract models is User Centric Contract. The third type of contract models is Provider Centric Contract. And the fourth type of contract models is Customizable contract. Options are also a type of contracts in which the customer don't have any obligations but only have rights. American and the European options are two well-known types of options normally termed as standard options [28]. Forward contract is a type in which there is firm commitment between both parties [29]. The inconsistency due to network latency can be managed by using predictive contracts [34].

Contracts are used in software engineering for different purposes. Some of the uses of contracts in software engineering are guarding one part from other part of the program [31], for checking the validity of the claims about different parts and flow of the values [32], as program contracts for describing the behavior [33], in web services in the form of service level agreements [35], in distributed systems for ensuring interoperability [36], interaction between components of software [37] and for e-commerce purpose [38].

Contract-based modeling has been used for software engineering. The contracts are composed of left-hand and right-hand side. In some contracts there is require and provide side. Whereas the terminologies of assumption and grantee has also been used. Software may be validated against the specifications by using contract-based modeling [39]. Blockchain technology provides a way of interaction among trustless parties. It has been adopted by different industries and especially for cryptocurrency [222]. It uses smart contracts which are programmable and secured in blocks [223].

Aspect-orientation is used for modularization and for separation of concerns i.e., crosscutting, and non-crosscutting concerns [14]. It provides a way to modularization, removal of conflicts among modules and interaction of the modules [15]. An aspect is often a requirement which is partially implemented in two or more classes [16].

Aspect oriented software engineering uses model driven architecture [17]. It may also use formal notations and graphical notations together [18]. For security and quality focused systems, it is very effective to save the system from various attacks and to enhance quality of services [19]. It may also be useful for distributed internet of things software systems development [20].

Service-oriented software development is based on service-oriented architecture. It provides a flexible composition of the system. In combination with multi-agent systems, it may be used for development of intelligent systems [79]. It is now replacing application-oriented solutions. Reliability of service in IoT systems is hard to analyze and model [80]. It may be used in combination with model driven architecture to deal with challenging issues of enterprise information systems [83]. One of the differences between software services and IoT service is the dependency on physical objects in IoT. For IoT system OWL-S model with some extensions may be [84]. There are four attributes to consider while using OWL-S inputs, outputs, preconditions and effects [84]. Micro-services are used for developing system by composition of subsystems [85].

A digital twin refers to a virtual representation of a physical system, product, or process, created using real-time data and other information. It allows organizations to simulate and analyze the behavior of their physical systems, processes, and products, helping them optimize operations and make informed decisions. Digital twins are widely used across a range of industries, including manufacturing, healthcare, and smart cities. Manufacturing: In the manufacturing industry, digital twins are used to simulate and optimize production lines. For example, a digital twin of a production line can help identify bottlenecks in the production process and predict maintenance needs. This can result in reduced downtime, increased efficiency, and improved product quality. Healthcare: In healthcare, digital twins can be used to model patients, enabling healthcare professionals to simulate treatment options and make informed decisions. For example, a digital twin of a patient with a complex medical condition can help healthcare professionals understand the impact of different treatments and make predictions about the patient’s future health. Smart Cities: In smart cities, digital twins can be used to simulate and optimize the performance of urban infrastructure and services, such as transportation systems and energy grids. For example, a digital twin of a city’s transportation system can help city planners optimize routes, reduce congestion, and improve energy efficiency. Buildings: In the building industry, digital twins are used to model and optimize the performance of buildings. For example, a digital twin of a building can help building owners and operators understand energy usage, predict maintenance needs, and improve indoor air quality.

A digital twin is a virtual representation of a physical object or system, and it can be used in the context of Internet of Things (IoT) systems to better understand and optimize the behavior and performance of these systems. The different modeling approaches, such as agent-based modeling (ABM), service-oriented modeling (SOM), contract-based modeling, network-based modeling, and fuzzy logic modeling, can be used to create digital twins for IoT systems. For example, ABM can be used to model the behavior and interactions of individual devices or components within an IoT system, and a digital twin of an IoT system can be created by aggregating the individual digital twins of its components. SOM can be used to model the services provided by an IoT system and the relationships between these services, and the digital twin can be used to represent and monitor these services. Contract-based modeling can be used to specify the relationships and interactions between different components of an IoT system, and the digital twin can be used to represent and monitor these relationships. Network-based modeling can be used to model the communication and data exchange between different components of an IoT system, and the digital twin can be used to represent and monitor these interactions. Fuzzy logic modeling can be used to model the uncertainty and imprecision inherent in IoT systems, and the digital twin can be used to represent and monitor these uncertainties. By using these different modeling approaches, the digital twin of an IoT system can provide a comprehensive and detailed representation of the system, and it can be used to analyze, optimize, and control the system. Additionally, the use of digital twins can help to reduce the gap between the physical and virtual worlds, and it can improve the understanding and management of complex IoT systems.

## 2.5 Summary

Foregoing survey of literature in view, we have categorized different attributes that are involved in modeling of IoT systems and environment as shown in Table 2.2. The referenced table consists of a comparison of the use of different conceptual modeling approaches against these attributes. The detailed analysis of the use of conceptual modeling approaches is as following:

- Privacy/ Security Modeling: Privacy and security are the main concerns to account for in IoT and are, therefore, pivotal to consider and represent while modeling IoT systems. From the literature review it appears that researchers have been investigating this attribute of Internet of Things since 2014. We have seen that Fuzzy-logic Modeling, Aspect-oriented Modeling and Object-based Modeling have been used for this purpose.
- Applications Modeling: The user is always interested in the reliable, easy to use and

customizable software applications. Since, proper modeling and development of the application in line with the end user's expectations is highly important. Object-based Modeling, Aspect-oriented Modeling, Agent-based Modeling and Service-oriented modeling are used for modeling Internet of Things applications by the researchers.

- Network Modeling: Internet of Things aims to connect trillions of devices with the use of internet. Hence, appropriate modeling of these links and connections is of paramount importance, for instance, analyzing the best and most effective routing protocol in a particular scenario and for a certain system. Network-based Modeling as well as Fuzzy-logic modeling have been employed for analyzing the network-related issues.
- Services Modeling: The presence of physical devices in IoT services makes them slightly different from web service and, therefore, service assumes the form of an important functionality. For instance, a temperature sensing and sharing service of a specific building will work by combing a software application as well as some temperature sensing device. Aspect-oriented Modeling, Service-oriented Modeling and Fuzzy-logic Modeling have been used to analyze services in Internet of Things.
- Data Related Modeling: Trillions of devices connected through internet shall, undoubtedly, lead to the storage and processing of trillions of terabytes of information for industrial and research uses. So, the acts of modeling, manipulating and extracting that data in an effective manner necessitate using proper modeling techniques. Fuzzy-logic Modeling and Object-based Modeling have been used to model and analyze IoT data centered applications and scenarios.
- Contracts Modeling: Whenever there is a deal or understanding between two or more stakeholders, contracts are of prime importance. Service level agreements are used as contracts for software and web services. Design by contract is a software development methodology which uses contracts throughout software development life cycle. Blockchain makes use of smart contracts for enabling an open environment for the trustless parties to interact. Provision of IoT services will also necessitate contracts in future. The contracts will serve to assure the quality and smooth flow of the services. Agent-based modeling has been used to model contract centered systems.

The above description makes it evident that none of the techniques is individually capable of encompassing all the aspects. Every technique comes with certain limitations. Nevertheless, there had been some researches that combined different techniques to model IoT systems, scenarios and processes. These combinations of approaches are:

- Agent-based Modeling: Agent-based Modeling used for IoT systems and processes modeling in combination with various approaches such as Aspect-oriented Modeling, Network-based Modeling, Fuzzy-logic Modeling, Object-based Modeling and Service-oriented Modeling [21][4][109][96][91][79][121].
- Fuzzy-logic Modeling: Fuzzy-logic Modeling combined with Agent-based Modeling. It has also seen a use of Service-oriented Modeling, Ambient-oriented Modeling and Fuzzy-logic modeling in combination [47].

In this chapter we discussed the previous related research work in detail. We analyzed the use of different modeling approaches and frameworks for IoT systems. Although many researchers used different modeling approaches to represent or describe certain specific aspects of IoT systems, in general there is no single unified framework for modeling IoT systems. There are various examples in literature where different modeling approaches have been used but there is lack of framework for modeling IoT systems from the perspective of software engineer. We are unable to find a framework that provides concrete guidance on how to model IoT systems that have fuzzy agents with in them. There does not exist framework that guides on modeling ambient that are a part of an IoT system. So, there is a need for a unified modeling framework that provides concrete guidelines on how to develop different types of models for IoT systems; how to develop models from different perspectives; how to develop models for different purposes; how to model for human involved IoT systems; how to model for ambient involved IoT systems and how to compose different models into a single model. Such a framework also assists in the removal of ambiguities in the usage of terms associated with modeling IoT systems.

# Chapter 3

# UNIFIED FRAMEWORK FOR MODELING IOT SYSTEMS

Internet of Things Systems are composed of three major components as shown in Figure 3.1*: Components of Internet of Things System*. Although the fourth component i.e., internet or communication is the key component of IoT, it is excluded from being treated as a separate component due to its inherent nature in modeling. The things or hardware devices collect data from the environment or perform actions according to instructions. The data is stored in online storage through internet. Applications are built that use the data and provide certain functions. Our unified framework provides a way to model these three components of IoT systems. The unified framework is composed of three sub frameworks. All these frameworks i.e., unified and sub frameworks are based on agents. We are extending agent-based modeling for representing a complete IoT system at different levels. We are using multi-level modeling concept and system of systems concept to provide these frameworks. As discussed in previous chapter, agent-based modeling is appropriate for the representation of ad-hoc and continuous time systems.

From the literature review we have found that agent-based modeling can be used with some other modeling approaches as well. Here we provide an insight into the use of agent-based modeling with other modeling approaches discussed in literature review.

Agent-based modeling (ABM) and fuzzy logic modeling (FLM) are both methods that can be used to model complex systems. Both methods are used to model systems with uncertain or imprecise information, but they are based on different principles and are used for different purposes. ABM is a bottom-up approach, where the system is modeled as a collection of autonomous agents that interact with one another and their environment based on predefined rules. FLM, on the other hand, is a top-down approach, where the system is modeled using fuzzy logic, which is a form of mathematical logic that allows for reasoning with uncertainty. Despite the differences, ABM and FLM can be combined to benefit from the strengths of each method. For example, FLM can be used to model the decision-making process of agents in an ABM, allowing for a more realistic representation of the agents' behavior. ABM can also be used to provide a more detailed representation of the interactions between agents and their environment, which can be used to validate the results obtained from FLM. Agent-based modeling (ABM) and fuzzy logic modeling (FLM) can be combined to model the behavior of Internet of Things (IoT) systems. The combination of ABM and FLM allows for a more detailed representation of the interactions between agents and their environment and a more realistic representation of the agents' decision-making process.

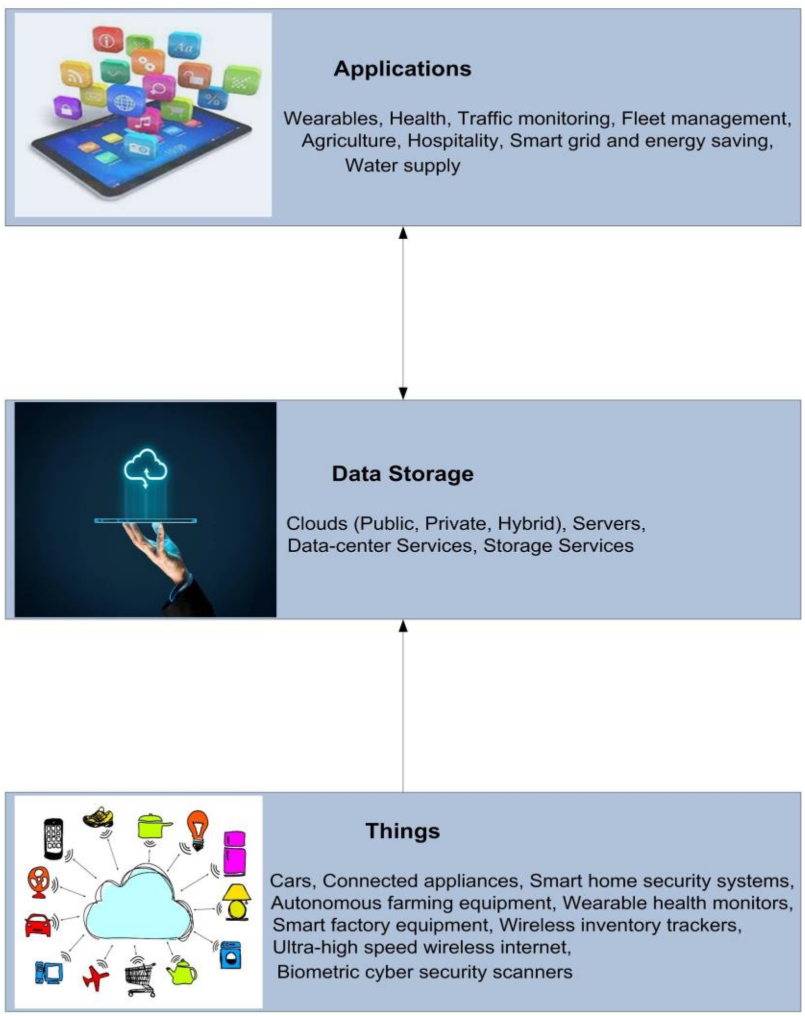


Figure 3.1: Components of Internet of Things System

Agent-based modeling (ABM) and aspect-oriented modeling (AOM) are both methods that can be used to model complex systems. ABM is a bottom-up approach, where the system is modeled as a collection of autonomous agents that interact with one another and their environment based on predefined rules. AOM, on the other hand, is a way to modularize the design and implementation of a system by identifying and isolating specific cross-cutting concerns, such as security or performance. Despite their differences, ABM and AOM can be combined to benefit from the strengths of each method. ABM can be used to model the behavior of the agents and their interactions in a system, and AOM can be used to identify and isolate specific cross-cutting concerns such as security or performance. These cross-cutting concerns can then be modeled as aspects that are integrated with the ABM model. For example, in an IoT system, ABM can be used to model the behavior of different devices such as sensors, actuators and controllers, and their interactions with one another and the environment. The cross-cutting concerns such as security, can be modeled as aspects that are integrated with the ABM model to ensure that the system's security is considered in all aspects of the design and implementation.

In an IoT system, agents can represent the different devices and their interactions with one another and their environment. Fuzzy logic can be used to model the decision-making process of the agents, allowing for reasoning with uncertainty in situations such as sensor data that is imprecise or uncertain. For example, in a smart home IoT system, ABM can be used to model the behavior of the different devices such as smart thermostats, lights and security cameras, and their interactions with one another and the environment. FLM can be used to model the decision-making process of the agents, such as how the thermostat adjusts the temperature based on sensor data and the preferences of the occupants. In a smart city IoT system, ABM can be used to model the behavior of different devices such as traffic lights, public transportation, and parking spaces, and their interactions with one another and the environment. FLM can be used to model the decision-making process of the agents, such as how traffic lights adjust the timing based on sensor data from traffic and weather conditions.

The Internet of Things (IoT) is a complex system that involves many devices and technologies that interact with one another and their environment. The behavior of these devices and technologies can be difficult to predict and understand, making it necessary to use advanced modeling techniques to study the system. Agent-based modeling (ABM) is a powerful tool for studying complex systems, as it allows for the modeling of the individual behavior of agents and their interactions with one another and their environment. However, ABM alone may not be sufficient to fully capture the complexity of an IoT system. This is where the combination of ABM with other modeling approaches such as fuzzy-logic modeling (FLM), ambient-oriented modeling (AOM), network-based modeling, service-oriented modeling, and contract-based modeling, can provide a more comprehensive understanding of the system. FLM can be used to model the decision-making process of agents in an ABM, allowing for reasoning with uncertainty in situations such as sensor data that is imprecise or uncertain. AOM can be used to model the interactions between agents and their environment, which is important in understanding the behavior of IoT systems. Network-based modeling can be used to model the communication and data flow between devices in the system. Service-oriented modeling can be used to model the services provided by the devices and how they interact with one another. Contract-based modeling can be used to model the agreements and obligations between the devices and the users.

Agent-based modeling (ABM) and network-based modeling (NBM) are two popular approaches in modeling and simulation of complex systems, including the Internet of Things (IoT) systems. These two approaches can be combined to address the complexity and heterogeneity of IoT systems and provide a more comprehensive understanding of their behavior. ABM is a simulation approach that models the behavior of individual entities (agents) in a system and their interactions with each other and with the environment. Agents in ABM can be used to model various components of IoT systems, such as devices, sensors, and actuators. The agents can have different behaviors, objectives, and decision-making rules, providing a rich representation of the system. NBM, on the other hand, focuses on modeling the relationships and interactions between entities in a system as a network. In NBM, nodes in the network represent the entities, while edges represent the interactions between them. The network structure and properties, such as connectivity and centrality, can provide insight into the behavior and dynamics of the system. By combining ABM and NBM, it is possible to address the complexity of IoT systems by modeling both the behavior of individual entities and their interactions at the network level. The combination of these two approaches can provide a more comprehensive understanding of the behavior and dynamics of IoT systems, and help identify patterns, relationships, and emergent phenomena that would not be evident from either approach alone. For example, an agent-based and network-based model of an IoT system could model the behavior of individual devices, such as sensors and actuators, as agents. The agents could interact with each other, forming a network of relationships. The network structure could be used to analyze the flow of information and data between devices, and the behavior of the individual agents could be used to analyze the performance of the system.

Service-oriented modeling (SOM) and agent-based modeling (ABM) are two approaches that can be used in the design and development of complex systems, such as Internet of Things (IoT) systems. The use of these two approaches in combination can provide a more complete and flexible solution for modeling complex IoT systems. SOM is a modeling approach that focuses on the design and development of services that are delivered over a network. It is based on the idea that the functionality of a system can be decomposed into a set of services, each of which is modeled and described using a service contract. The use of contracts enables the services to be loosely coupled and allows for the dynamic discovery and composition of services at runtime. ABM, on the other hand, is a modeling approach that focuses on the behavior and interactions of individual agents within a system. In ABM, each agent is modeled as an autonomous entity that can perceive, reason, and act in the environment. The behavior of the agents is defined using rules, which are used to control the agent's behavior and interactions with other agents.

The combination of ABM and SOM can provide a more comprehensive and flexible solution for modeling complex IoT systems. By using ABM to model the behavior and interactions of individual agents, and SOM to model the services provided by the system, the entire system can be modeled in a more complete and flexible manner. For example, consider an IoT system that is used to monitor the environment and control several sensors and actuators. In such a system, individual agents could be used to model the behavior of the sensors and actuators, while services could be used to model the functionality provided by the system. Using ABM, the behavior of the individual agents could be modeled and defined, while using SOM, the services provided by the system could be modeled and described using service contracts.

Contract-based modeling is a software engineering approach that defines a set of agreements or contracts between different entities in a system. These contracts specify the obligations, responsibilities, and expected behavior of each entity. In the context of IoT systems, contract-based modeling can be used to define the interactions between IoT devices, users, and other stakeholders in a standardized and enforceable manner. Agent-based modeling, on the other hand, provides a software engineering perspective for modeling complex and decentralized systems, where each device can act as an autonomous agent. By combining these two approaches, contract-based and agent-based modeling, we can create a more sophisticated and flexible IoT system. For example, consider a smart home system where multiple IoT devices are connected and interact with each other and with the users. The contract-based modeling approach can be used to define the interactions between the devices, for example, by specifying that one device should turn off when another device is turned on. The agent-based modeling approach can then be used to implement these contracts by assigning an autonomous agent to each device, which acts based on the defined contracts and performs actions on behalf of the user. Another example is a connected healthcare system, where various IoT devices are used to monitor the health of patients. The contract-based modeling approach can be used to define the interactions between the devices, such as specifying the data format and data exchange protocols between devices. The agent-based modeling approach can then be used to implement these contracts, for example, by assigning an autonomous agent to each device that collects data, performs analysis, and communicates with other devices and healthcare providers.

We aim at extending agent-based modeling by incorporating concepts of other modeling approaches for provision of details that are required for representing complete systems.

To use agent-based modeling in combination with other modeling approaches, the first step is to distinguish between different entities. There are certain properties which distinguish different types of agents.

Let's ρ is a set of properties to be considered in agent-based and ambient-oriented modeling that agents or ambient may hold. We represent these properties by Lower-case Greek alphabet letters i.e. Location is denoted as $\lambda$, identification is denoted as $\iota$, autonomicity is denoted by $\zeta$, Inclusion is denoted by $\mu$, Mobility is denoted by $\delta$ , Granularity is denoted by $\varepsilon$, Cognition is denoted by $\eta$ and Flexibility is denoted by $\kappa$.

$\rho = \{\lambda, \iota, \zeta, \mu, \varepsilon, \eta, \kappa\}$

Autonomicity of an agent is a must hold property in modeling. An entity $e$ is an agent *iff* $e$ holds autonomicity $\zeta$ for specific functionality.

**Definition 1: Autonomicity**

If $X$ is a set of inputs such that $X = \{x_1, x_2, x_3...., x_n\}$ and $Y$ is a set of outputs such that $Y = \{y_1, y_2, y_3, ..., y_n\}$ then, autonomicity is defined as mapping between input and output such that $\forall\, Y = f$

$(X)$ is defined and $x \neq y$.

While modeling, an agent under examination should have a unique identifier to track the agent. We call the property of owning unique identification as Identification $\iota$. Functions and rules to any specific agent are assigned using this identification.

**Definition 2: Identification**

If $I$ is a set of identities such that $I = \{ i_1, i_2, i_3, ....., i_n \}$ and $i_1 \neq i_2 \neq i_3 \neq \ ... \ i_n$ then identification $\iota$ for a set of entities $E = \{ x_1, x_2, x_3, \ \ , x_n \}$ is mapped as for $x_1$ is $x_1 i_1$, for $x_2$ is $x_2 i_2$, so on and every element of the set $E$ should have a unique value of $i$.

Every agent in the agent-based modeling should have a position in the environment which shows its location. The location of the agent is also, an important property which every agent must hold. Generally, location is identified by $x$ and $y$ co-ordinates in a 2D environment. In 3D environment location have $x$, $y$ and $z$ co-ordinates.

**Definition 3: Location**

Let L be as set where $L = \{£_1, £_2, £_3,..., £_n\}$ where $£_1, £_2, £_3, \ \ , £_n$ are different locations and $£_1 \neq £_2 \neq £_3 \neq ... \neq £_n$ and $\forall L \ \exists (x,y,z)$ where $x, y, z$ are the co-ordinates then the Location $\lambda$ may be defined as $L := (x, y, z)$ such that in 2D $z = null$. The agents which can move from a point to other own the property of mobility. In a model such agents are at one point over the first-time interval whereas at the other point over other time interval.

**Definition 4: Mobility**

If $\forall \ (£_1, ..., £n) \in L \ \forall$ agent $a_{i,j} \in A$ such that $£_i \neq £_j$ and an agent holds the property of mobility $\delta \ Iff \ £_1 \neq £_2 \neq £_3 \neq ... \neq £_n$ over $t_1, t_2, t_3, ..., t_n$ where $t$ is time against each location $£$.

Some agents are capable of including other agents in themselves. This property of including is named as inclusion.

**Definition 5: Inclusion**

Let, ( $a_1$, $a_2$, $a_3$, ...., $a_n$ ) $\in A$ where $A$ is an agent and $a_1, a_2, a_3, \dots, a_n$ are also agents then $A$ holds the property of inclusion $\mu$ and $A' \rightarrow a_1, a_2, a_3, \dots, a_n$ $iff$

1. $Ai \neq a_1 i \neq a_2 i \neq a_3 i \neq \cdots = a_n i$
2. $Al = a_1 l = a_2 l = a_3 l = \cdots = a_n l$
3. $\delta Al_1 \rightarrow Al_2 \Rightarrow \delta a_1 l_1 \rightarrow a_1 l_2 \Rightarrow \delta a_2 l_1 \rightarrow a_2 l_2 \Rightarrow \delta a_3 l_1 \rightarrow a_3 l_2 \Rightarrow \cdots \Rightarrow \delta a_n l_1 \rightarrow a_n l_2$

The inclusion $\mu$ is defined as $\forall x, y, z, \dots \in a$ at an interval of time $t$ in the model where, $x, y, z, \dots$ and $a$ are all agents then $a$ is holding the property of inclusion where, $l$ is location at an interval.

The process of learning from the environment and making decisions accordingly is called cognition of agent. In case of reflexive agents, they only use if-then rule. In utility-based decision making a utility function is provided to the agent and the agent takes decision based on the function. Adaptive agents adopt themselves according to different situations and decide accordingly. In goal-based decision making there are defined goals for the actions to take. So, appropriate technique should be used for decision making.

**Definition 6: Cognition**

The cognition $\eta$ of an agent is defined as the capability of an agent to get information $I$ from environment $e$ and take a decision $d$ with respect to $I$.

The granularity of the agent describes the level to which the complexity of the agent is modeled. Different agents have different granularity levels. This is the level of abstraction aimed to be achieved in the model. Inappropriate selection of granularity level may lead to difficulty of modeling or failure of modeling objectives.

**Definition 7: Granularity**

$\forall( g_1 , g_2 , g_3 , . . . , g_n ) \in \varepsilon \wedge \varepsilon \neq \varphi$ the granularity $\varepsilon$ of an agent $a$ or a system $s$ is defined as $\exists \varepsilon$ $\forall a \vee s$ , where ( $g_1$ , $g_2$ , $g_3$ , . . . , $gn$) are the levels of details about an agent or a system.
The properties $\rho$ can be characterized in different categories. The first one is set of those must own by an agent for modeling. The must own properties are autonomicity, identification and location on the surface of environment. Second set is the properties that are optional to agents. Another category of properties may be the properties that are based on levels such as flexibility and granularity.

$O\rho = \{\mu, \varepsilon, \eta, \kappa\}$

## 3.1 Agents

Agents are the basic building unit of agent-based modeling. In agent-based modeling, agents are autonomous entities. They may have intelligence and decision-making capability. Agents may be a living or non-living thing which interact with other entities in the model. Agent may be a static thing as well as a dynamic thing. The decision-making agents may be termed as cognitive agents. Cognitive agents can learn from the environment in which they exist and based on learning they may take decisions. Another type of agent may be proto agents. This type of agent is not fully specified but provides information about a specified agent to a non-specified agent. The third type of agent are meta-agents which are composed of other agents.

An example of a meta-agent may be a room which may have a temperature sensor as an agent, AC controller as an agent, Humidity sensor as an agent, TV as an agent and Refrigerator as an agent. Based on behavior there are three types of agents. The first one is which can move and is termed as Mobile agent. The second one is Stationary, and it can’t move across the landscape. The third one is which connects two agents and is named as “Connecting agent”.

**Definition 8: Agent**

An agent in agent-based modeling is a tuple $x = (\ \zeta f,\ i,\ L,\ O\rho\ )$ where,

- $\zeta f$ is the property of autonomicity for function $f$.
- $i$ is the identifier of the agent.
- $L$ is the location of agent where it may exist on landscape and $L \in \lambda$ .
- $O\rho$ is set of optional properties owned by an agent. $O\rho$ may be empty, single, or multiple for specific agent.

So, from the definition we can make a set of mandatory properties $M\rho$ of agents in agent-based modeling.

$M\rho = \{\lambda,\ \iota,\ \zeta\}$

We are using upper case Greek alphabets letters for types of agents. We represent Meta-agent with $A$ termed as Alpha, Cognitive-agent with $B$ termed as Beta, Mobile-agent with Γ termed as Gamma, Static-agent with $E$ termed as Epsilon, Proto-agent with $Z$ termed as Zeta and Connecting-agent with $H$ termed as Eta.

**Definition 9: Cognitive Agent**

Cognitive agent $B$ is defined as a tuple $B = (\ \zeta f,\ i,\ L,\ O\rho\ ) \land O\rho \neq \{\} \land \eta \in O\rho$.

**Definition 10: Mobile Agent**

Mobile agent Γ is defined as a tuple $\Gamma = (\ \zeta f,\ i,\ L,\ O\rho\ ) \land O\rho \neq \{\} \land \delta \in O\rho$.

**Definition 11: Static Agent**

Static agent $E$ is defined as a tuple $E = (\zeta f, i, L, O\rho) \wedge \delta \notin O\rho$.

**Definition 12: Proto Agent**

Proto agent $Z$ is defined as a tuple $Z = (\zeta f, I, L, O\rho) \wedge \forall Z \exists Y$ where, $Y$ is a specified agent.

**Definition 13: Meta Agent**

Meta agent $A$ is defined as a tuple $A = (\zeta f, I, L, O\rho) \wedge O\rho \neq \{\} \wedge \mu \in O\rho$ .

**Definition 14: Connecting Agent**

Connecting agent $H$ is defined as a tuple $H = (\zeta f, i, L, O\rho) \wedge U \cap Y = H \wedge H \neq \Phi$ where, $U \wedge Y$ are two agents.

The movement of the agent can be determined by the direction of the agent where it is facing. In simulation the color of the agent to display its state of action can be determined. Defined behavior of the agent that what certain action should it performs or in which state should it move. The key considerations of the agent are, first the Cognition of Agent and second Granularity of Agent.

**Definition 14: Thing**

An IoT thing, is an autonomous entity having a unique identifier *i*, an embedded system *S* and the ability to transfer data over a network *td*.

**Theorem 1**

IoT Thing is an agent $iff$ it has a location $L$.

**Proof**

Let set of properties owned by IoT Thing is denoted by $T\rho$ and properties owned by Agent are denoted by $A\rho$. Since, $T\rho = (\zeta f, \iota, S, td)$. $\zeta f$ is the property of autonomicity for function $f$ and $\zeta f \in A\rho \land \zeta f \in T\rho$. $\zeta f$ is common in both Agent and IoT Thing. $i \in A\rho \land i \in T\rho$ means that the property of unique identity is common in both. In agent-based modeling there should be a location of every agent. Since, if a thing has a location $L$ in the environment of model, then the thing is an agent. Moreover, IoT Thing should have embedded system and ability to transfer data over a network which are not conflicting the property of the agent. Since, IoT Thing is an agent.

## 3.2 Framework for Modeling IoT System from Software Engineering Viewpoint

While modeling one should be clear about the target stakeholders and the purpose of the model. The target stakeholders and purpose of the model can be well defined by asking a set of questions. These questions may include: who is going to use this model? How much details should be provided to keep a balance between the abstraction level and complexity? Is the target a professional or a normal person? What is the business expertise of the person? To which domain the target groups belong? Will a single model be sufficient for all the stakeholders or different models should be developed? After answering these questions there is always possibility of compromise between the details, clarity, and level of abstractions. To keep the balance and reduce the compromises different models are developed. Different views are developed according to different stakeholders. Models demonstrate the composition of the system and behaviors and interactions of the subsystems.

Internet of things involves physical things connected through internet. These things may have continuous-time data like temperature of an incubator, speed of a car and pressure of an electric rice cooker. For continuous time systems continuous modeling and simulation approaches are required. Whereas on the other hand there are various IoT systems which are generating and using discrete time data. Such systems are usually state based. State-based systems require discrete time modeling and simulation approaches and tools. Different aspects are required to consider while modeling IoT systems like either use a graph-based approach or an entity focused approach. Another point to focus is either go for an ad hoc or general one.

Internet of Things systems design and development may involve personals from different backgrounds to work in collaboration. Mechanical objects may be equipped by electronic kit and controlled through software connected to internet. This combination promotes a need of mechanical, electrical/ control system, telecoms/ network, and software personals to work in collaboration. Here, we focus on software engineering perspective of this system. The software engineer may require to develop software for the electrical toolkit attached, for software defined network (SDN) for the purpose of communication, develop a service to be used by other systems, implement business process, implement an application for the end user i.e. human being. Meanwhile, software engineers involve different roles like requirement engineering, design and architecture, implementation, testing, deployment, and evolution. In such a diverse system development, we may consider the system with the term used by Seymour Papert “an object to think with”. We may use modeling approaches for the purpose of communication among such diverse stakeholders.

This section is contributing in the following ways:

Provides a framework which allows using discrete and continuous time modeling and simulation approaches in combination for IoT systems.

The proposed framework demonstrates on how to model ad hoc and general systems IoT systems for software engineering purpose.

It also considers the procedure for modularization and composition of the software for IoT systems.

**Definition 15: Object**

Object can be entity; physical or virtual with an identity in modeling and it owns a set of properties that are considered for modeling purpose.

An object in modeling is a tuple $x = (i, O\rho)$ where, $i$ is the unique identifier of the object and $O\rho$ is set of optional properties owned by an agent. $O\rho$ may be empty, single, or multiple for specific object.

**Definition 16: Component**

A part of complex system that can be treated as an independent subsystem and it contains some entities or individuals that play certain role in the performance of its duties.

A component in modeling is a tuple $x = ( i , fr , f\rho )$ where, $i$ is the unique identifier of the component, $fr$ is the set of required input by the component and $f\rho$ is the set of provide output by the component.

**Definition 17: Contract**

Contract is a condition that is agreed mutually by two or more parties or stakeholders of system or entity about certain functionality, a set of functionalities, qualitative aspects of a system or a service.

A contract in modeling is a tuple $x = ( i , Cr , C\rho )$ where, $i$ is the unique identifier of the component, $Cr$ is the set of obligations by the consumer and $C\rho$ is the set of obligations by the provider.

**Definition 20: Service**

An entity either physical or virtual with standardized way for interoperability which cannot be owned but can be consumed.

If $X$ is a set of assumptions and $Y$ is a set of guarantees then $Z$ is a service $Z = \{ \zeta f , i , X , Y , In , Im \}$ $iff$ $\forall Z \; \exists ( a, b ) , a \in X \wedge b \in Y, b \neq \varphi \wedge a = \varphi$ where, $\zeta f , i$ means autonomous and identity as described earlier. $X$ and $Y$ are sets of assumptions and guarantees, $In$ means interface and $Im$ means implementation.

According to [224], A family of a set of web service properties is $V = Vi : i \in I$, where $I$ is finite or infinite index set. A web service is a relation on these set of properties $S \subset x Vi : i \in I$.

**Theorem 2**

Service Agent is a type of agent *i f f* it has a location *L*.

**Proof**

Let set of properties owned by Service is denoted by $Z\rho$ and properties owned by Agent are denoted by $A\rho$. Since, $Z\rho = (\zeta f, i, X, Y, In, Im)$. $\zeta f$ is the property of autonomicity for function $f$ and $\zeta f \in A\rho \wedge \zeta f \in Z\rho$. $\zeta f$ is common in both Agent and Service. $i \in A\rho \wedge i \in Z\rho$ mean that the property of unique identity is common in both. Hence, an agent must have a location in the environment means it should have connection with other sub-systems/ components of system. Moreover, Service has some additional properties which are not present in other types of agents i.e., it has assumptions and guarantees, interfaces and implementations.

**Definition 21: Aspect**

Aspect in aspect-oriented modeling is taken as a functionality, a component, an object, or a requirement of a system.

**Definition 22: Crosscutting Aspect**

A functionality, feature, property, or requirement of a system that is required for the quality of or execution of other more than one feature/ functionality/ property/ requirement is called a crosscutting feature/ functionality/ property/ requirement.

If $X$ and $Y$ are sets of concerns for different requirements, $X = a, b, c$ and $Y = c, d, e$ then the concern ”*c*” will be crosscutting as it exists in more than one requirement.

**Definition 23: Non-crosscutting Aspect**

A functionality, feature, property, or requirement that may not be required for the quality or execution of the other more than one feature/ functionality/ property/ requirement is called non-crosscutting feature/ functionality/ property/ requirement. If $X$ and $Y$ are sets of concerns for different requirements, $X = a, b, c$ and $Y = c, d, e$ then the concern "*a, b, d, e*" will be non-crosscutting as these don't exist in more than one requirement.

The framework that we propose here, uses a combination of four modeling approaches for modeling complex IoT systems for the purpose of software engineering. These approaches are agent-based modeling, contract-based modeling, aspect-oriented modeling and service-oriented modeling. These techniques have been used for software engineering and especially model-based software engineering [225]. Due to the use of different techniques this framework provides a way to model IoT systems at different levels. Agent-based modeling is better for flexible models with minimum details. This approach may also be helpful in modeling where target people have limited domain knowledge and simulations are needed for demonstrations.

Aspect-oriented modeling provides more details compared to agent-based modeling. It is useful when target viewer has at-least some domain knowledge. It may be used to prone out the requirements and remove the clashes among different modules. It may be used for requirements engineering of IoT systems. Contract-based modeling provides more details. It includes the features of the objects as well. It lies very close to the programming and hence it may be best understood by people of relevant domain. Service-oriented modeling is based on service-oriented architectures. It represents the system as a combination of services and micro services. Software services have "provides" interface but not "requires" interface. So, services are much independent components of the system. In Figure 3.2*: IoT-A Functional Model with highlighted software engineering layers*, we highlighted the layers of the IoT-A reference architecture where key contribution of software engineering for system development is expected. Our framework shown in Figure 3.3: Block-diagram of Modeling Framework, provides mechanism to model system keeping these highlighted layers under consideration. The framework is based on three phases that are, Inception, Elaboration and Composition. In the Inception phase the agents which basically are components, are identified. In the Elaboration phase the details about the agent or components are elaborated. In the Composition phase the identified and elaborated agents are composed in system.

The steps involved in modeling while using our framework are: first of all, defining the purpose, target and affordability of the model. The second step is defining the building blocks as agents. The third step is defining the features of the agents for semantic representations. The fourth step is elaborating the behavior of the agents using the services concepts and agents' relations. The fifth step is using the contracts and aspects for the composition of the system.

### 3.2.1 Purpose, Target and Affordability

There are various purposes of modeling. Modeling may be used to analyze systems based on the collected data, to transfer knowledge in an efficient way, to test a system or concept before implementation and to visualize any system or any component for design, development, and testing. While modeling, the first and foremost important thing one must define is the purpose of model. The second thing to know is the target viewer of the model. The modeling approach and the level of details highly depend on the target user of the model. The third point to consider is affordability of the model means how much efforts justify the importance of the model. The model may be of one-time use or sometimes it may have a lasting impact.

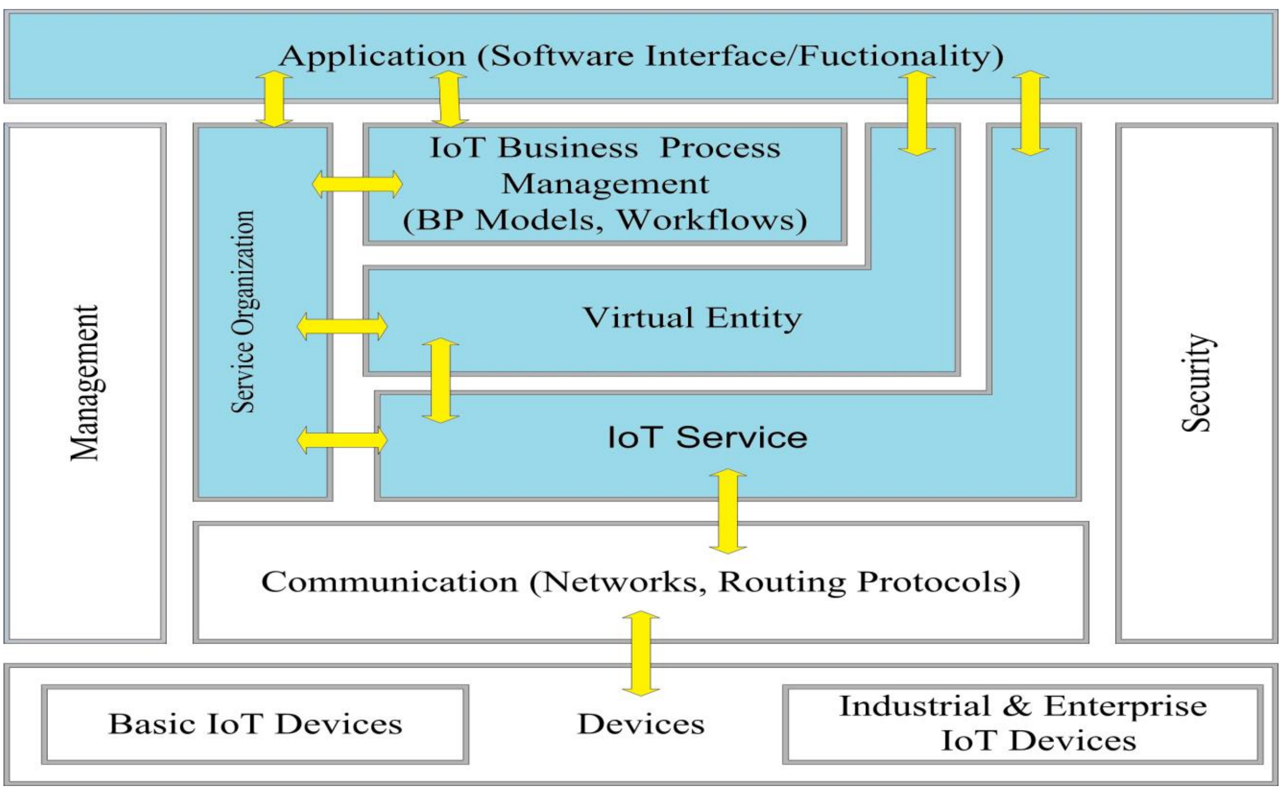


Figure 3.2: IoT-A Functional Model with highlighted software engineering layers

### 3.2.2 Inception by Identification of Agents

While modeling agents include any autonomous hardware or software or a unit of software. Agents may include but are not limited to Device, Component, Object, Service, or a Human. Sub system or super system may also be treated as agent. There are some common characteristics of agents. The agent should be autonomous and should own a behavior for responding automatically. Other characteristics include Learning from environment, adoptability which means changing behavior according to situation and work in decentralized environment. Cognitive agents are a type of agents which have decision power. Agents have identity, name, and behavior. There may be a group of agents with same behavior.

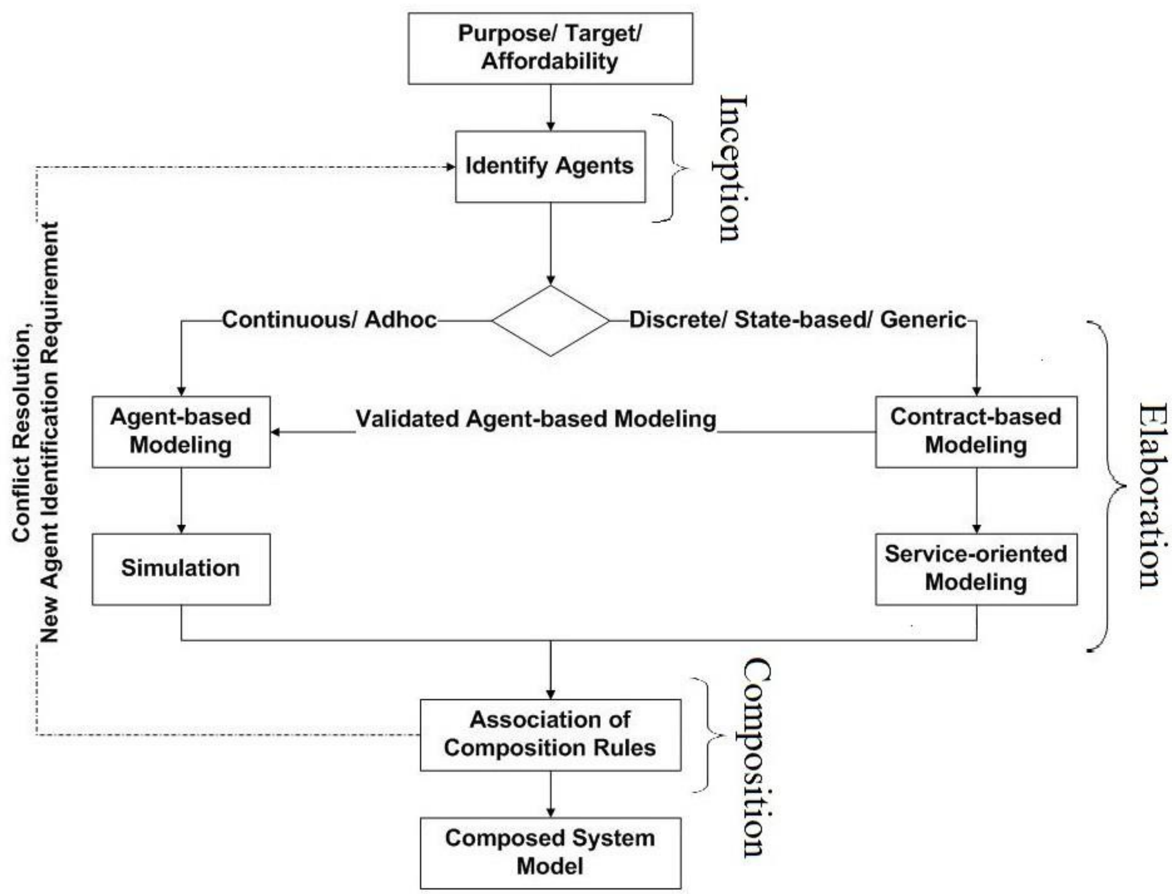


Figure 3.3: Block-diagram of Modeling Framework

### 3.2.3 Elaboration by Using Contracts

Contracts are the on-record requirements and provisions by an agent explaining the way of interaction with other agents. In other words, it defines the features of the agents which lead to interaction. Contract is the conditions against which actions are taken. There are pre-conditions and post-conditions. These pre-conditions and post-conditions explain the behavior of entities. Pre-conditions and Post-conditions may also be termed as assumptions and guarantees. Contracts may be visualized for better understanding. Visual contracts may be used to describe the correlation of different entities.

We use contracts to describe agents' behavior in terms of their operations. We make use of visual contracts for this purpose, which are defined as a set of graphs representing pre and post-conditions. We use double pushout approach to model these contracts, as production rules, and we represent system as a typed attributed graph transformation system (TAGTS). These rules have associated rule signatures as explained in [226]. We represent class diagram as type graph and initial state of the system in terms of a start graph. We also make use of these contracts to define agent interactions where pre and post-conditions explain behavior of entities, assumptions and guarantees.

### 3.2.4 Elaboration of Services

When the agents identified are services then they are elaborated using the concepts of service-oriented modeling. Software Services are a part of service-oriented architecture. Service-oriented architectures are used for large software applications. In service-oriented architecture distributed, independent and deployed components are integrated. This architecture is useful in development of web services and micro-services-based systems. A service in service-oriented architecture may be a platform independent module of software accessible by other application. For the purpose of composite modeling, we term the service as a functionality provided by an agent to the other agent or a human.

Extensible Markup Language (XML) is commonly used for modeling service-oriented systems. There are several extensions of XML as well. XML uses tags which are not normally pre-defined. Based on XML new modeling languages for service-oriented systems have emerged. Web Service Description Language (WSDL) is based on XML. It is used to describe web services. XML Schema Definition (XSD) provides a way of formally describing the elements in XML. Extensible Stylesheet Language Transformations (XSLT) provides a way of transforming documents from

XML to XML or other formats. Simple Objects Access Protocol (SOAP) is a protocol for web services which is also based on XML. Resource Description Framework (RDF) is also based on XML and is used to describe resources on web. XML and its extensions are open standards used by web services. So, XML plays a key role in modeling web services. Universal Description, Discovery and Integration (UDDI) contains information about the public interface of service, centralization of services and ease of publish/find operations. WSDL uses four major tags first one is < *types* >, this tag is used to define data types being used. Second one is < *message* >, this tag is used to define data elements. The third tag is < *portType* >, this tag normally contains the name of the port, the operation to be performed and the messages associated with the operation. The fourth tag is < *binding* >, this tag is used to define data format and protocol of port-type for the purpose of binding the service.

### 3.2.5 Separation of Concerns and Composition of System

Concern is information associated to the functionality of software. It may either be general or specific to a part of software. Concern is also term as aspect in software engineering. Concerns are Identified, Specified and Composed. So, discussing in terms of agents, aspects are the concerns of operation of any agent. A concern may be any activity required by an agent to perform its own task or any provision of functionality required by other agents to perform their tasks. Aspect-oriented modeling plays an important role in conflict resolution.

The first step in aspect-oriented modeling is identification of the concerns. As far as the concerns are identified the next step is specification of the concerns. The specification of concerns includes a set of parameters such as Name of the concern, Description of the concern, Sources of the information of concern, Classification means assigning the type to the concern that either it is emerged from functional requirement or non-functional requirement, Stakeholders who are linked with the concern, Responsibilities that mean what functionality it will have to provide, Contributions is the positive or negative interaction and Required Concerns that mean which other concerns are mandatory to perform its responsibilities. After specification, the concerns are composed. The concerns are combined and finding a match point which is a composition rule, and this composition is continued till we find whole system. Composition rule starts and ends with a term tags. The composition operators are:

### 3.2.5.1 Enabling:

The enabling inference of two agents is represented by $C1 \;>>\; C2$ which show that the process $C2$ starts after the completion of the process $C1$. In composition of system, we use this symbol to show the relation between two agents that are sequentially interlinked.

### 3.2.5.2 Disabling:

The disabling inference of two agents is represented by $C1 \;[>\; C2$ , means $C1$ is interrupted by $C2$. In composition we use this symbol to represent the relation of two conflicting agents. If this relation exists between two agents, then there should not be a direct link between those two agents.

### 3.2.5.3 Full Synchronization:

The full synchronization inference of two agents is represented by $C1 \,||\, C2$ , means both processes should execute parallel in synchronization. In composition we use this symbol to represent relation of two agents that have a direct link and execute in parallel.

### 3.2.5.4 Pure Interleaving:

The pure interleaving inference of two agents is represented by $C1 \;|||\; C2$ , means that the processes $C1$ and $C2$ can execute in parallel and if both the $C1$ and $C2$ are in ready state then either of them can execute first. In composition this symbol is used to represent relation of two agents that can execute in parallel as well as in sequence.

### 3.2.5.5 Direct Link:

The direct link of two agents is represented $C1 - C2$. In composition the direct link represents relation of two agents that directly interlinked. The expression $C1 - C2 - C3 - C4$ shows that $C1$ *and* $C2$ , $C2$ *and* $C3$ also, $C3$ *and* $C4$ are directly linked but $C1$ is not directly linked to $C3$ or $C4$ in this expression.

## 3.3 Framework for Modeling IoT Systems having Fuzzy Agents

Internet of Everything aims at connecting everything with the internet. It is an extended concept of Internet of things. Hence, Internet of Everything systems are more complex compared to internet of things systems. Modeling these Internet of Everything based systems is quite tricky. When there are fuzzy values involved in modeling any system, it is better to use fuzzy logic modeling. Fuzzy logic provides better, and more accurate results as compared to traditional modeling approaches. Every modeling approach has its own strengths and weaknesses. In this section we propose a framework for modeling cognitive fuzzy Internet of Everything systems. This framework uses concepts of Agent-based Modeling, Network-based Modeling, Fuzzy-logic Modeling and Aspect-oriented Modeling. For the proof of concept, we used a scenario example of an Internet of Everything system. We modeled the scenario using our framework proposed herein. We identified different agents from the scenario, extracted different symmetric and asymmetric relations and showed relations among agents in graphical representation. We implemented the fuzzy logic inference using MATLAB Fuzzy logic toolbox for the proof of concept. This framework provides a mechanism for more detailed Cognitive Fuzzy Internet of Everything models.

Internet of Everything (IoE) is an enhanced form of Internet of Things (IoT). In IoT there are three basic types of devices i.e., sensors, actuators and tags. In IoE, everything is aimed to connect with internet either human, devices, systems, or processes. When we have human in consideration while modeling a system, we expect fuzzy values as well. Fuzzy values are Likert scale values and are normally qualitative. When we use fuzzy logic, we convert these Likert scale values into semantic scale values in-between 0 and 1, which are normally intervals.

**Definition 24: Fuzzy**

The entity either an object, a system, an ambient or a component which considers the in-between binary values of 0 and 1 or true and false is known as fuzzy.

A fuzzy subset $A$ of a set $X$ is a function $A : X \rightarrow N$ where, $N$ is the interval $[0, 1]$

**Definition 25: Fuzzification**

Fuzzification is defined as the process of transforming a crisp set to a fuzzy set or a fuzzy set to fuzzier set. Basically, this operation translates accurate crisp input values into linguistic variables. In fuzzy sets, each element is mapped to [0, 1] by membership function.

$\mu : X \longrightarrow [0, 1]$

where, [0, 1] means real numbers between ”0” and ”1” (including ”0” and ”1”).

**Definition 26: Defuzzification**

Defuzzification is the inverse of fuzzification and it may be defined as the process of reducing a fuzzy set into a crisp set or to convert a fuzzy member into a crisp member.

In defuzzification, each element is mapped to “0” and “1” by membership function.

$\mu : X \longrightarrow 0, 1$

where, “0, 1” means crisp values 0 and 1.

**Theorem 3**

Fuzzy Agent is a type of agent i f f Fuzzy entity is an agent.

**Proof**

Let Properties owned by Agent are denoted by Aρ. As a fuzzy entity has fuzzy values. As fuzzy values are not conflicting with properties of agent. Hence, if a fuzzy entity is an agent, then it is termed as fuzzy agent. So, Fuzzy agent is a type of agent.

**Definition 27: Fuzzy Agent**

A fuzzy agent can be mathematically defined as follows: Let X be the input space and Y be the output space of the agent.

A fuzzy agent is defined by a set of fuzzy rules R = $\{r_1, r_2, ..., r_n\}$ where each rule $r_i$ is of the form:

IF x is $A_i$ THEN y is $B_i$

where, $A_i$ is a fuzzy set defined on X and $B_i$ is a fuzzy set defined on Y. The fuzzy agent maps an input x to an output y by applying the fuzzy inference process.

This process consists of the following steps:

Fuzzification: the input x is transformed into a fuzzy set by evaluating its membership in each fuzzy set $A_i$.

Inference: for each fuzzy rule $r_i$, the degree of support for the rule is calculated based on the membership of x in $A_i$.

Defuzzification: the output y is calculated as a crisp value by aggregating the results of the inference process and applying a defuzzification method, such as the center of gravity method.

Our proposed framework is based on the concepts of different modeling approaches. These modeling approaches include Agent-based Modeling, Network-based Modeling, Fuzzy-logic Modeling and Aspect-oriented Modeling. Agent-based modeling is helpful in modeling continuous time systems. The building block of ABM are treated as autonomous entities and are termed as agents. Fuzzy-logic modeling is based on Fuzzy Inference System, which is composed of Inputs, Fuzzy rules and Outputs. Network-based modeling uses symmetric and asymmetric relations and graphically represent these relations among entities. Aspect-oriented modeling separates the crosscutting and non-crosscutting concerns and provides rules for composition of systems. The proposed framework encompasses following stages as shown in Figure 3.4*: Framework for Fuzzy Internet of Things System*:

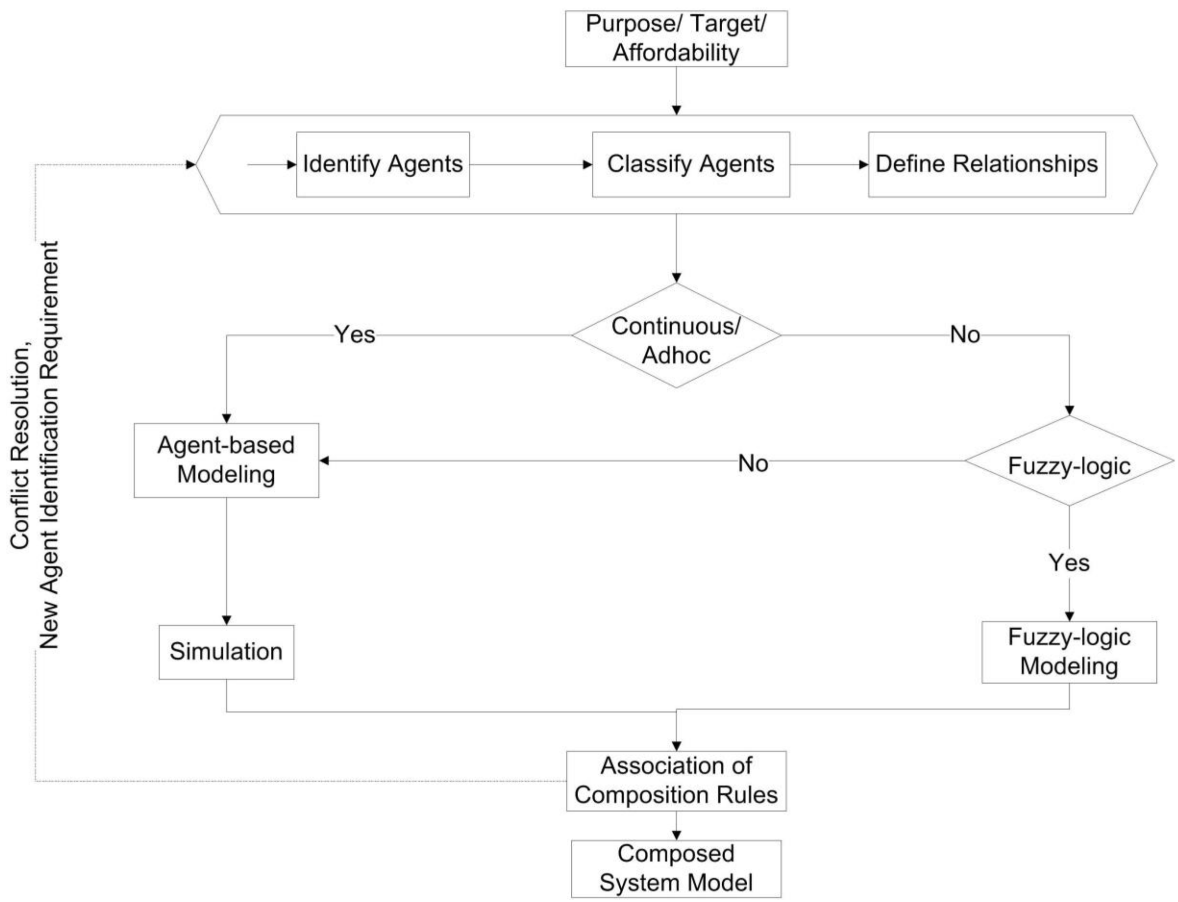


Figure 3.4: Framework for Fuzzy Internet of Things System

- Identification of Agents: at this stage the entities with autonomous behavior are identified. These entities are termed as agents and are assigned an identity.
- Relations of Agents and Networks: at this stage symmetric and asymmetric relations among the entities are formalized. On the basis of these relations network graphs are designed.
- Separation of Fuzzy and Non-Fuzzy agents: at this stage the fuzzy and nonfuzzy agents are separated based on their behavior and values.

- Fuzzy Inference for Fuzzy agents: at this stage rules are defined for fuzzy agents and their behavior based on the values they provide. Decisions of agents are based on these fuzzy rules.
- Composition of System: at this stage system is composed to check the consistency and completeness of the model.
- New Agent Requirement: when the system is composed, requirement for new systems can emerge. In this case new agent is identified. Hence, whole model will be updated according to newly identified agent.

Composition rule starts and ends with tags. The composition operators are:

### 3.3.1 Enabling:

The enabling inference of two agents is represented by C1 >>C2 which show that the processC2 starts after the completion of the process C1. In composition of system, we use this symbol to show the relation between two agents that are sequentially interlinked.

### 3.3.2 Disabling:

The disabling inference of two agents is represented by C1[> C2 , means C1 is interrupted by C2. In composition we use this symbol to represent the relation of two conflicting agents. If this relation exists between two agents, then there should not be a direct link between those two agents.

### 3.3.3 Full Synchronization:

The full synchronization inference of two agents is represented by C1||C2 , means both processes should execute parallel in synchronization. In composition we use this symbol to represent relation of two agents that have a direct link and execute in parallel.

### 3.3.4 Pure Interleaving:

The pure interleaving inference of two agents is represented by C1|||C2 , means that the processes C1 and C2 can execute in parallel and if both the C1 and C2 are in ready state then either of them can execute first. In composition this symbol is used to represent relation of two agents that can execute in parallel as well as in sequence.

### 3.3.5 Direct Link:

The direct link of two agents is represented $C1 - C2$. In composition the direct link represents relation of two agents that directly interlinked. The expression $C1 - C2 - C3 - C4$ shows that C1 and C2, C2 and C3 also, C3 and C4 are directly linked but C1 is not directly linked to C3 or C4 in this expression.

## 3.4 Framework for Modeling IoT Systems having Ambient Agent

Emergence of technologies such as internet of things, cloud computing, multi-agent systems, artificial intelligence and fast communication systems are disrupting existing business and process models. New models are frequently emerging. From households to industry, everything is demanding updates with respect to technology. Sophisticated and interlinked technologies are assisting and augmenting systems development without preliminary models is a risky task [227].

We find autonomous entities as basic elements in agent-based modeling (ABM), which are termed as agent. In cognitive agent-based models, agents have additional decision mechanism. There are certain rules defined in such models which guide an agent to decide or interact with other agent. The rules are defined in visual workflows in visual tools of agent-based modeling such as Anylogic whereas the rules are defined in code in some other tools as in NetLogo. While modeling a system with agent-based modeling approach, the composition of the model also contains an environment. One of the most prominent features of agent-based modeling is its openness to be used in combination with other modeling approaches [52]. According to [165] ABM has three major benefits that are its flexibility, capturing emergent phenomena and the provision of the natural description. According to [228], agent-based modeling is beneficial when the system being modeled contains heterogeneous and autonomous components. These components can act in distributed, local, parallel, has self-organizing capability and is emergent. Where, there is uncertainty in environment, along with spatial-temporal scales.

Ambient-oriented modeling is based on ambient calculus. An ambient is basically a container such as an airplane which contains other ambient in itself. The children ambient inside the parent ambient are dependent on parent ambient. This approach of modeling is very helpful for discrete time systems. It uses mathematical symbols to represent systems. The basic elements of ambient-oriented modeling are ambient, process, location, and contextual expressions. It is better for modeling spatial aspects of things in a model [171]. Although some researchers have attempted to develop simulations for this approach yet, much contribution is needed to make it simulation friendly. There is lack of mature simulation tool which provides facility to represent ambient-oriented models. It has been used for context-aware systems and for software development purposes. However, due to attraction of ambient intelligence it seems to gain more popularity in near future [138].

Advancements in information and communication technologies have influenced the transportation systems as well. IoT is used for tracking goods during shipments. Long route transport systems are also using latest technologies to remain updated on weather, roads congestion and other uncertainties. Within cities, people use different internet-based applications for booking appropriate vehicles, detecting suitable routes and other relevant points such as petrol pumps, tyre shops etc., Smart and autonomous vehicles have gained significant attention from research community and industry. All the vehicles are containers, they have ability to move, and they move in certain boundaries. So, these vehicles fulfill the properties of ambient. While modeling transport systems, one may use ambient-oriented modeling. Modeling and simulation are a crucial tool used in creating, maintaining, and optimizing transport system schedules.

To deal with the problem of traffic congestion is a common problem in urban areas and especially in peak hours. To solve this problem most of countries are moving toward BRT and Trains [229]. However, operation management and scheduling are still difficult for BRTs. This difficulty increases in case of peak hours and unusual burdens like holidays and events. Real-time monitoring and forecasting of crowed can be helpful in managing it effectively in case of such peak time [230]. BRT systems are usually cost effective and time effective [231]. Computer simulations may be used to examine different aspects of BRTs such as management, time effectiveness and cost effectiveness. For the purpose of simulation, the first step is to identify the problem and define the objectives of the model. Then in the second phase the model is developed and then tested and validated. When the model is formulated then on the basis of model the simulations are performed for specific abstraction level of the model [232]. Trip attribute approach may be used to compare BRT with other transport systems with respect to attractiveness [233]. In trip attribute approach there is a trip origin and a trip destination. Between trip origin and destination there are processes such as bus stops, bus moving and transfer from vehicle to vehicle. For all these processes attributes are defined such as access walk, wait time, travel time and egress walk. So, these attributes will be used to calculate total travel time and cost of the trip for comparison.

This section is based on one of previous chapter in which we examined different modeling approaches for internet of things (IoT). In previous chapter while mapping, we showed the use of both agent-based modeling and ambient-oriented modeling at device layer and virtual layer of IoT-A reference architecture. We also showed the use of these two approaches at the virtual entity layer. However, in this section we are providing an integrated use of ABM with AOM. The section aims in answering the following research questions:

- How complex systems having agents that contain other agents and have the ability to move within a limited location can be modeled?
- How can one represent different agents of different levels based on their dependencies?
- How can we add details like message sending or receiving by certain agents in link with the representation of dependencies?

### 3.4.1 Agent-based Modeling

According to [234] and [235], Agent-based modeling is helpful in examining interactions between individuals as well as to determine the implications of different hypotheses. It helps to create empirically supported models keeping the assumptions realistic rather than idealized. It also helps in modeling entities at different levels in a system. It helps to keep interdependence while modeling agents in a system. It also provides the facility to model heterogeneous agents in an efficient and easy way. Rules may be defined at different levels i.e., separate rule for a type of agent or population and rules of interaction among different agents and subsystems in a system. Hence, it provides a flexible way of modeling. Apart from these benefits there are also some considerations to keep in mind while using agent-based modeling approach. It may not be helpful in modeling discrete time events. Availability for standardized procedures for agent-based modeling while constructing and analyzing models is also a critical issue. Understanding the rules of the model in the simulation may also be difficult. However, agent-based modeling may be used in combination with other modeling approaches to overcome these issues.

### 3.4.2 Integrated Framework for Modeling

We propose a framework that uses concepts of both agent-based modeling and ambient-oriented modeling for a detailed model. This integrated use will help us in obtaining a comprehensive model covering all practical aspects necessarily required for a detailed model. The Figure 3.5*: Meta-model of Modeling by integrated use of ABM and AOM* shows a meta-model to depict the relation among different attributes and entities involved in modeling. Secondly, integrated use will help in handling both discrete time and continuous time systems. In complex systems at one component of the system may behave as continuous time whereas, the other component of the system may be a discrete time. Figure 3.6*: Integrated Framework Flow-diagram* shows a framework that is based on the combination of ABM and AOM.

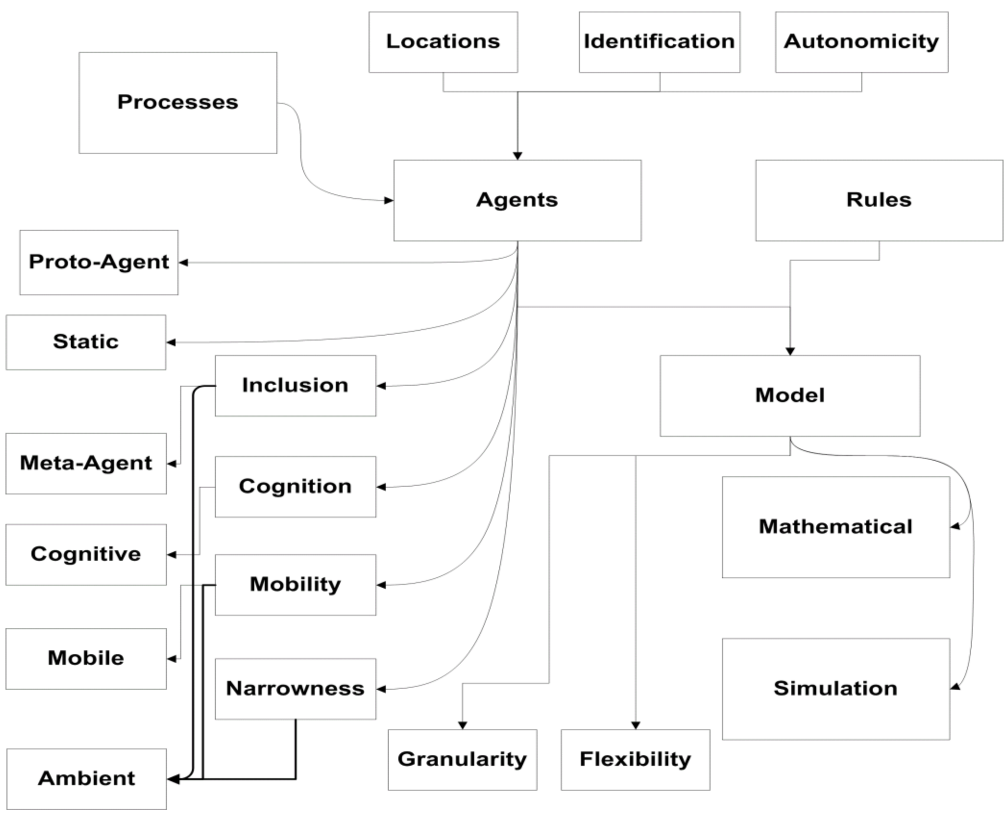


Figure 3.5: Meta-model of Modeling by integrated use of ABM and AOM

The framework shows a flow of modeling as Identification of agents, classification of agents, determining the relations of agents, formalization of processes, formalization of messages, formation of rules and at the last phase the simulation according to the rules. The model will be composed of a mathematical representation along with simulation. However, the model keeps the type of the agent and controls its behavior accordingly.

Considerations-before-starting: Before modeling one must select his desired level of abstraction or granularity and flexibility. The level of granularity differs according to the requirements of the system to be modeled. Also, different agents have different input variables. The flexibility of the system based on different agents and variables is defined such that one may analyze or learn the system using different values.

The proposed framework involves following steps:

Identification-of-Agents: The first step while using these approaches in combination is identification of agents. In this step the agents are identified, and they have assigned any identity as a unique identifier. At this step their functionality is also defined for which the agent is autonomous.

Classification-of-Agents: Upon the identification of the agent the agent is analyzed based on the properties it owns. After analyzing the agents are assigned different types for the purpose of elaborating their behavior in modeling.

Relations-of-Agents: Different agents in a model have relations with one another. Every agent should have a relation to at least one of its counterparts involved in the systems modeling. These relations may be symmetric or asymmetric. These relations are represented in the form of networks for better understanding.

Formalization-of-Processes: After defining the relations of the agents, the next phase is writing the processes of the agents according to the rules of ambient-oriented modeling. This will help in elaborating the functionality of certain agents and their interaction with other agents. These processes along with relations will be used for rules formation of the model.

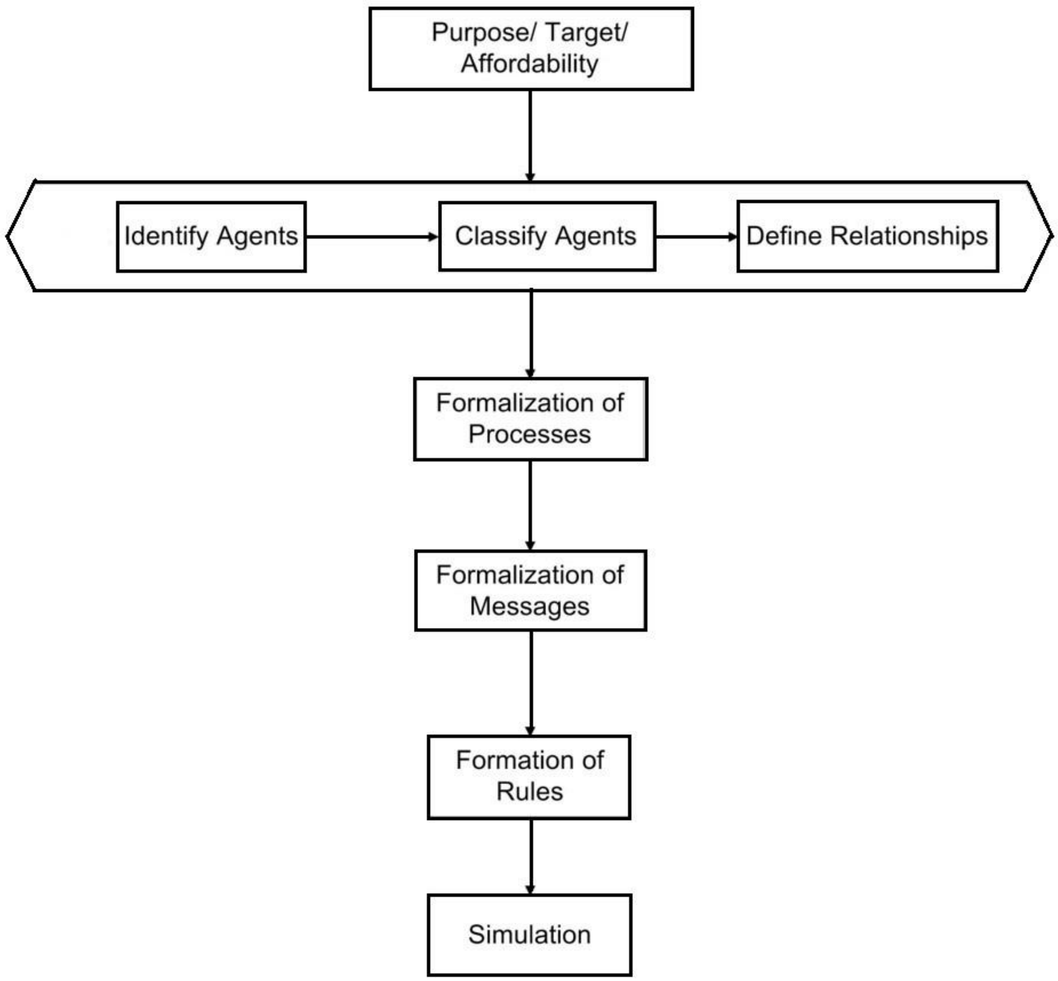


Figure 3.6: Integrated Framework Flow-diagram

Formalization-of-Messages: The messages between the agents will be modeled by using ambient-oriented-modeling symbols. The messages modeling will include the relations of the agents as well. The messages will contain symbols <> and () receiving, along with relations like parent, child, and sibling symbols.

Formation-of-Rules: The rules of the model will be elaborated. Based on the rules the simulations will be developed. The rules are coded in some agent-based simulation tools whereas; visual workflows are developed in others. Some common operations involved in rules will be set-up, creation of agents, interaction of agents, movement of agents, for ambient there will be operations included in rules such as in, out, move and copy.

Simulation: The simulation will represent the behavior of the agents in the system. It provides flexible and dynamic model. It is better for representing specific behaviors of system.

There will be three artifacts of using this modeling approach. The first artifact will be a simulation. The simulation will be helpful in representation of the model in both two and three-dimensional space. The second artifact will be representations of relations based on ambient properties. The third artifact will be the representation of the messages communicated between different agents along with ambient relations.

### 3.4.3 Ambient

Ambient are entities which fulfill the three basic properties. These properties are Inclusion, Mobility and Narrowness. We may also define ambient as an agent with the property of inclusion, mobility, or narrowness. Another definition of an ambient may be an agent which is both Mobile and Meta-agent along with limited change in location. Inclusion means that an ambient has property to include another ambient in it just like Meta-agent. Mobility is the property which means that an ambient can change its location and move from one place to the other like Mobile agent. Narrowness means that the movement of an ambient should be in a limited space. In ambient-oriented modeling, every ambient should have an Identifier and processes. Like agent, ambient also possesses identifier for the purpose of identification. This identifier helps in tracing and tracking ambient. Processes are like behavior of an agent. These processes include the rules an ambient must follow and the action an ambient has to perform.

We denote Narrowness by $v$ and is defined as:

**Definition 28: Narrowness**

The narrowness ν is defined as the limitation of agent O to move between specific points (x1.y1, x2.y2,..., xn.yn) on location L and can't move outside the specific points.

**Definition 29: Ambient**

An ambient in ambient-oriented modeling is a tuple $x = (\zeta f, i, O\rho)$, $\nu \wedge \mu \in O\rho \wedge \delta \in O\rho \wedge O\rho \neq \{\}$ where,

- $\zeta f$ is the property of autonomicity for function f .
- i is the identifier of the ambient.
- $O\rho$ is set of optional properties owned by an ambient. $O\rho$ must hold mobility and inclusion.
- $\nu$ is the narrowness which an ambient must hold.

Ambient-oriented modeling is based on ambient calculus, so it uses mathematical symbols to represent model. Primarily an ambient is considered as a container which contains other ambient. So, on the basis of this property, different types of ambient have been defined. There are three types of relations between ambient i.e., parent, child, and sibling.

- The symbol "↑" denotes parent. Mean if we say "X" is a parent of "Y" then the relation may be represented as: X ↑ Y. If "X" is an airplane and "Y" is a passenger, then ambient "X" is the parent of ambient "Y".
- The symbol "↓" denotes child. Mean if we say "Y" is a child of "X" then the relation may be represented as: Y ↓ X. If "X" is an airplane and "Y" is a passenger, then ambient "Y" is the child of ambient "X".
- The symbol "::" denotes siblings. Mean if we say "X" and "Y" are two siblings then the relation may be represented as: X :: Y. If "X" and "Y" both are two passengers, traveling in same airplane then we may call "X" and "Y" as siblings.

- The symbol "<>" denotes sending a message.
- The symbol "()" denotes receiving a message.

There are four syntax categories first is Location represented by α. Second is Opportunities represented by M. Third is Processes and it is represented by P. The fourth is Contextual Expressions represented by k. The primitives of an ambient may be:

**in:** An ambient may enter in a sibling. When the ambient enters in a sibling then it becomes child. If “X” and “Y” are two siblings and “X” enters in “Y” can be written as: X in Y.

**out:** An ambient may leave the parent. When the ambient leaves the parent then it becomes sibling of the parent. If “X” is parent and “Y” is child and “Y” leaves “X” can be written as: Y out X.

**open:** It is used to dissolve the boundaries of an ambient.

**copy:** The copy is used to create duplicates of an ambient.

**Lemma 1:**

If an ambient holds the Narrowness property, it must be limited to move between specific points on location L.

**Proof:**

As per the definition 27, Narrowness n is defined as the limitation of the agent O to move between specific points on location L and can't move outside the specific points. Hence, if an ambient holds the Narrowness property, it must be limited to move between specific points on location L.

**Lemma 2:**

If an ambient enters into a sibling, it becomes a child.

**Proof:**

As per the text, if X and Y are two siblings and X enters into Y, it can be written as X in Y. Hence, if an ambient enters into a sibling, it becomes a child.

### 3.4.4 Relation of Agent and Ambient

**Theorem 4**

Ambient is a type of agent i f f it has a location L.

**Proof**

Let set of properties owned by Ambient is denoted by Amρ, Since, Amρ = ( ζ f , ι, μ, δ, ν ) and if it has location L then Amρ = ( ζ f , ι, L, μ, δ, ν ) hence, $Am\rho \supset M\rho$. As, $A\rho = M\rho \cup \mu$ and, $\mu \in Am\rho$ so, $A\rho \subset Am\rho$. As $\Gamma\rho = M\rho \cup \delta$ and, $\delta \in Am\rho$ so, $\Gamma\rho \subset Am\rho$. Ambient own property of narrowness ν which is neither available in any other type of agent nor this property contradicts with the mandatory properties of agents. Ambient must own all the mandatory properties Mρ as well as some optional properties Oρ with an extra property which can be added to the set of properties ρ of agent. Therefore, ambient can be treated as a type of agent.
So, the sets of properties and optional properties can be revised as:

$\rho = \{\lambda, \iota, \zeta, \mu, \varepsilon, \eta, \kappa, \nu\}$
$O\rho = \{\mu, \varepsilon, \eta, \kappa, \nu\}$

## 3.5 Unified Framework

Our proposed unified framework is based on different layers. In the first section we have presented an overview of the proposed unified framework including decision making workflow. The second section contains the flow and relations of different constituents of the framework. The third section contains definitions.

The levels of unified framework are shown in Figure 3.7*: Unified Framework for Modeling Service-oriented Internet of Things systems*. The unified framework is based on different layers of service oriented IoT systems. The framework has been provided on the basis of different desired viewpoints involved during the feasibility study, analysis, design and development of a service oriented IoT system. The levels of the framework and the decision making are based on the desire viewpoint.

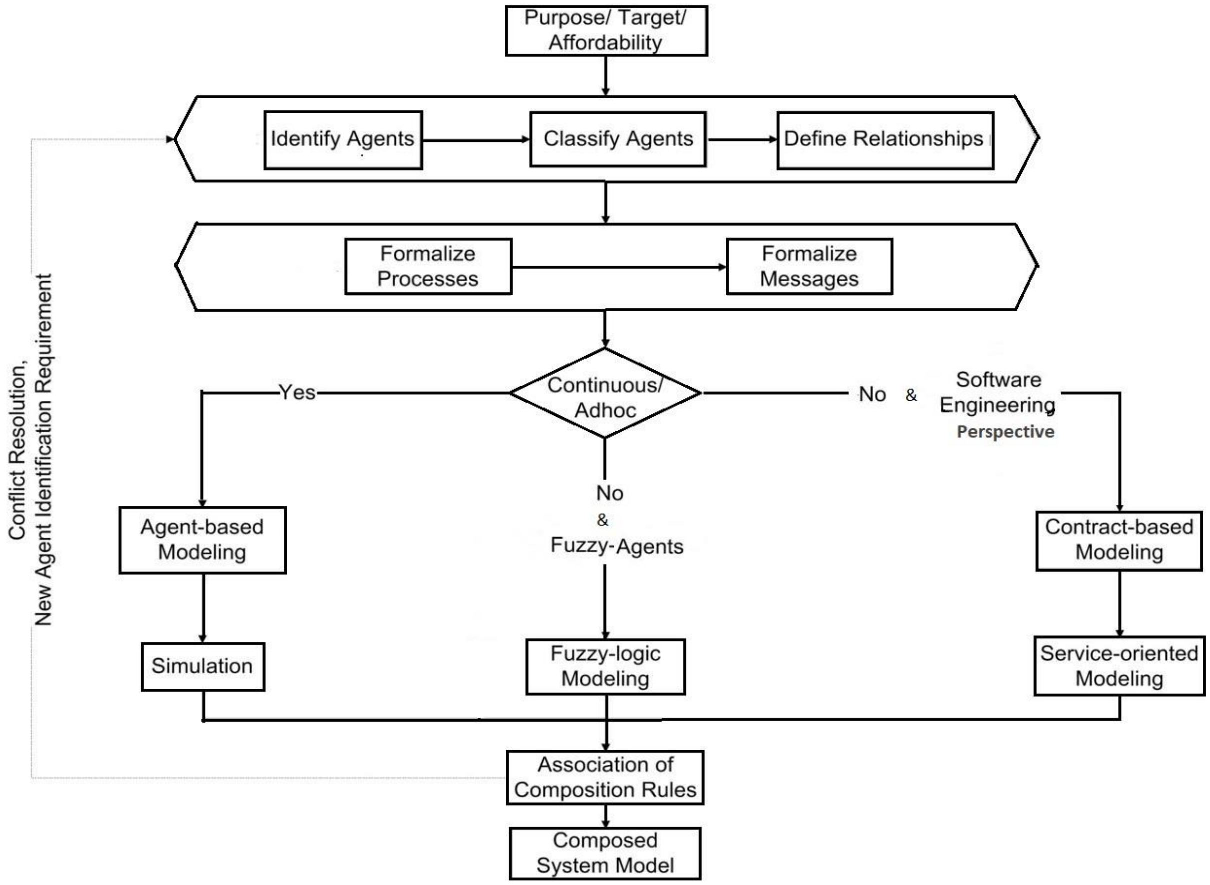


Figure 3.7: Unified Framework for Modeling Service-oriented Internet of Things systems

In Figure 3.8*: Proposed Framework with Different Entities Relations,* we have shown the relations of different objects, processes and agents and their composition towards the development of a model. Agent identification is a common task in almost all the frameworks included in this thesis. The identified agents play certain role in the system hence, the agents have some relation with other agents in the system. When agents collaborate with each other for performing certain task then they will work on the basis of certain contract. The contracts will be designed on the base of the functionalities of the agents. Moreover, every agent will hold some of the types of agents based on its structure and function. Similarly, in service-oriented system the agents will be a consumer or provider of certain service or services. Moreover, certain agents will have the power of decision making. Since, these agents belong to sub-systems, or they are themselves the sub-systems. In any case these sub-systems or agents will be composed to make a system. This composed system will lead towards a detailed model.

The proposed unified framework involves following steps:

Identification-of-Agents: The first step while using these approaches in combination is identification of agents. In this step the agents are identified, and they have assigned any identity as a unique identifier. At this step their functionality is also defined for which the agent is autonomous.

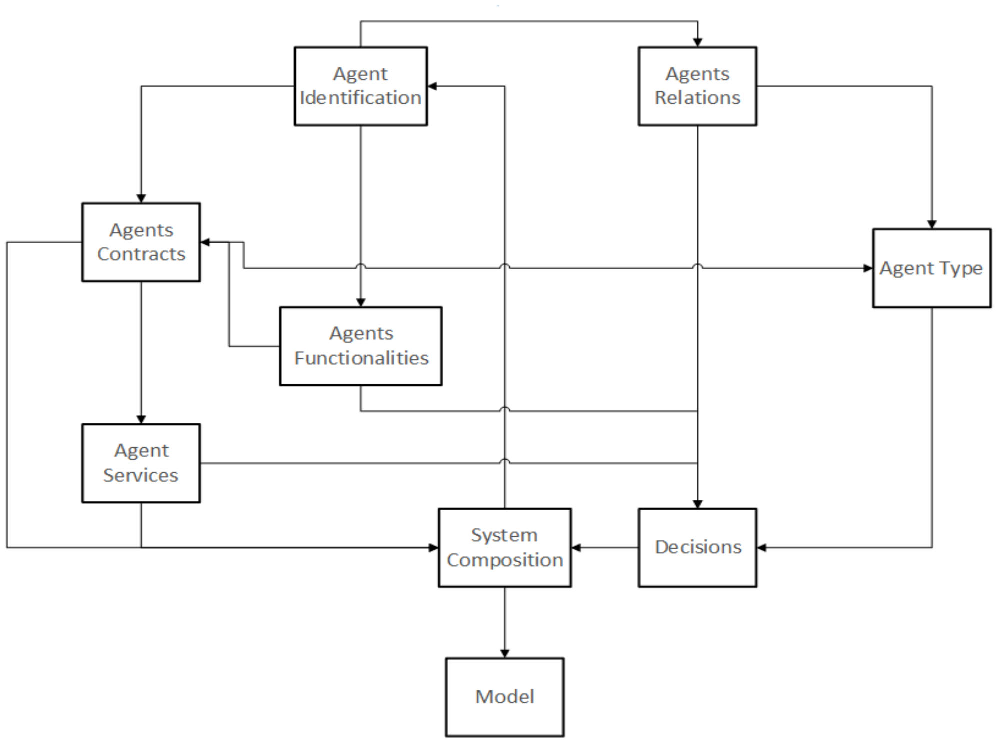


Figure 3.8: Proposed Framework with Different Entities Relations

Classification-of-Agents: Upon the identification of the agent the agent is analyzed based on the properties it owns. After analyzing the agents are assigned different types for the purpose of elaborating their behavior in modeling.

Relations-of-Agents: Different agents in a model have relations with one another. Every agent should have a relation to at least one of its counterparts involved in the systems modeling. These relations may be symmetric or asymmetric. These relations are represented in the form of networks for better understanding.

Formalization-of-Processes: After defining the relations of the agents, the next phase is writing the processes of the agents according to the rules of ambient-oriented modeling. This will help in elaborating the functionality of certain agents and their interaction with other agents. These processes along with relations will be used for rules formation of the model.

Formalization-of-Messages: The messages between the agents will be modeled by using ambient-oriented-modeling symbols. The messages modeling will include the relations of the agents as well. The messages will contain symbols <> and () receiving, along with relations like parent, child, and sibling symbols.

Decision about Continuous/ Ad hoc: If the system or subsystem being modeled is continuous or a specific aspect of a system is to be modeled then agent-based modeling is a better choice. In this case make rules and simulate the model. In other case move to next decision.

Formation-of-Rules for Agent-based Modeling: The rules of the model will be elaborated. Based on the rules the simulations will be developed. The rules are coded in some agent-based simulation tools whereas; visual workflows are developed in others. Some common operations involved in rules will be set-up, creation of agents, interaction of agents, movement of agents, for ambient there will be operations included in rules such as in, out, move and copy.

Simulation: The simulation will represent the behavior of the agents in the system. It provides flexible and dynamic model. It is better for representing specific behaviors of system.

Decision about Software Engineering Viewpoint: If the purpose of modeling the system or subsystem to facilitate the software engineers and application development of IoT systems then better to elaborate the system with using the concepts of service-oriented modeling and contract-based modeling. Otherwise move to next decision.

Elaboration by Using Contracts: Contracts are the on-record requirements and provisions by an agent explaining the way of interaction with other agents. Contract is the conditions against which actions are taken. There are pre-conditions and post-conditions. These pre-conditions and post-conditions explain the behavior of entities. Pre-conditions and Post-conditions may also be termed as assumptions and guarantees. First step is in elaboration by using contracts is that identify the agents associated by contracts. Identify assumptions and guarantees. Visualize the contracts using tools like Attributed Graph Grammar (AGG).

Elaboration by Using Services: When the agents identified are service agents then they are elaborated using the concepts of service-oriented modeling. A service in service-oriented architecture is a plate-form independent module of software accessible by other application. For the purpose of composite modeling, we term the service as a functionality provided by an agent to the other agent or a human. Extensible Markup Language (XML) is commonly used for modeling service-oriented systems. Other languages for modeling services such as WSDL can be used to elaborate services. However, specific service description language for IoT systems is required to properly elaborate IoT services.

Decision about Fuzzy Agent: If the agents contained by system or sub-system are fuzzy agents then fuzzy logic modeling will better elaborate the system. Otherwise, move to the composition of the system.

Association of Composition Rules: After modeling different sub-systems, the sub-systems are composed to represent the complete model of the system.

### 3.5.1 Ontological Completeness

The ontological completeness of the system is checked with four parameters that are: construct overload, construct redundancy, construct excess and construct deficit. We used different modeling approaches in combination. We used bottom up approach. We defined different sub frameworks and then we unified these frameworks. Our constructs are used to model different specific perspective and hence, the construct overload does not exist. Moreover, construct redundancy has also been minimized by defining different frameworks at different levels and then combining them to a unified framework which removes the repetition. As we defined the frameworks with respect to IoTA reference architecture which is a well-known reference architecture it helps to avoid construct excess and construct deficit.

# Chapter 4

# RESULTS, DISCUSSION AND ANALYSIS

To validate our proposed framework and sub frameworks, we applied our framework and sub frameworks to case studies. In this chapter we, provide the results and discussed the frameworks with respect to our research questions.

## 4.1 Modeling Software Engineering Viewpoint of an IoT System

Let us consider scenario of service oriented IoT application in the universities located in twin cities of Pakistan i.e., Islamabad and Rawalpindi for the purpose of collaboration and resource sharing. There are almost twenty-six Higher Education Commission of Pakistan (HEC) recognized universities in twin cities i.e., twenty-one in Islamabad and five in Rawalpindi. Suppose there emerged a consensus among universities to be smart and share resources. To be smart includes, all universities have smart parking, emergency respond systems, smart waste management systems, smart classrooms, and smart air quality monitoring systems. Secondly, these universities have signed a Memorandum of Understanding (MOU) to share their free resources with each other. Now, consider a scenario of this cooperation among universities where Air University Islamabad is organizing a three-day international conference. It is expected that two hundred additional vehicles will arrive university. University has a parking space of just one hundred and twenty additional vehicles. Hence, Air University requires arrangement for this additional parking space. Meanwhile, there are some other universities at a walking distance from Air University i.e., Bahria University Islamabad and National Defense University Islamabad. Bahria University has a free space of two hundred and thirty vehicles. Air university pay rent for the free space and accommodate additional vehicles there. Also, Air University needs some additional garbage collectors due to the arrangements of refreshments and lunch within university as well as increase in number of visitors. NDU can provide the required garbage collectors. Now, these are some services provided by other organizations. The specification of the system is: Higher Education Commission (HEC) a governing body at the top. HEC can add or delete Universities. University can add or remove resources. University can publish resources or call for resources. University can provide feedback after the utilization of resources. Universities can search for Resources in the registry. Resources may be bonded to the Requester/Consumer. When the resources are rented out there exist Contracts for service against that. Payment system for universities and billing of services.

The newly emerged requirements are: Log-in, Log-out, University Management by HEC, Resource Management by Universities and Security. Resources may be different smart devices such as Smart-parking, Trash collecting vehicles, University buses, Smart dustbins, Air Quality Equipment and Cloud Storage. The resources may be composed in sub-systems such as Smart garbage monitoring and collection system, Smart parking system, Smart transportation system and Air quality control system.

### 4.1.1 Decomposition of the System for Agents Identification

In the Inception phase first thing is to identify different components of system. Based on the requirements we draw an abstract representation of our system as shown in Figure 4.1*: Decomposing the System into Three Parts*. In this diagram we decompose the system in three parts. The first is application and business process connected to resources. The second one shows resources as subsystems and the third is aggregator for connection of resources with application. These subsystems include Smart Parking, Smart Buses, Air Quality Control and Smart Garbage Monitoring and Collection. So, we model this system as per decomposition to subsystems i.e., Agent A, Agent B and Agent C. Clearly, this is a heterogeneous system with multiple agents, multiple aspects and multiple consumers and providers. We provide with the help of our framework a simplification in the form of an aggregator which helps us in understanding, identification, communication, and composition.

### 4.1.1 Modeling for Elaboration of Agent *A*

In Elaboration phase we elaborate the identified agents. We treat agent A as a system composed of several agents. First, we have to identify the agents. The agents in this system are

HEC, University, Resource, Accounts, Bill, and Interface. We use contract-based modeling for this system. We used Attributed Graph Grammar (AGG) for contract-based modeling of this system. Figure 4.2*: A Contract and Type-graph of the System Modeled in AGG*, shows a visual-contract in upper part and a type graph of Agent A modeled in AGG using contract-based modeling. The visual contract has pre-condition on Left Hand Side (LHS) and post-condition on Right Hand Side (RHS). When the model in AGG is executed a start graph is generated. Figure 4.*3*: Start-graph of the System Modeled in AGG, shows the start graph of this system. Figure 4.*4*: XML Representation for Information Modeling of System, shows the use of XML for information modeling of a system A. XML may also be useful for modeling the system in hierarchical order.

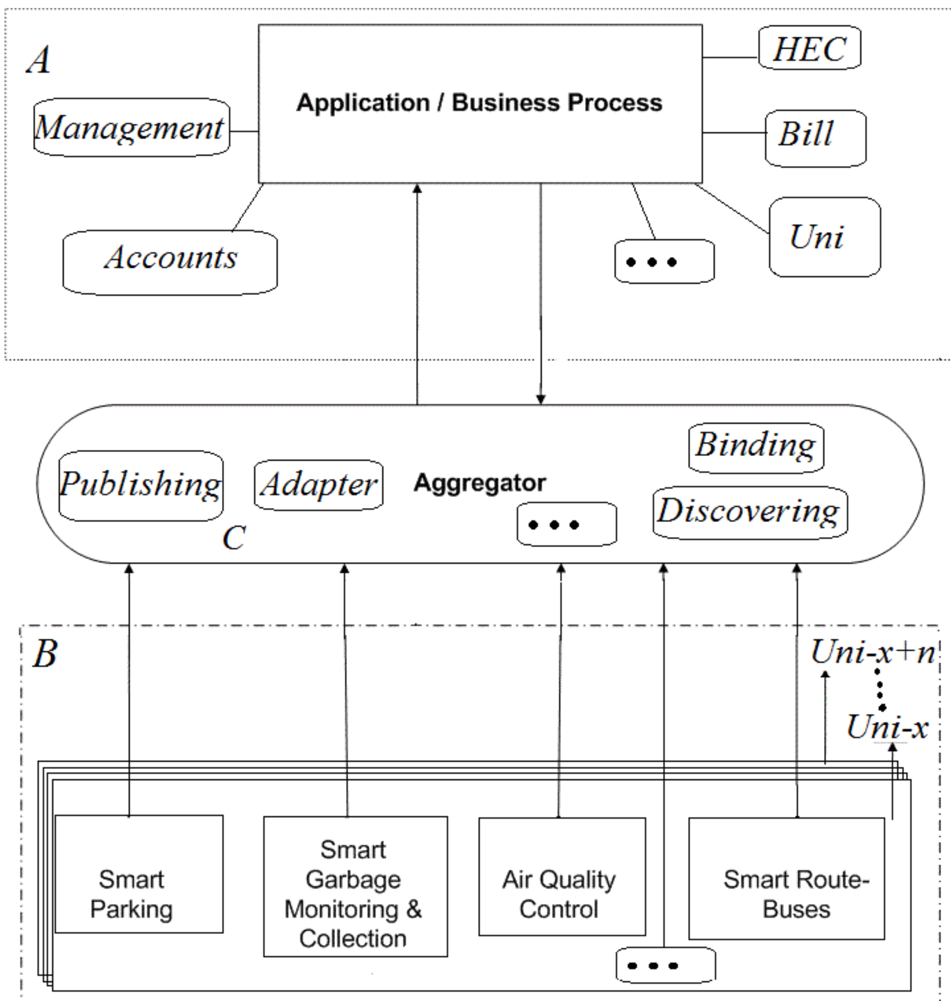


Figure 4.1: Decomposing the System into Three Parts

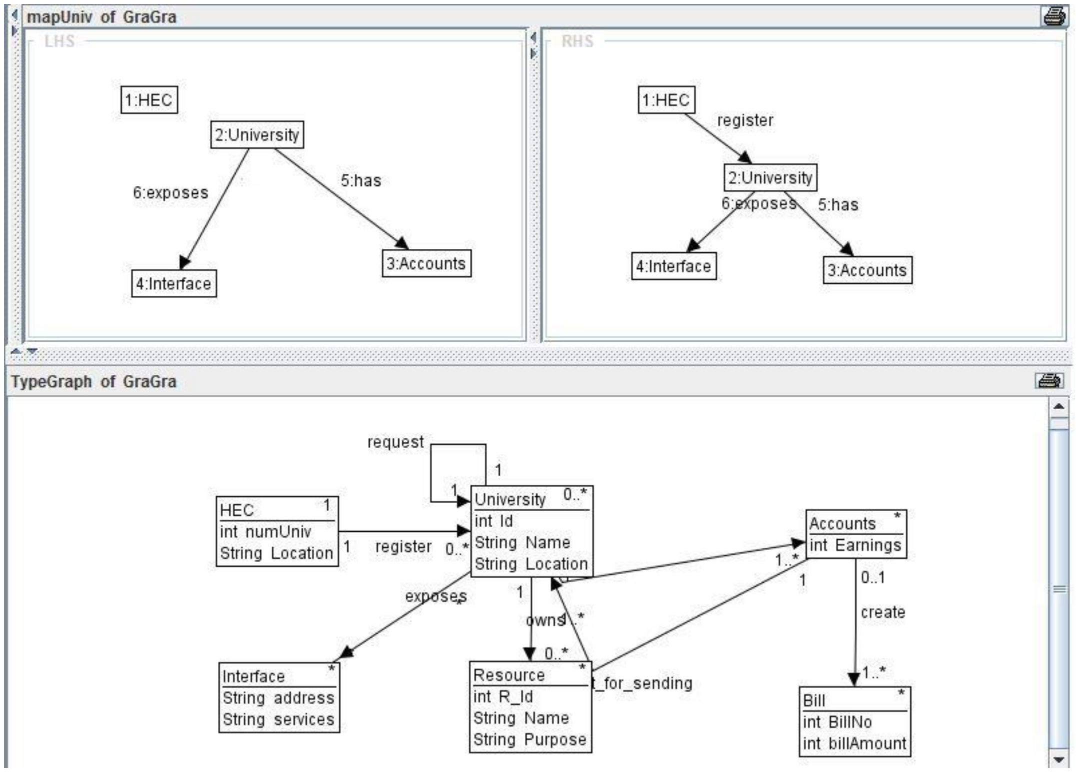


Figure 4.2: A Contract and Type-graph of the System Modeled in AGG

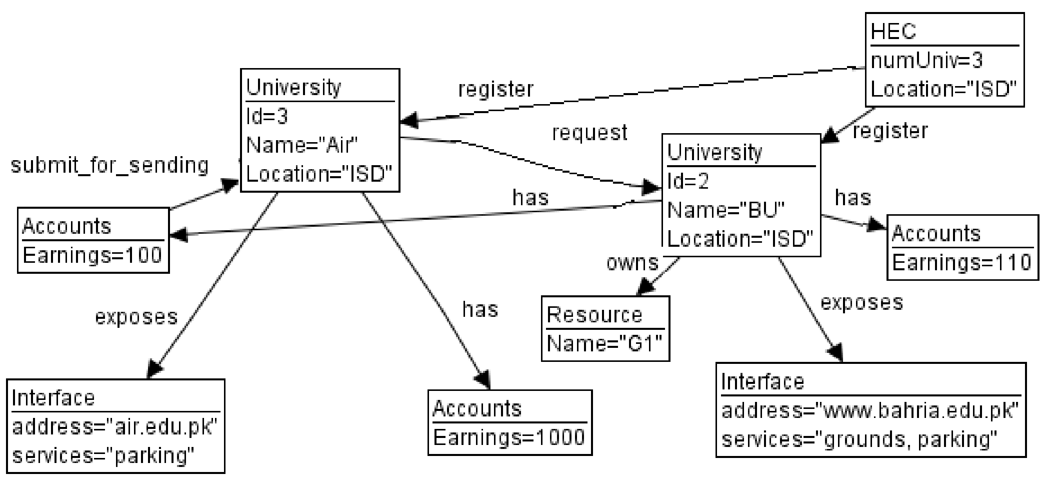


Figure 4.3: Start-graph of the System Modeled in AGG

```
▼<HEC>
  ▼<University category="">
     <Name lang="en"/>
     <Location/>
    ▼<Resource>
       <addResource Name="" Description=""> </addResource>
       <deleteResouce Name="" Identity=""> </deleteResouce>
       <freeResource Name="" Price=""> </freeResource>
     </Resource>
    ▼<accountManagement>
       <addAccount name="" Identity="" Role="" Password=""> </addAccount>
       <deleteAccount Identity=""> </deleteAccount>
     </accountManagement>
     <contract> </contract>
   </University>
  ▼<bill>
     <createBill billNo="" billAmount="" SenderId="" ReceiverId=""/>
     <payBill billNo="" billAmount="" SenderId="" ReceiverId=""/>
   </bill>
  ▼<account>
     <addAccount name="" Identity="" Role="" Password=""> </addAccount>
     <deleteAccount Identity=""> </deleteAccount>
   </account>
 </HEC>
```


Figure 4.4: XML Representation for Information Modeling of System

### 4.1.3 Modeling for Elaboration of Agent *B*

Here we are elaborating agent B. The agent B may be composed of different subsystems. These subsystems contain different devices and provide different functionalists. Here, we are considering four subsystems. These subsystems include Smart parking system, Smart Garbage monitoring and collection system, smart air quality control system and smart route bus system.

#### 4.1.3.1 Elaboration of Smart Parking System

In next level of elaboration, we elaborate the agents identified in elaboration of agent B. Smart parking system involves various sensors, display screens, signal devices, storage, and application. In Figure 4.5: *Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System*(a) shows workflow of smart parking system. Here VDS stands for vehicle detection sensor. Space means vacant positions. Side means the left, right or forward indication on screen where there are multiple paths. These vacant positions are displayed usually on Light Emitting Diodes screen. OHD stands for overhead indicators which are used to identify that the parking slot is empty or full.

Cloud storage represents a service where data obtained from these processes is stored. One state change means that either a parking slot is filled by a vehicle, or a vehicle leaves the parking slot. Filled means that vehicle covers the parking slot and vacant means that vehicle leaves the parking slot. Update means that data is stored on cloud storage.

Figure 4.*6*: WSDL Used to Show Parking Sensors Service, shows the description of different devices of smart parking system as services. It starts with description of vehicle detection sensor. Then overhead display has been described. In last the WSDL describes LED direction screen. VDS only sends a message and OHD displays a message. LED direction screen receives message from VDS and provides direction as output. Figure 4.*7*:WSDL of Cloud Storage Service, shows the WSDL representation of cloud storage.

#### 4.1.3.2 Elaboration of Smart Garbage Monitoring and Collecting System

Smart garbage monitoring and collecting system is used to make the dustbins smart. These smart dustbins should be able to ask for service to vacate them. For the purpose of vacation there may be vehicles for outside placed and persons for in-building placed dustbins.

Figure 4.*5*: Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System; (b), shows the work-flow of smart garbage monitoring and collection system. Here, SDB represents smart dustbin and on state change means dustbin changes from empty to half-filled, or half-filled to full. In-building means that the dustbin is placed inside the building where vehicle can’t reach, and somebody is assigned duty to collect the trash. CP means that the person to whom duty is assigned. CV means trash collecting vehicle. Accept means that one who accepts the duty of collecting trash. Complete means that the trash has been collected. Notify means to notify others who have received the request about the assignment of the duty and send the data to cloud.

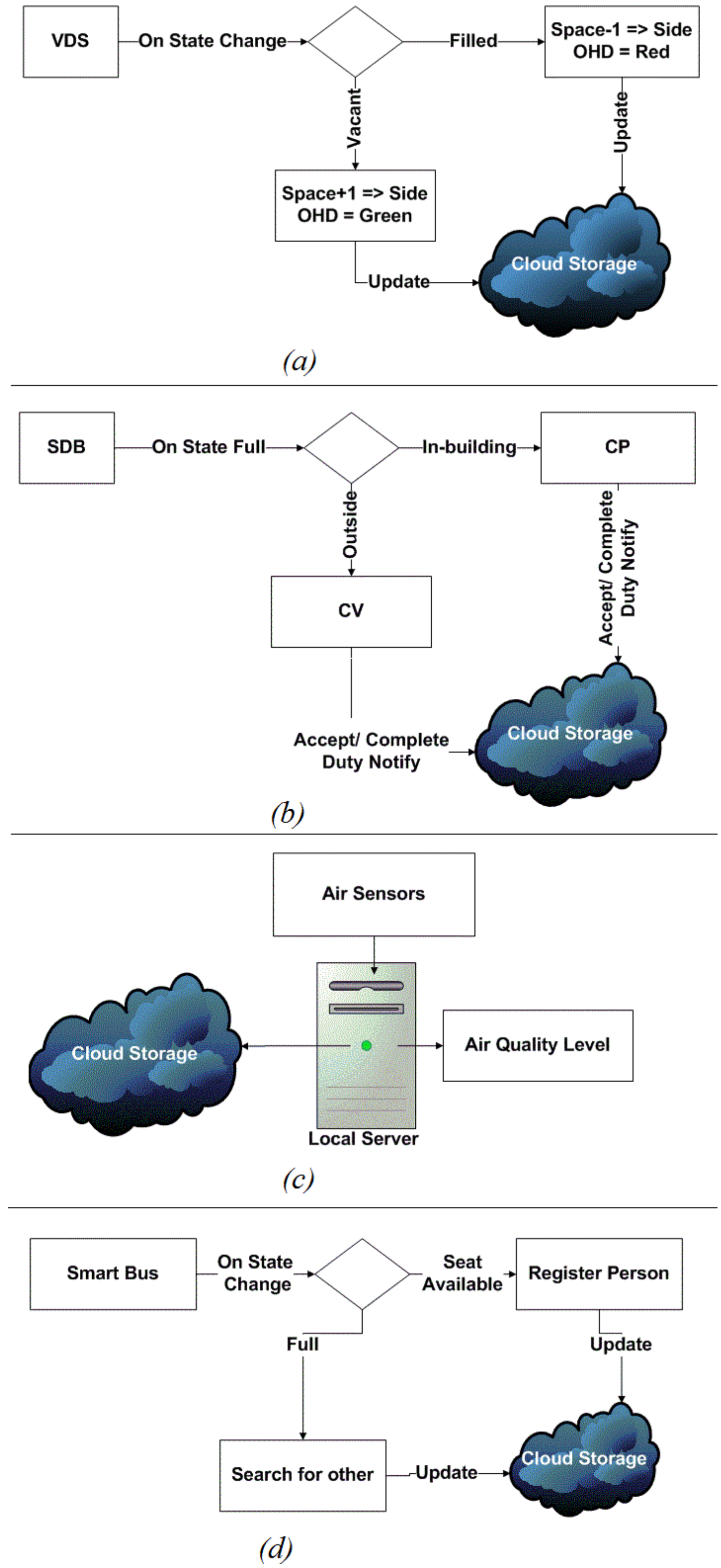


Figure 4.5: Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System

```xml
<Service-Discription-Language>
  <Vehicle-Detection-Sensor>
    <message name="SlotValue">
      <part name="SlotId" type="xs:string"/>
      <part name="Value" type="xs:string"/>
    </message>
    <portType name="VDS">
      <operation name="stateChange">
        <output name="SlotId" message="value"/>
      </operation>
    </portType>
  </Vehicle-Detection-Sensor>
  <Over-Head-Display>
    <message name="SlotValue">
      <part name="SlotId" type="xs:string"/>
      <part name="Value" type="xs:string"/>
    </message>
    <portType name="OHD">
      <operation name="stateChange">
        <input name="SlotId" message="value"/>
      </operation>
    </portType>
  </Over-Head-Display>
  <LED-Directional-Screen>
    <message name="SlotValue">
      <part name="SlotId" type="xs:string"/>
      <part name="Value" type="xs:string"/>
    </message>
    <message name="space">
      <part name="spaceSide" type="xs:string"/>
      <part name="Value" type="xs:string"/>
    </message>
    <portType name="LDS">
      <operation name="stateChange">
        <input name="SlotId" message="value"/>
        <output name="spaceSide" message="value"/>
      </operation>
    </portType>
  </LED-Directional-Screen>
</Service-Discription-Language>
```

Figure 4.6: WSDL Used to Show Parking Sensors Service

```
<Cloud-Storage>
  <message name="System-Id">
    <part name="SystemId" type="xs:string"/>
    <part name="Value" type="xs:string"/>
  </message>
  <portType name="storage">
    <operation name="Store">
      <input name="SystemId" message="value"/>
    </operation>
  </portType>
  <message name="System-Id">
    <part name="SystemId" type="xs:string"/>
    <part name="Value" type="xs:string"/>
  </message>
  <message name="Reply">
    <part name="RetId" type="xs:string"/>
    <part name="Value" type="xs:string"/>
  </message>
  <portType name="storage">
    <operation name="Retrieve">
      <input name="SystemId" message="value"/>
      <output name="RetID" message="value"/>
    </operation>
  </portType>
</Cloud-Storage>
```

Figure 4.7:WSDL of Cloud Storage Service

#### 4.1.3.3 Composition of Smart Garbage Monitoring and Collecting System

We consider SDB, CP, CV and CloudStorage as agents. While modeling this system using our proposed framework after modeling agents as services as described in Figure 4.5: Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System, we compose the system using Aspect-oriented composition rules. So, the composition rules will be as following:

1. SDB >> CP
2. SDB >> CV
3. CV >> CloudStorage
4. CP >> CloudStorage

According to the rules above, there are two paths that connect elements from SDB to

CloudStorage. First one is SDB −CP −CloudStorage and second one is SDB − CV −CloudStorage. So, every agent is connected to at least one other agent and there doesn't exist any conflict.

#### 4.1.3.4 Elaboration of Smart Air Quality Monitoring System

Smart air quality monitoring system uses sensors to detect the level of harmful gases and oxygen in the air. Different sensors are installed at different positions and then accumulated air quality is measured. Figure 4.5: Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System; (c), shows work-flow diagram of an air quality monitoring system. In this system the air sensors send information to local server and the local server processes the information. After processing the information, the data is stored in the cloud and also, the information is displayed to the people.

Due to the limitation of space, we are not repeating the service-oriented, contract-based and aspect-oriented approach for this sub-subsection and the next one.

#### 4.1.3.5 Elaboration of Smart Route Bus System

Smart route bus system helps the registered students to track the bus and time to reach the desired stop. Hence, the workflow of the registration system of the bus is shown in Figure 4.5: Work-flow Diagrams (a) Smart Parking (b) Smart Garbage (c) Smart Air Quality Monitoring System (d) Smart-Route Bus System; (d). Smart Bus notifies when a registered student leaves or a new student is registered. So, if after the notification there are number of registered persons is less than total capacity of the bus than new person can be registered. In other condition where the bus is already full someone interested to register on specific route will search for other buses. This information is updated on cloud storage.

### 4.1.4 An Ad hoc and Flexible Representation of System

Here, the system is elaborated with the help of simulation tools and flexibility of change is provided. While modeling the system there may be a need of an abstract level model. This abstract level, ad hoc and continuous time models of the system or subsystems and devices may be developed using Agent-based modeling.

*Figure 4.8: Netlogo Model Showing an Ad hoc Model where the "Number of Universities" is Flexible*, shows an ad hoc and abstract representation of the system in the form of simulation implemented in Netlogo. This simulation creates a turtle HEC in the setup. Then universities are attached to the system and the number of universities may be selected from the slide-bar. The slide-bar provides flexibility in selecting the number of universities. In next step resources are added and randomly attached to the universities. Here we create two resources per university at this step. In the last step "Run" button may be used to change the links to resources to the universities i.e., showing service provision.

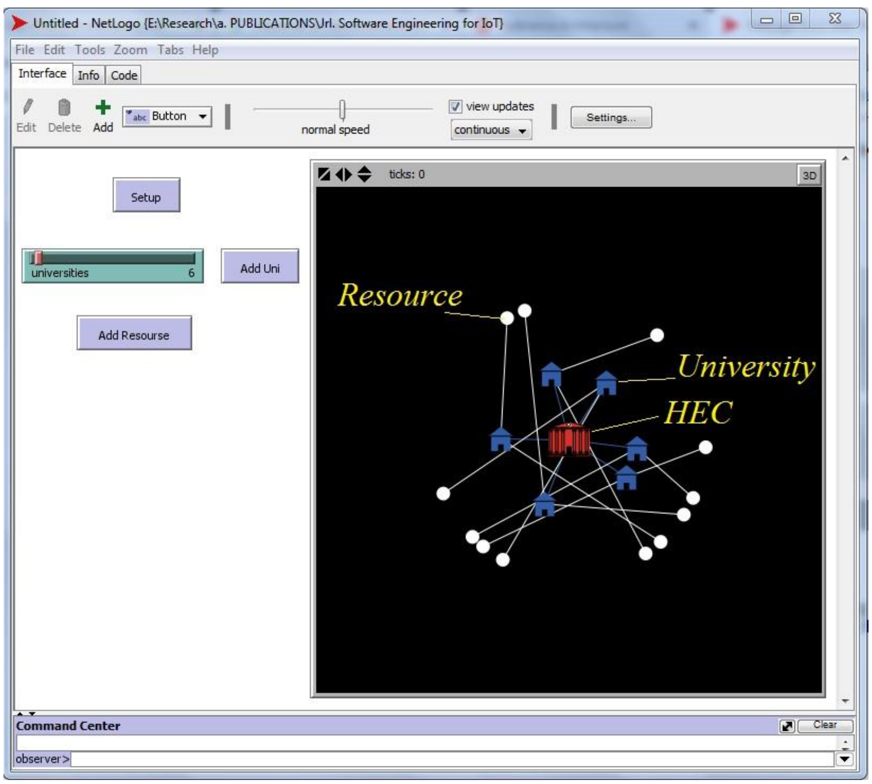


Figure 4.8: Netlogo Model Showing an Ad hoc Model where the "Number of Universities" is Flexible

### 4.1.5 Composition of System using Composition Rules

In the composition phase we compose the systems from the elaborated subsystems and identify new agents if required. The last phase of our framework is associating the composition rules. However, what we find from modeling of agent B after decom-position is that CloudStorage is common in multiple subsystems. Thus, it’s better to treat the CloudStorage as a fourth agent with the previous three i.e. agent A, agent B and agent C. Now, the relation between these four agents will be:

1. A||C
2. B||C
3. CloudStorage||C

The connection of the agents is A−C−B, A−C−CloudStorage, B−C−CloudStorage that A is directly connected or has direct link with B and B is connected with CloudStorage through aggregator C. It shows that all the agents are connected.

## 4.2 Analysis and Discussion on Modeling Applications in IoT

Reference architectures are abstract architectures of systems, and they may differ from actual system [235]. IoT-A reference architecture provided a way to split an IoT system in different components. This architecture provided domain model, information model, functional model and communication model. The functional model of IoT-A reference architecture includes different layers such as device, communication, security, management, service organization, service, virtual entity, business process and application. While analyzing these layers it seems that layers such as application, business process, virtual entity, service organization and security involve major role of Software Engineer during design and development.

The industrial internet reference architecture (IIRA) focuses on internet of things with respect to industrial aspect [236]. It takes in account four viewpoints i.e. Business, Usage, Functional and Implementation Viewpoint. It also, discusses the cross-cutting concerns and characteristics of the system. The functional viewpoint has seven different domains such as Controls, Operations, Information, Application, Business, Functional and Crosscutting functions. The control domain contains the functions related to control systems. The operation domain functions include management, monitoring and optimization. The information domain includes data related functions. Application domain is same as application layer of IoT-A reference architecture which contains application logic. Business domain has almost similar function as business process layer of IoT-A reference architecture. Crosscutting domain includes the function like connection. Functional domain contains the technologies which support IoT systems such as cloud computing.

Reference Architecture Model for Industrie 4.0 (RAMI 4.0)) is also a reference architecture for industrial internet of things [237]. This architecture is based on service-oriented concepts and has three dimensions i.e., Layers, Life cycle & Value Stream and Hierarchy Levels. The layers of RAMI 4.0 are Business, Functional, Information, Communication, Integration and Assets. The Business layer of RAMI 4.0 is similar to the business process layer of IoT-A where business models are mapped with processes. Functional layer contains the rules and decision-making logic and, services are also modeled at this layer. The information layer contains data to represent models based on formally described rules and execution of event related rules. Communication layer is similar as in IoT-A reference architecture. Integration layer provides interaction with human as well as interaction of devices with each other. Asset layer is similar to device layer of IoT-A reference architecture where there are physical entities. Life cycle & Value Stream and Hierarchy Levels are about the industry processes.

In [238], Service-oriented architecture has been used in-combination to SoaML for solving the heterogeneity issues. In [239], SysML has been used for modeling internet of things applications and to deal with system engineering problem. In [240], IoT-A reference architecture, model driven architecture and separation of concerns have been used in-combination for proposed framework. The framework is effective for Quality-of-Service attributes management in early stages of modeling. Both horizontal and vertical perspective have been considered by using the principle of Separation of Concerns.

Service-oriented approach has been adopted in all reference architectures and also in above cited articles. The principle of Separation of Concerns has been used in IIRA as well as in [240]. Our framework uses four modeling approaches i.e., Service-oriented modeling, aspect-oriented modeling, agent-based modeling, and contract-based modeling. When services are provided then there exists an agreement between service provider and service consumer. The agreement is a set of assumptions and guarantees. Contract-based modeling provide us a way to model system with pre-conditions or assumptions and post-conditions or guarantees. In internet of things there are smarter devices on nodes which are autonomous as well. Autonomous and smarter devices can be treated as agents. Agent-based modeling is useful while modeling random behaviors and ad hoc systems. Hence, we used these four modeling approaches in combination.

In software engineering the domain model is used to represent context information. Agent-based model can be used for the representation of domain model. Our framework can be used for domain models of IoT systems as shown in case study. Contracts can be used for modeling of specification since our framework provides a mechanism for specification modeling as well. The modularization and composition of system is a part of design model. Also, the attributed graph grammar (AGG) is used to represent the design model. Service-oriented modeling is also a part of design modeling. Hence, our framework provides a mechanism for modeling IoT based software system from different aspects.

### 4.2.1 WSDL to TSDL

For service description of things, we need a standard language. The new language can be an extension of web service description language. Here we provide a possible extension of a WSDL for Internet of Things which we name Things Service Description Language (TSDL).

**Element for TSDL (** < message > **)**

< messagename = ”request − for−the−service” >
< partname = ”term”type =”xs : string”/ >
< /message >
< messagename = ”location−o f −thing” >
< partname =”value”type = ”xs : string”/ >
< /message >
< messagename = ”communication− response − f or −the − service” >
< partname = ”value”type = ”xs : string”/ >
</message >
The element < *message* > defines the name of all messages and data types used by these messages.

**Element for TSDL (** < portType > **)**

< portTypename = ”Thing−service−name−or−thing−id” >
< operationname =”ThingOperation” >
< requestmessage = ”request − f or−the−service”/ >
< phyResponsemessage”action− perform−by−thing”/ >
< locationmessage = ”location−o f −thing”/ >
< comResponsemessage = ”communication − response − for − the − service”/ >
</operation >< /portType >

In “portType” element name will be the thing service name or service name for example garbage collection service is the name of the service for garbage collection in city. “ThingOperation” is the name of operation that define by “portType” element. For example, collect garbage is the operation of smart garbage collection system. Through < request > element proposed to use for request for the service of things for example someone request to smart garbage system to collect garbage from my house.

< *phyResponse* > element proposed for action performance after receiving request for service message for example garbage collection van driver trace your location and move towards home where someone request. < *location* > of thing is the very important element for IoT system because on the base of location of both requester and responder action will be performed. < *comResponse* > element proposed for the communication response from thing to requester after performing physical response. There are four types of operations in WSDL, that are:

1. One-way
2. Request-response
3. Solicit-response
4. Notification

But in case of IoT there is physical existence of things therefore, we defined an extra element < *phyResponse* > within < *operation* > element with < *request* > element and < *comResponse* > element.

### 4.2.2 SOAP to SoTAP

To access services provided by things, Simple object access protocol (SOAP) also needs to be modified. Here we provide possible extension with the name Service-oriented Thing Access Protocol (SoTAP).

**Envelope for SoTAP**

*< sotap : Envelopexmlns : sotap = ”asperpublication/”sotap : encodingStyle =”asperprovide” > ...*
*< /soap : Envelope >*

**Header for SoTAP**

*< sotap : Header >*
*<tid : ThingID > xmlns : tid = ”thingid/”sotap : mustUnderstand = ”1” > 234 < /tid : ThinID >*
*<tow : Thingowner > xmlns : tow = ”owner−name/” </tow : Thingowner >*
*<tre : Thingrequester > xmlns : tre = ”requesrer−name/id/” </tre : Thingrequester >*
*< /soap : Header >*

**Body for SoTAP**

*< sotap : Body >*
*< m : Getservice−namexmlns : m = ”thing−id/service−repository” >*
*< m : services > request − service < /m : service >*
*< /m : Getservice − name >*
*< /sotap : Body >*

## 4.3 Modeling IoT System having Fuzzy-agent in Composition

Let us take the scenario of a smart classroom. The building has no central air conditioning system rather air conditioning is decentralized. We want to automatize the heating and cooling system of a classroom. We consider a room just like classrooms of “Sir Syed Block” at “Bahria University E-8 Islamabad” campus. We assume that the classroom has two fans, a window air-conditioner for cooling and a heater for heating. First of all, we will determine that what are the parameters on the basis of which a user decides for switching fans, air-conditioner and heater. The first parameter is heat, second is humidity and the third is number of people in the room. Number of people is considered as a key variable in classroom because average body temperature of human is approximately 37 degrees centigrade. Whereas comfortable room temperature for human is 25 degrees centigrade. Hence, as the number of people in a room increase the average temperature of room also increases.

The general view of fuzzy inference system using MATLAB Fuzzy-logic Toolbox is shown in Figure 4.9: General View of Fuzzy Inference System Simulated in MATLAB Fuzzy-logic Toolbox. There are three input variables i.e., Humidity, Temperature and Number-of-People and four output variables i.e., AC-Cool, Fan-1, Fan-2 and Heater. Humidity normal 30 percent to 60 percent.

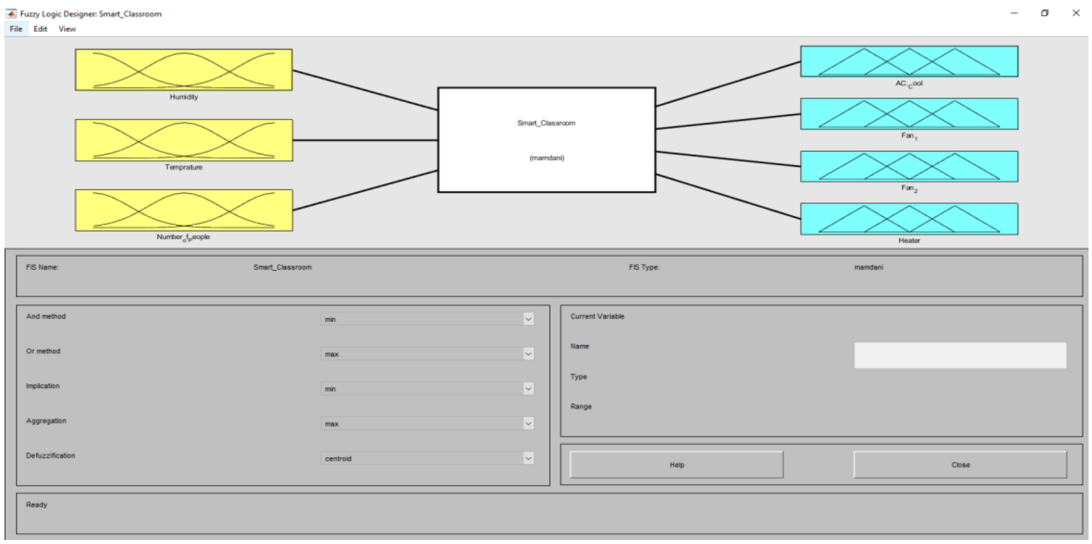


Figure 4.9: General View of Fuzzy Inference System Simulated in MATLAB Fuzzy-logic Toolbox

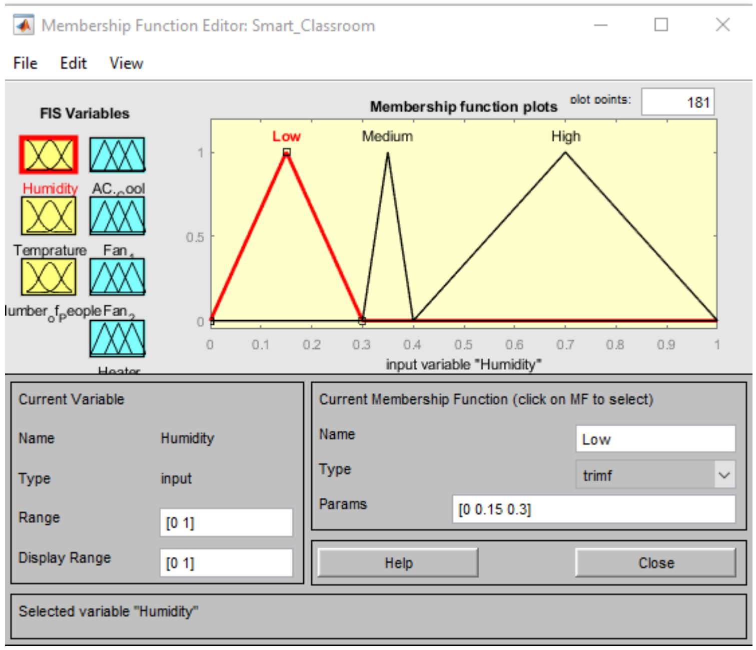


Figure 4.10: Input Variable Humidity

### 4.3.1 Input Variables

There are three input variables of this fuzzy inference system. These variables are Humidity, Temperature and Number-of-People present in the smart room. Humidity is based on percentage. So, the percentage can be divided into three fuzzy member functions i.e., low, medium, and high. Temperature normally ranges from -10 degree to 60 degree centigrade. So, Temperature below 18 degree is considered as low. Temperature ranging from 18 degree to 35 degrees is considered as medium and above 35 degrees is considered as average. Similarly, in a room where capacity is 50 persons, we take 0 to 10 persons as minimum, 10 to 35 as average and above 35 as maximum. Based on this we implemented these input variables in MATLAB fuzzy logic toolbox.

$$Humidity_{Low} \cong \begin{cases} 1, & x \geq 1 \\ 29, & x \leq 29 \end{cases} \quad (4.0.1)$$

$$Humidity_{Medium} \cong \begin{cases} 30, & x \geq 30 \\ 60, & x \leq 60 \end{cases} \quad (4.0.2)$$

$$Humidity_{High} \cong \begin{cases} 61, & x \geq 61 \\ 100, & x \leq 100 \end{cases} \qquad (4.0.3)$$

$$Temperature_{Low} \cong \begin{cases} 0, & x \geq 0 \\ 18, & x \leq 18 \end{cases} \qquad (4.0.4)$$

$$Temperature_{Medium} \cong \begin{cases} 19, & x \geq 19 \\ 35, & x \leq 35 \end{cases} \qquad (4.0.5)$$

$$Temperature_{High} \cong \begin{cases} 36, & x \geq 36 \\ 100, & x \leq 100 \end{cases} \qquad (4.0.6)$$

$$NumberOfPeople_{Minimum} \cong \begin{cases} 0, & x \geq 0 \\ 10, & x \leq 10 \end{cases} \qquad (4.0.7)$$

$$NumberOfPeople_{Average} \cong \begin{cases} 11, & x \geq 11 \\ 35, & x \leq 35 \end{cases} \qquad (4.0.8)$$

$$NumberOfPeople_{Maximum} \cong \begin{cases} 36, & x \geq 36 \\ 50, & x \leq 50 \end{cases} \qquad (4.0.9)$$

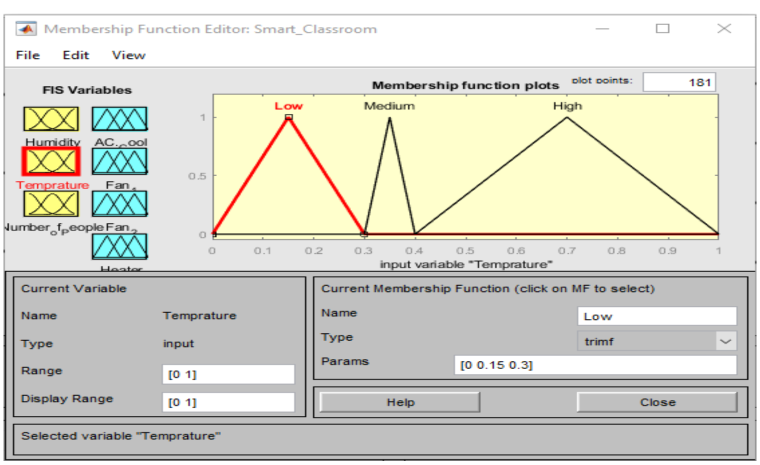


Figure 4.11:Member functions of Variable Temperature

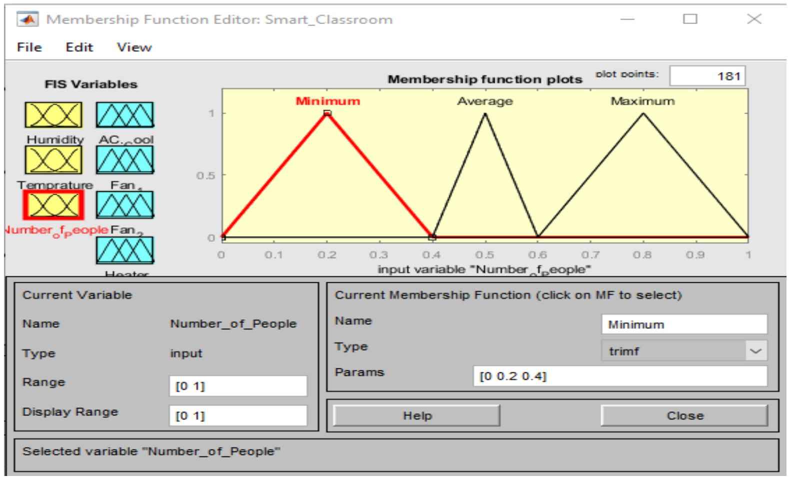


Figure 4.12: Member functions of Variable Number-of-People

Figure 4.*10*: Input Variable Humidity, shows the membership functions representations of the input variable Humidity. The member functions include low, medium, and high. This variable is based on fuzzy logic, and we take fuzzy values. The graphical representation shows the range of this input variable. Figure 4.*11*:Member functions of Variable Temperature, shows the membership functions representations of Temperature. The member functions include low, medium, and high. Figure 4.*12*: Member functions of Variable Number-of-People, shows the membership functions representations of input variable Number-of-People. The member functions include Minimum, Average and Maximum.

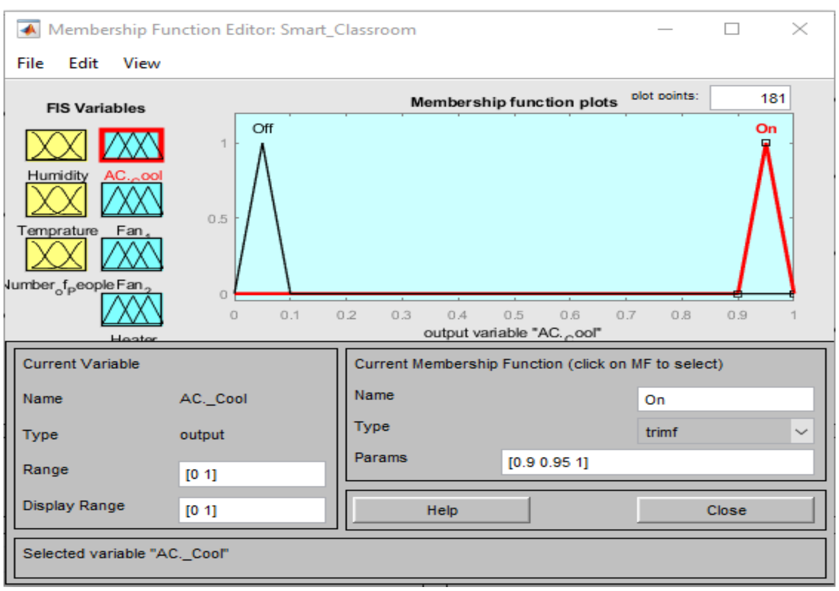


Figure 4.13: Output Variable AC-Cool

### 4.3.2 Output Variables

There are four output variables of this fuzzy inference system. These variables are AC-Cool, Fan-1, Fan-2 and Heater. We consider all these output variables as binary variables, which may either be on or off. Although there may be different in-between values of these variables but to reduce the complexity of the systems, we take binary only.

Figure 4.*13*: Output Variable AC-Cool, shows the membership functions representations of the output variable AC-Cool. The member functions include on and off as the Cooling will either be on or off and it's a binary function variable. Figure 4.*14*: Output Variable Fan-1, shows the membership functions representations of the output variable Fan-1. The member functions include on and off as the Fan will either be on or off and it's a binary function variable. Figure 4.*15*: Output Variable Fan-2, shows the membership functions representations of the output variable Fan-2. The member functions include on and off as the Fan will either be on or off and it's a binary function variable. Figure 4.*16*: Output Variable Heater, shows the membership functions representations of the output variable Heater. The member functions include on and off as the Heater will either be on or off and it's a binary function variable.

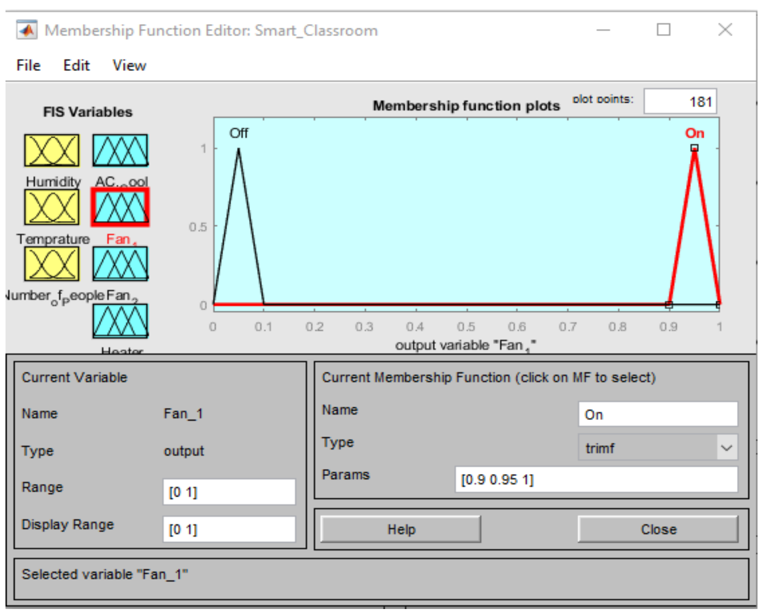

Figure 4.14: Output Variable Fan-1

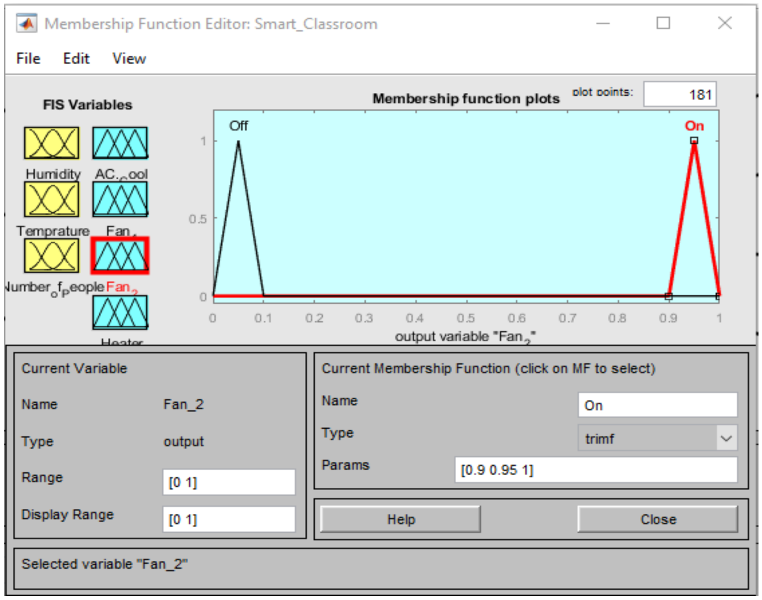


Figure 4.15: Output Variable Fan-2

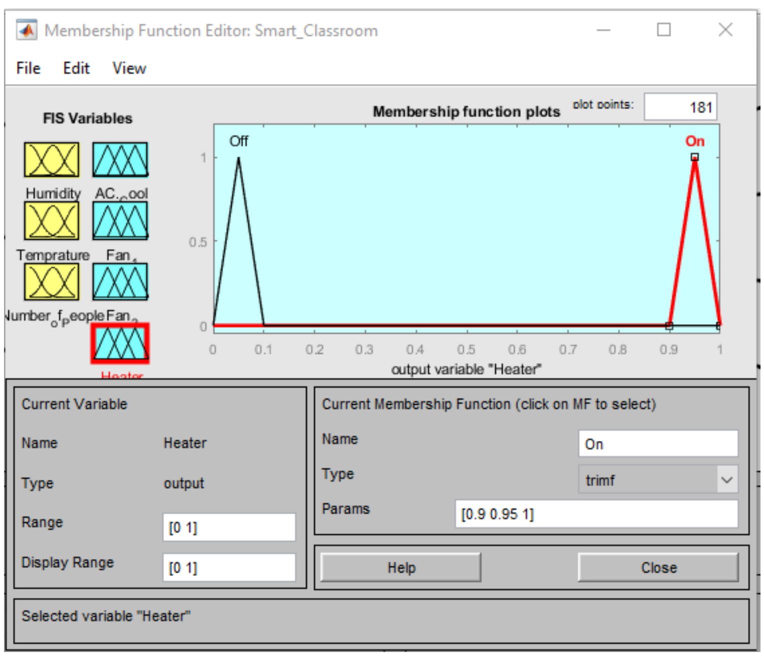


Figure 4.16: Output Variable Heater

### 4.3.3 Rules of Fuzzy Inference System

As there are three input variables and every variable has three member functions. So, we have 3x3 rules means 27 rules. In Figure 4.*17*: Textual Representation of Rules Part-1, the rules from Rule number 1 to Rule number 20 are shown in textual form. In the lower portion of this figure the input variables i.e., humidity, temperature, and number of people along with their respective member functions are shown. Below these variables there are connections of the variable i.e. AND & OR. Also, every variable has a not checkbox. If not, checkbox is selected then it will negate the member function values of certain variable. Then there is weight which may be allocated to variable in any rule. And then there are buttons which provide the functionality of adding new rule, deleting an existing rule, or updating any rule. In Figure 4.*18*: Textual Representation of Rules Part-2, the rules from Rule number 8 to Rule number 27 are shown in textual form. In the lower portion of this figure on the right of “Then” there are output variable and their member functions. In Figure 4.19: Graphical Representation of Rules, the rules and their relations along with their member functions and ranges is shown graphical form.

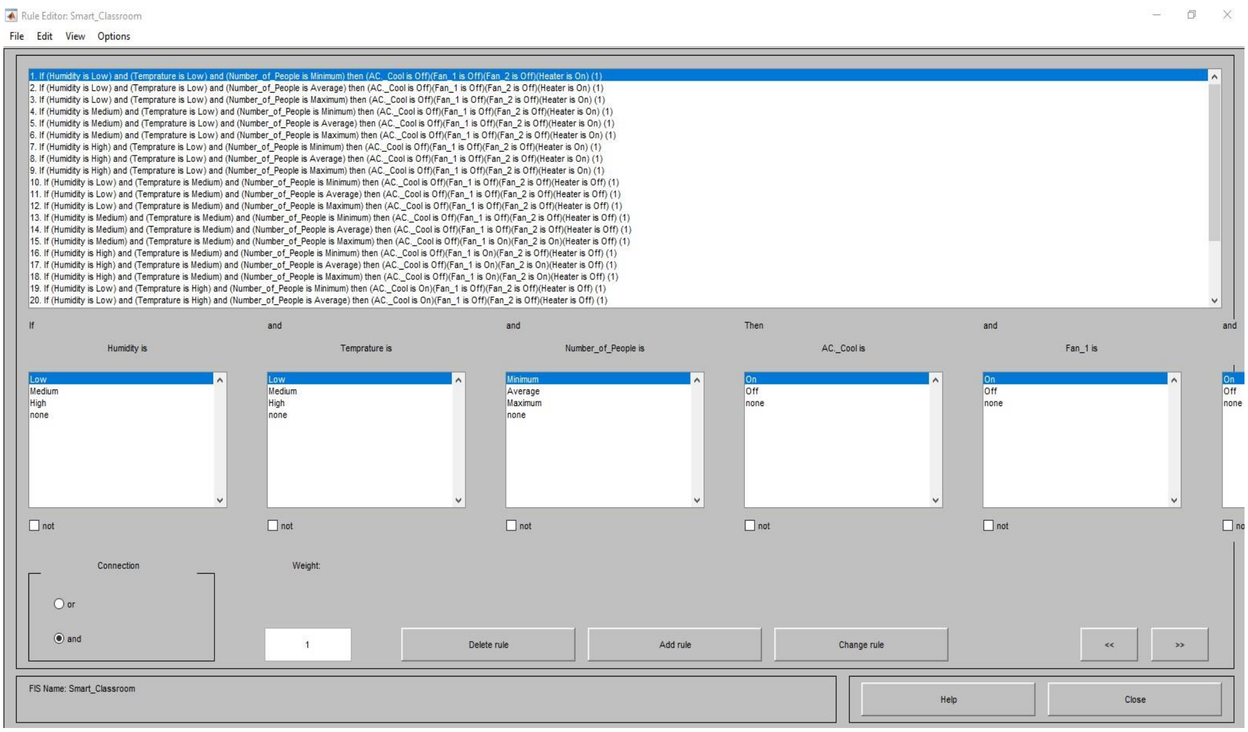


Figure 4.17: Textual Representation of Rules Part-1

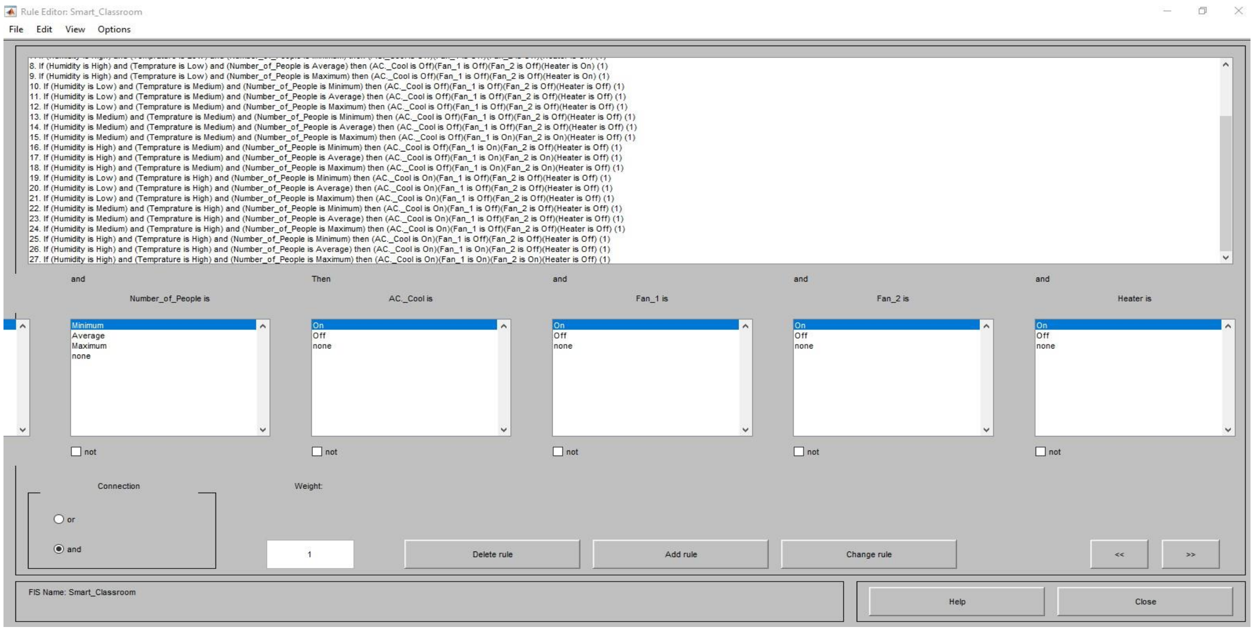


Figure 4.18: Textual Representation of Rules Part-2

### 4.3.4 2-D Representations

Relations between different variables show dependencies. 2-D graphical representation shows relation between two variables. The first variable is input, and the other variable is output. In other words, by 2-D representation one can understand the proportionality of one input with one output variable. Figure 4.*20*: Relation Between Humidity and AC-Cool, shows relation between one input variable i.e., Humidity and one output variable i.e., AC-Cool. Figure 4.*21*: Relation Between Humidity and Fan-1shows relation between one input variable i.e., Humidity and one output variable i.e., Fan1. Figure 4.*22*: Relation Between Humidity and Fan-2, shows relation between one input variable i.e., Humidity and one output variable i.e., Fan2. Heater shows relation between one input variable i.e., Humidity and one output variable i.e., Heater. Figure 4.*24*: Relation Between Temperature and AC-Cool, shows relation between one input variable i.e., Temperature and one output variable i.e., AC-Cool. Figure 4.*25*: Relation Between Temperature and Fan1, shows relation between one input variable i.e., Temperature and one output variable i.e., Fan1. Figure 4.*26*: Relation Between Temperature and Fan2, shows relation between one input variable i.e., Temperature and one output variable i.e., Fan2. Figure 4.*27*: Relation Between Temperature and Heater, shows relation between one input variable i.e., Temperature and one output variable i.e. Heater.

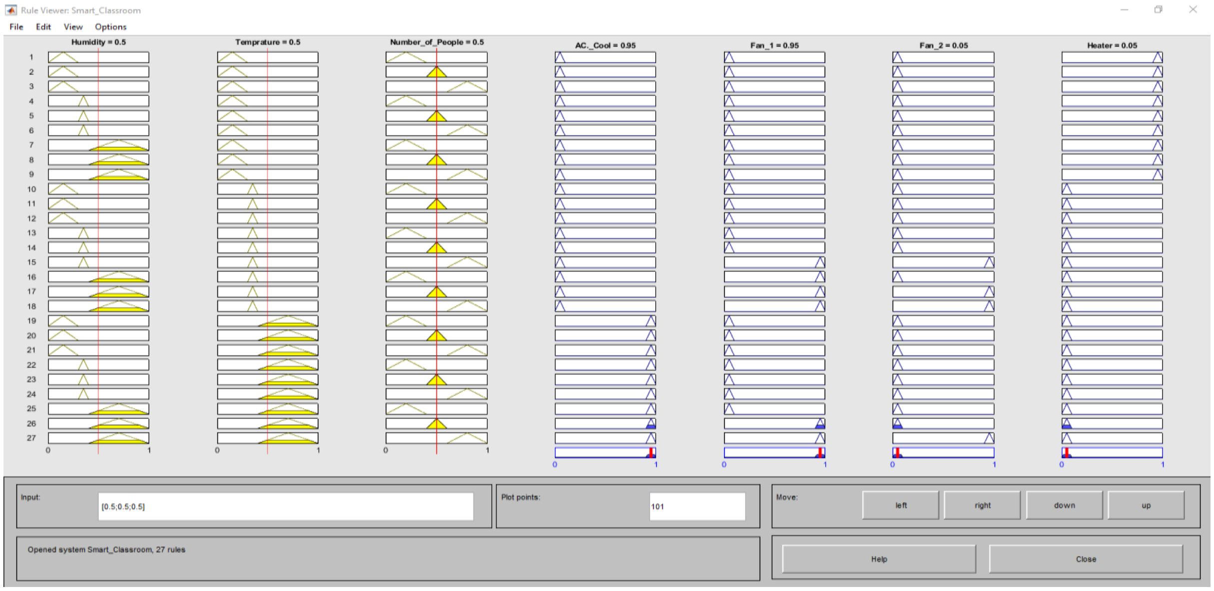


Figure *4*.19: Graphical Representation of Rules

Figure 4.28: Relation Between Number-of-People and AC-Cool, shows relation between one input variable i.e., Number-of-People and one output variable i.e., AC-Cool. Figure 4.*29*: Relation Between Number-of-People and Fan1, shows relation between one input variable i.e., Number-of-People and one output variable i.e., Fan1. Figure 4.*30*: Relation Between Number-of-People and Fan2, shows relation between one input variable i.e., Number-of-People and one output variable i.e., Fan2. Figure 4.*31*: Relation Between Number-of-People and Heater, shows relation between one input variable i.e., Number-of-People and one output variable i.e., Heater.

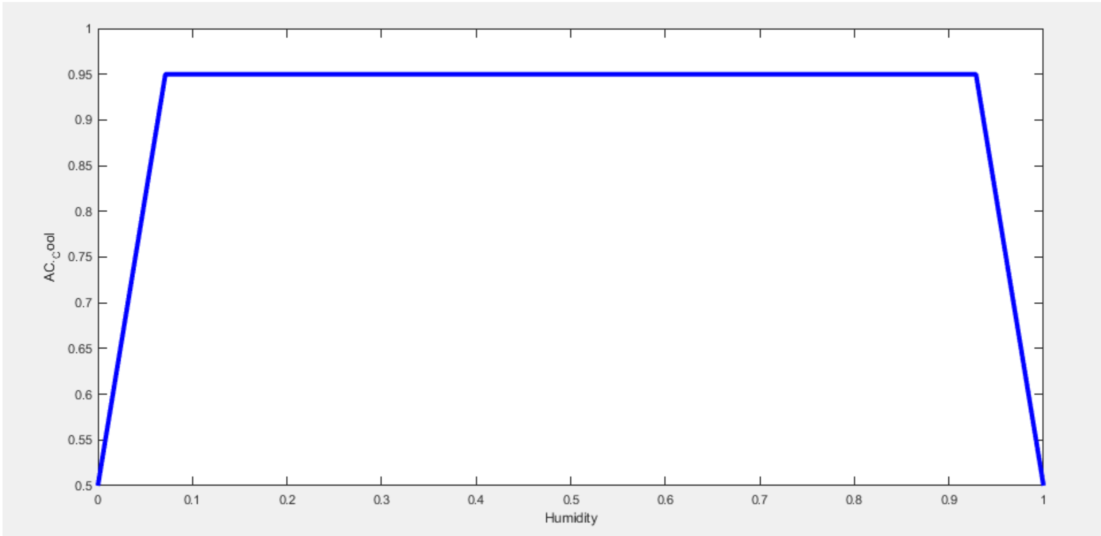


Figure 4.20: Relation Between Humidity and AC-Cool

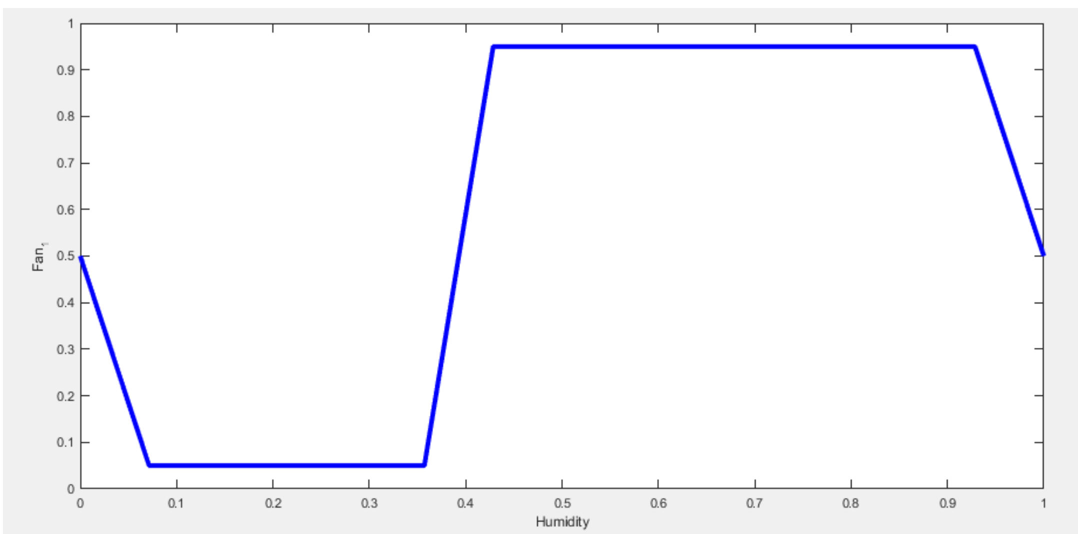


Figure 4.21: Relation Between Humidity and Fan-1

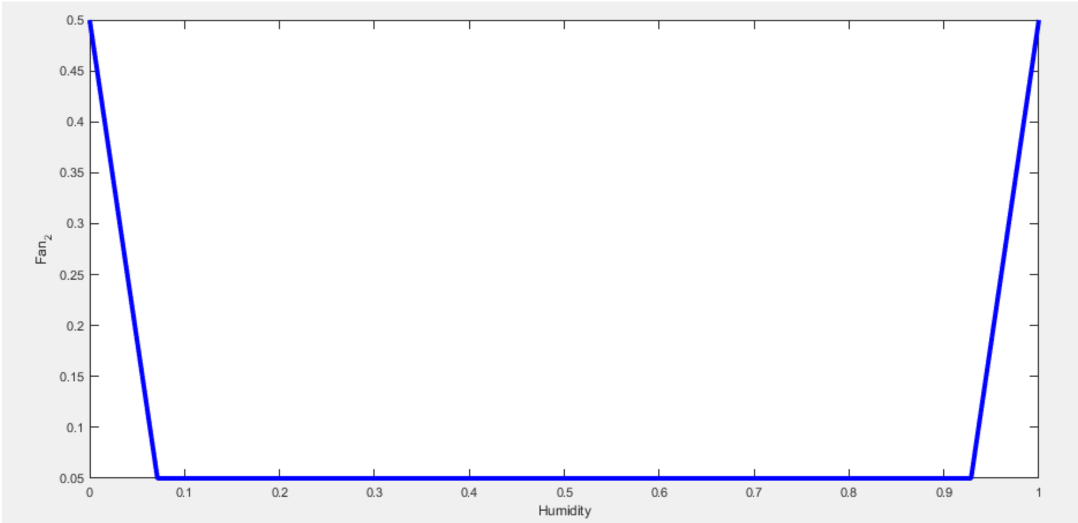


Figure 4.22: Relation Between Humidity and Fan-2

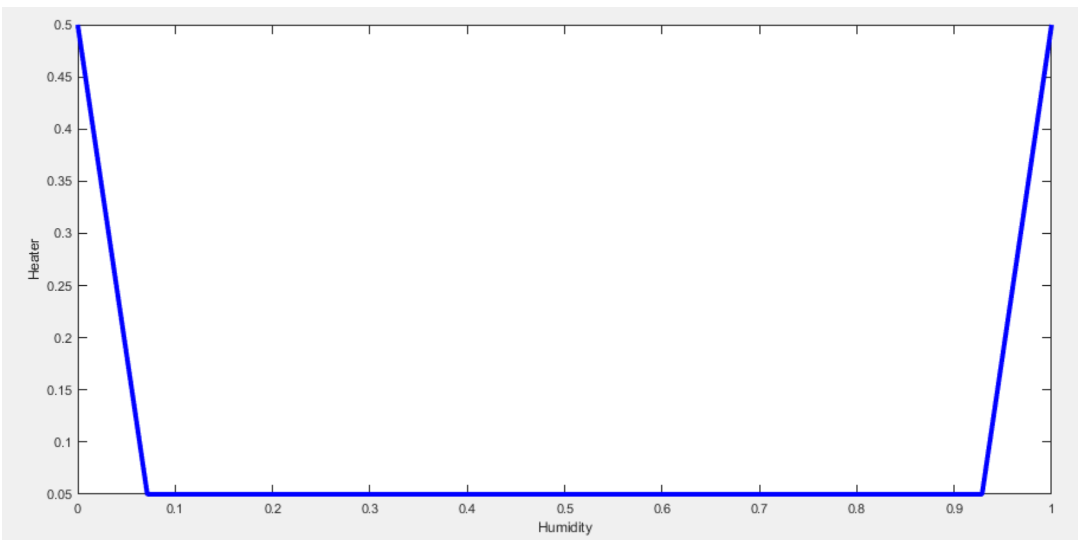


Figure 4.23: Relation Between Humidity and Heater

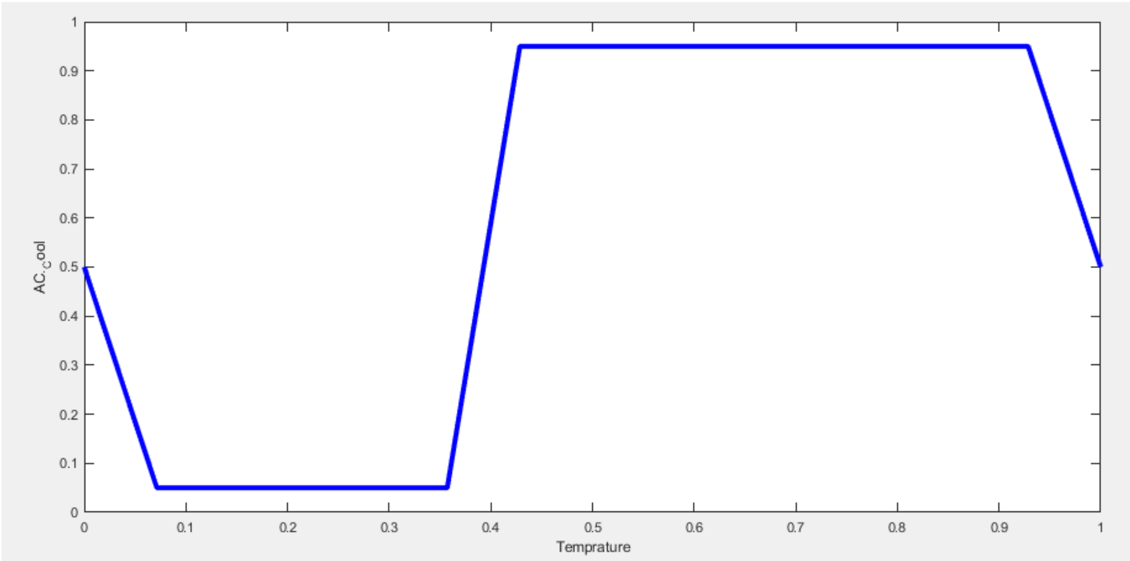


Figure 4.24: Relation Between Temperature and AC-Cool

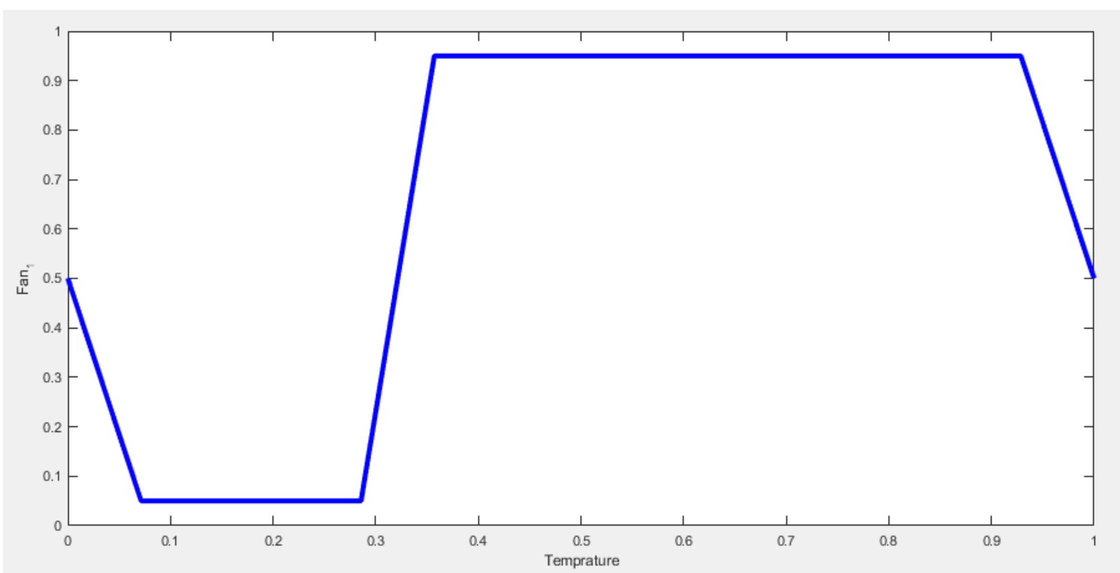


Figure 4.25: Relation Between Temperature and Fan1

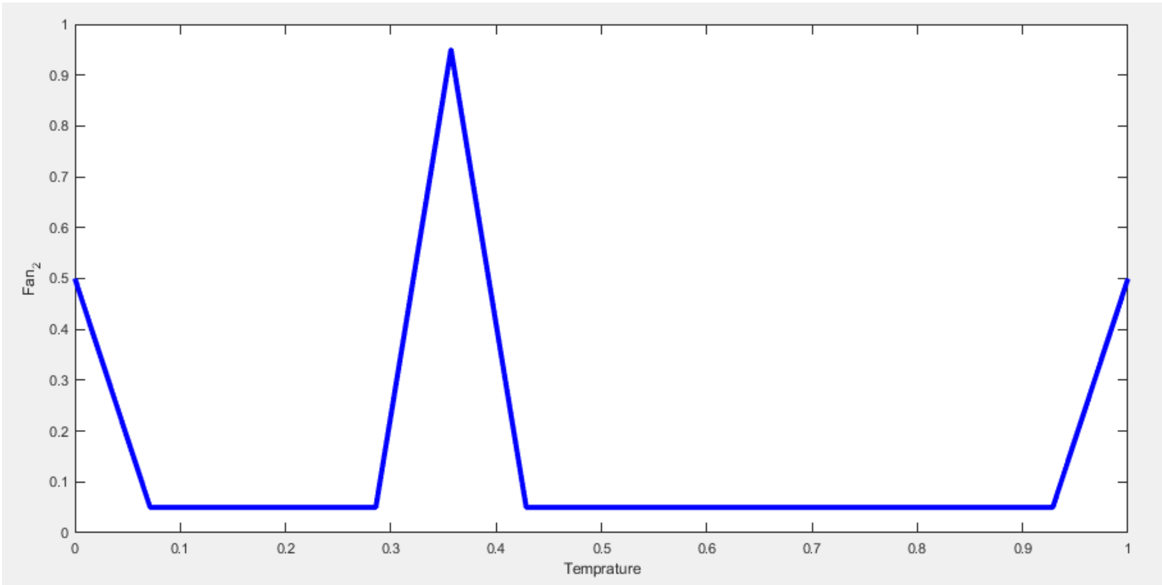


Figure 4.26: Relation Between Temperature and Fan2

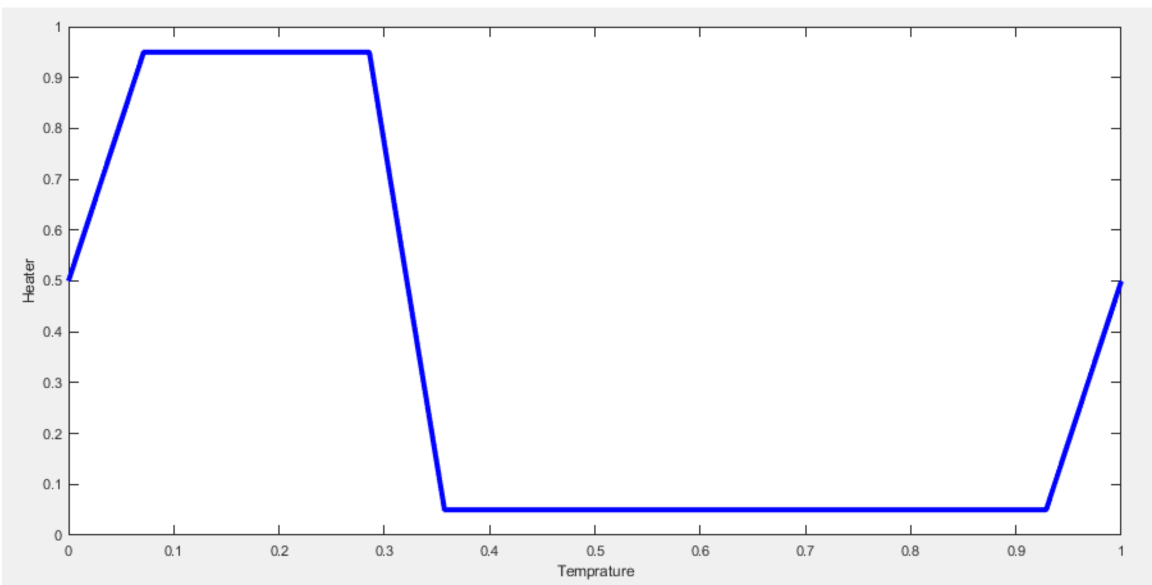


Figure 4.27: Relation Between Temperature and Heater

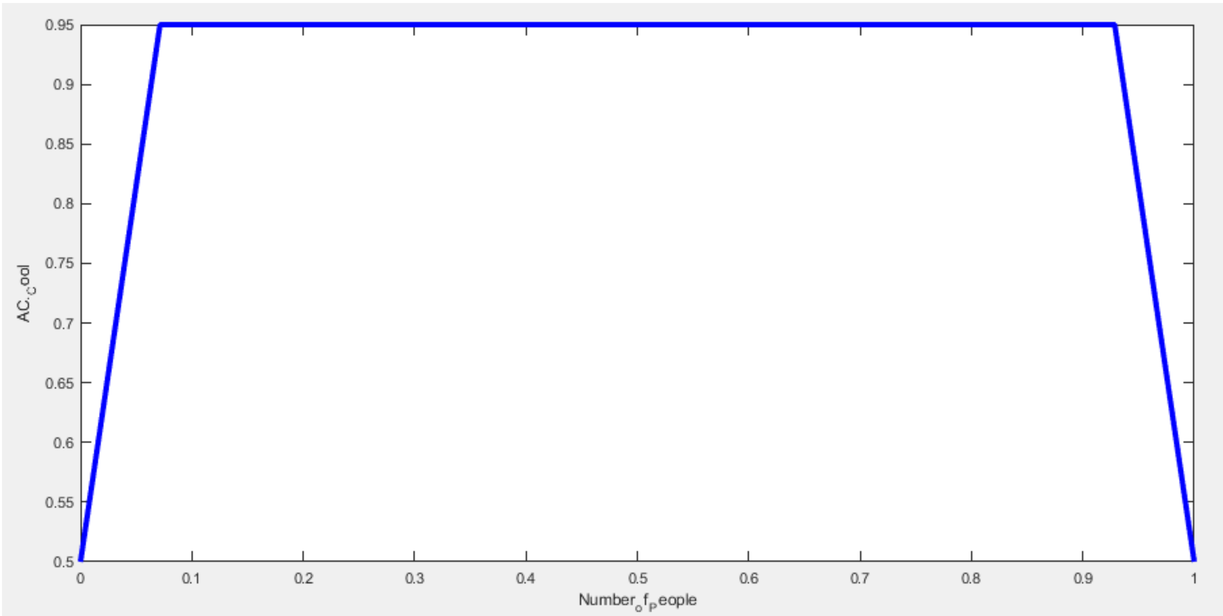


*Figure 4.28:* Relation Between Number-of-People and AC-Cool

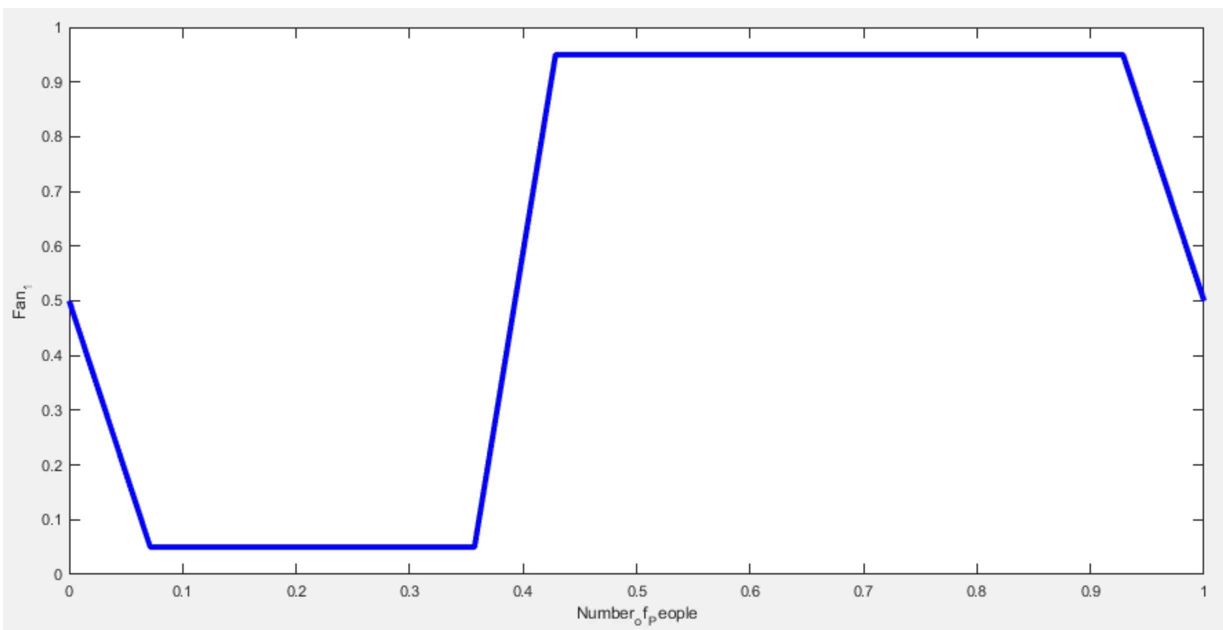


Figure 4.29: Relation Between Number-of-People and Fan1

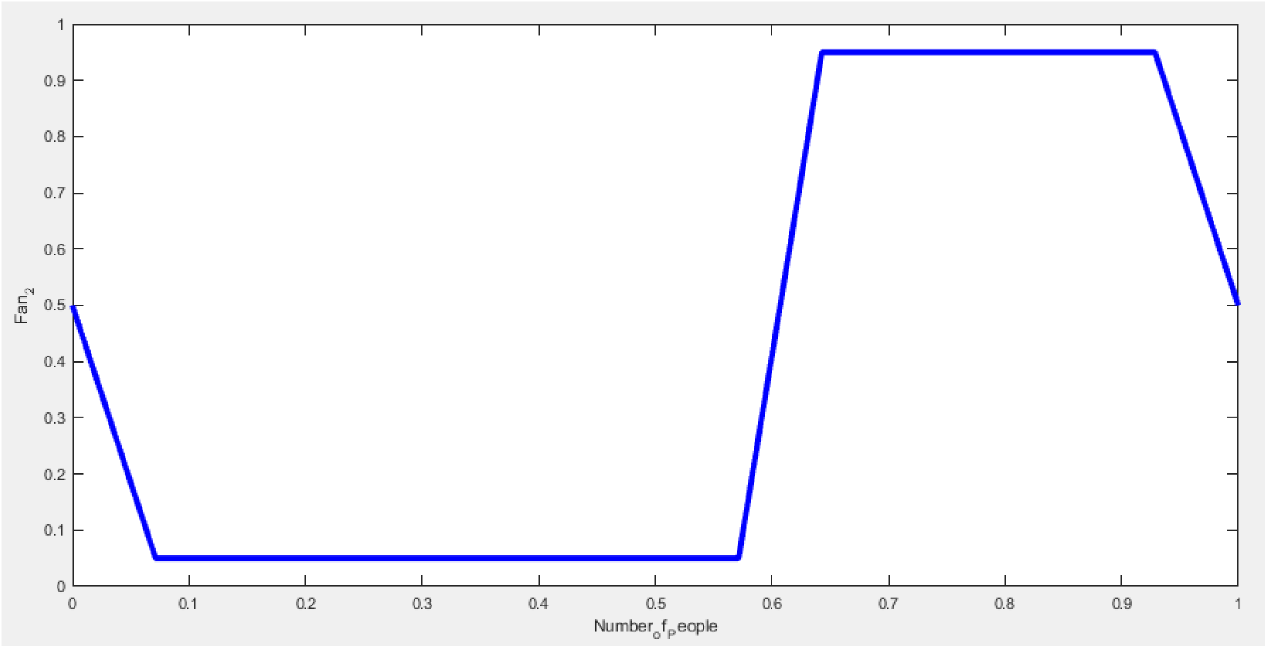


Figure 4.30: Relation Between Number-of-People and Fan2

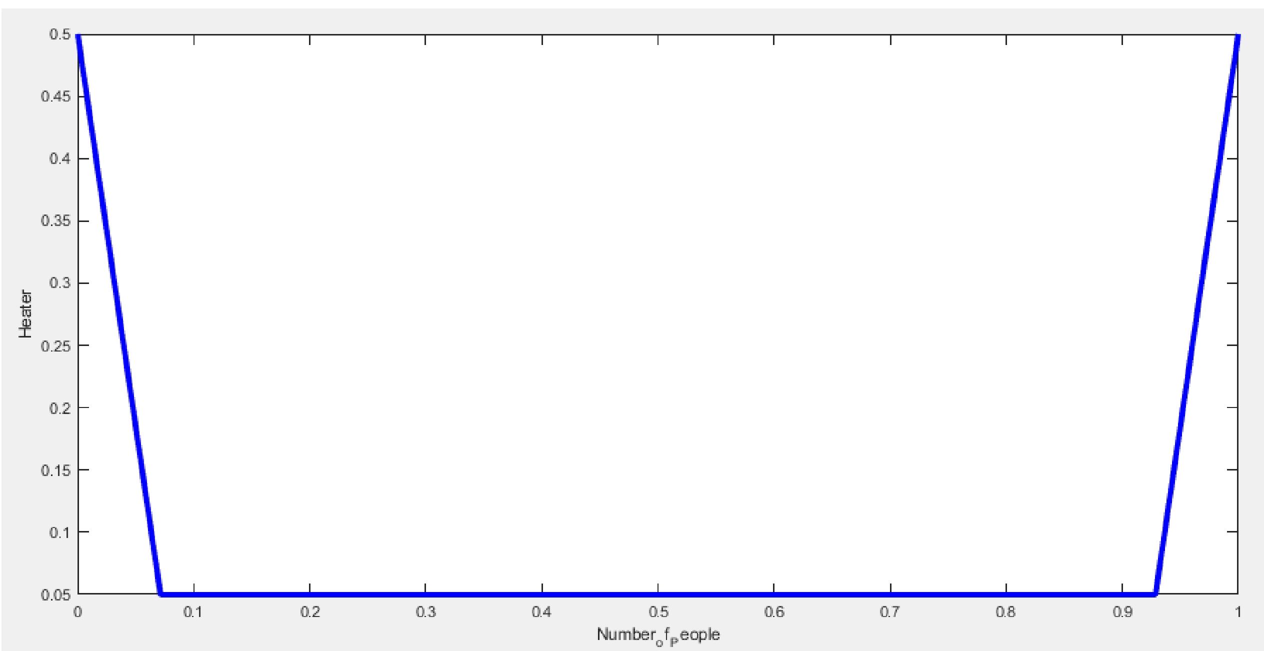


Figure 4.31: Relation Between Number-of-People and Heater

### 4.3.5 3-D Representations

3-D graphical representation shows relation between three variables. The first variable is input “X”, second variable is input “Y” and the third variable is output “Z”. In other words, by 3-D representation one can understand the proportionality of two input variables “X” and “Y” with one output variable “Z”.

Figure 4.*32*: Relation Between Input Variables Humidity & Temperature and Output Variable AC-Cool, shows relation between two input variables i.e. Humidity & Temperature and one output variable i.e. AC-Cool. Figure 4.*33*: Relation Between Input Variables Humidity & Temperature and Output Variable Fan-1, shows relation between two input variables i.e. Humidity & Temperature and one output variable i.e. Fan1. Figure 4.*34*: Relation Between Input Variables Humidity & Temperature and Output Variable Fan-2, shows relation between two input variables i.e. Humidity & Temperature and one output variable i.e. Fan2. Figure 4.*35*: Relation Between Input Variables Humidity & Temperature and Output Variable Heater, shows relation between two input variables i.e. Humidity & Temperature and one output variable i.e. Heater.

Figure 4.*36*: Relation Between Input Variables Humidity & Number-of-People and Output AC-Cool, shows relation between two input variables i.e. Humidity & Number-of-People and one output variable i.e. AC-Cool. Figure 4.*37*: Relation Between Input Variables Humidity & Number-of-People and Output Variable Fan1, shows relation between two input variables i.e. Humidity & Number-of-People and one output variable i.e. Fan1. Figure 4.*38*: Relation Between Input Variables Humidity & Number-of-People and Output Variable Fan2, shows relation between two input variables i.e. Humidity & Number-of-People and one output variable i.e. Fan2. Figure 4.*39*: Relation Between Input Variables Humidity & Number-of-People and Output Variable Heater, shows relation between two input variables i.e. Humidity & Number-of-People and one output variable i.e. Heater.

Figure 4.*40*: Relation Between Input Variables Temperature & Number-of-People and Output Variable AC-Cool, shows relation between two input variables i.e. Temperature & Number-of-People and one output variable i.e. AC-Cool. Figure 4.*41*: Relation Between Input Variables Temperature & Number-of-People and Output Variable Fan-1, shows relation between two input variables i.e. Temperature & Number-of-People and one output variable i.e. Fan1. Figure 4.*42*: Relation Between Input Variables Temperature & Number-of-People and Output Variable

Fan-2, shows relation between two input variables i.e. Temperature & Number-of-People and one output variable i.e. Fan2. Figure 4.*43*: Relation Between Input Variables Temperature & Number-of-People and Output Variable Heater, shows relation between two input variables i.e. Temperature & Number-of-People and one output variable i.e. Heater.

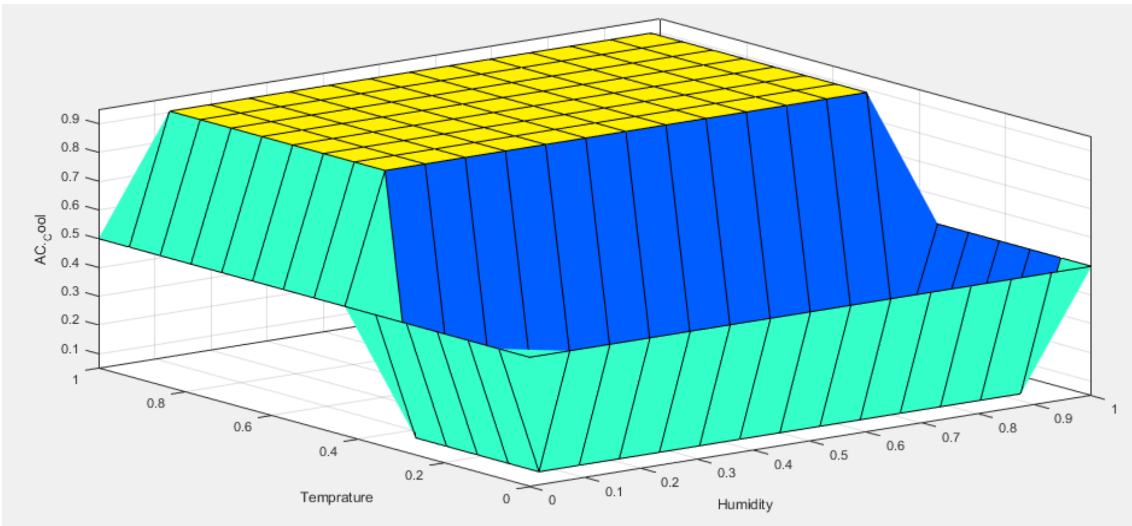


Figure 4.32: Relation Between Input Variables Humidity & Temperature and Output Variable AC-Cool

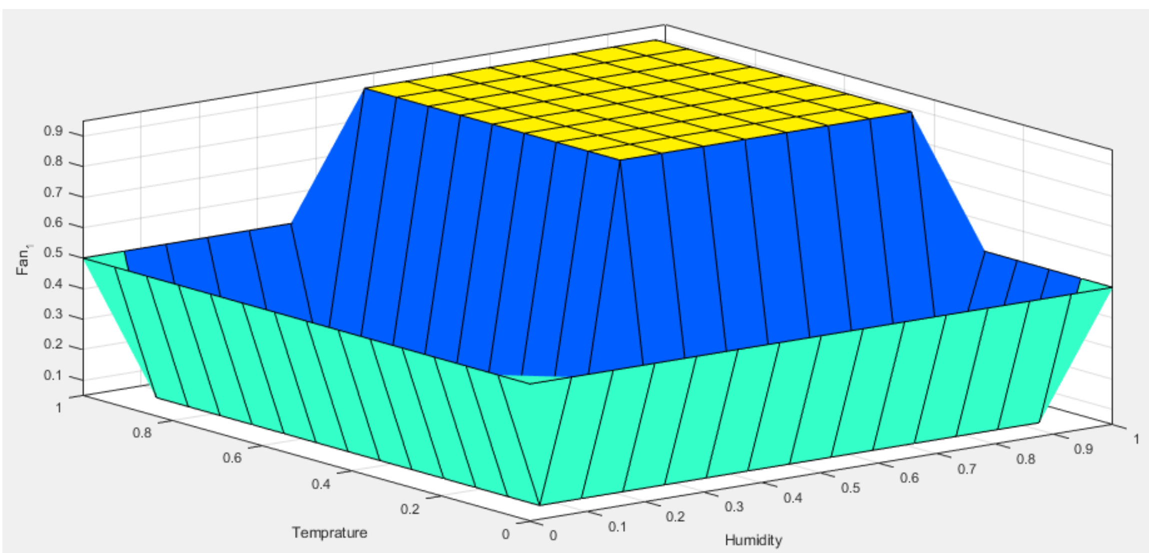


Figure 4.33: Relation Between Input Variables Humidity & Temperature and Output Variable Fan-1

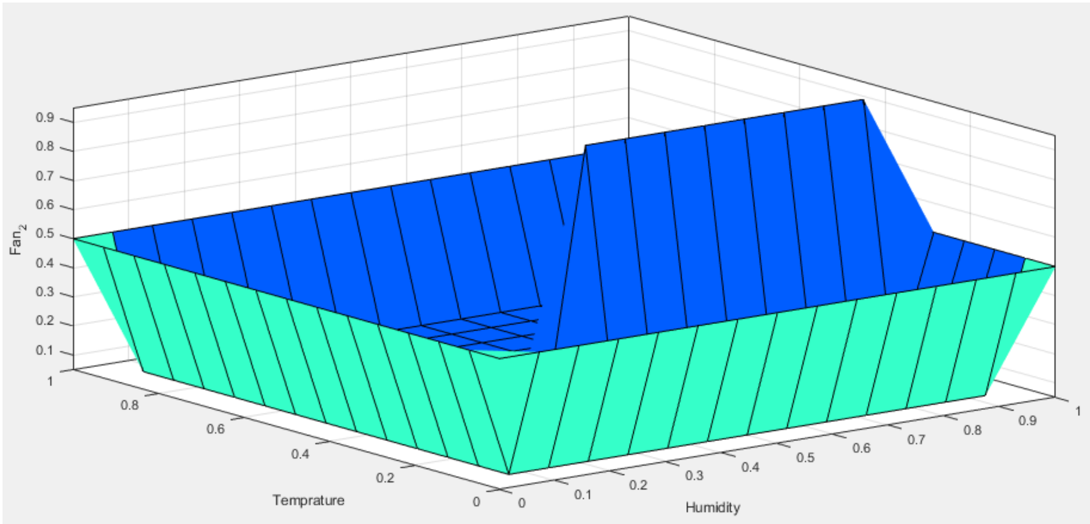


Figure 4.34: Relation Between Input Variables Humidity & Temperature and Output Variable Fan-2

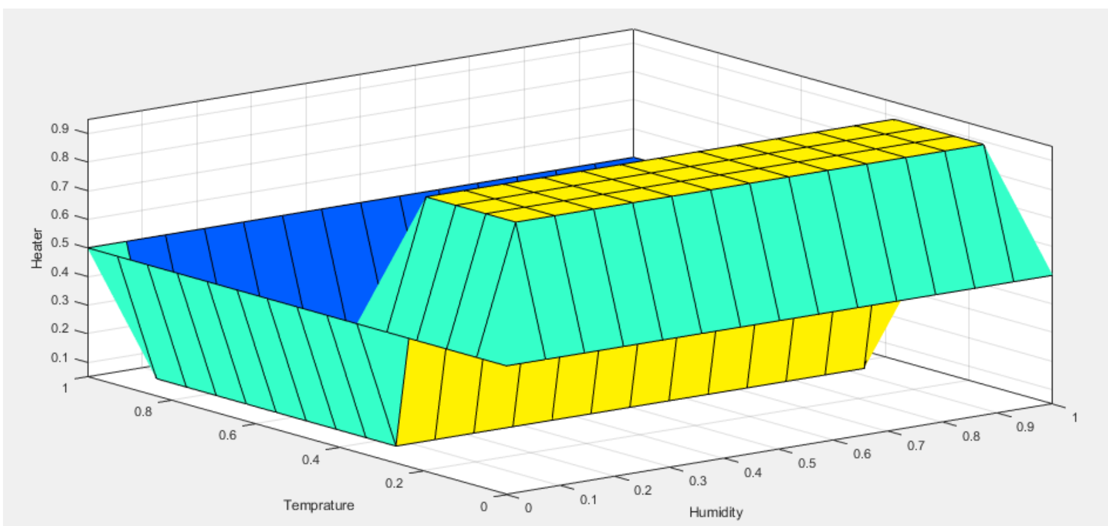


Figure 4.35: Relation Between Input Variables Humidity & Temperature and Output Variable Heater

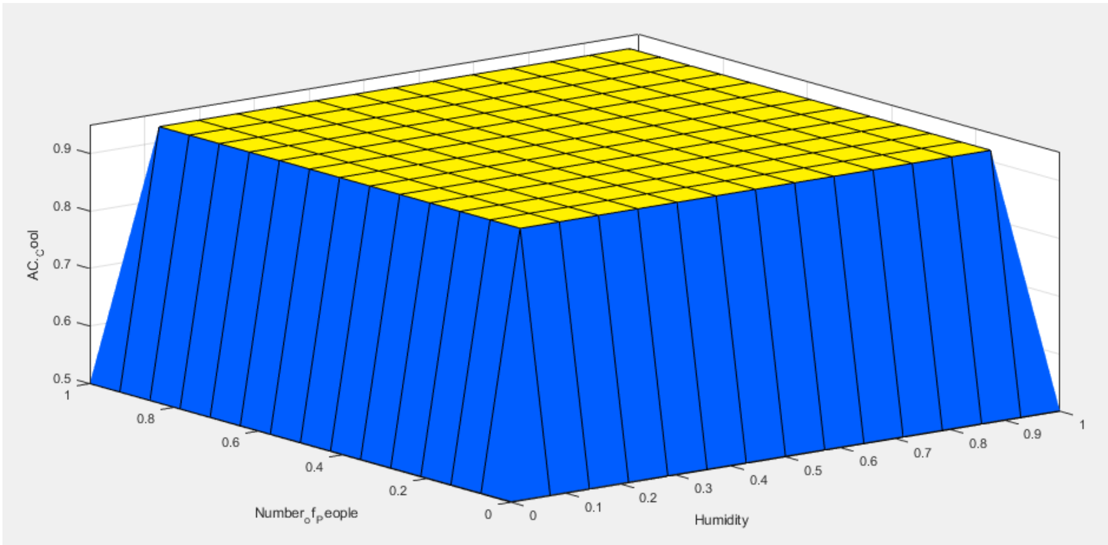


Figure 4.36: Relation Between Input Variables Humidity & Number-of-People and Output AC-Cool

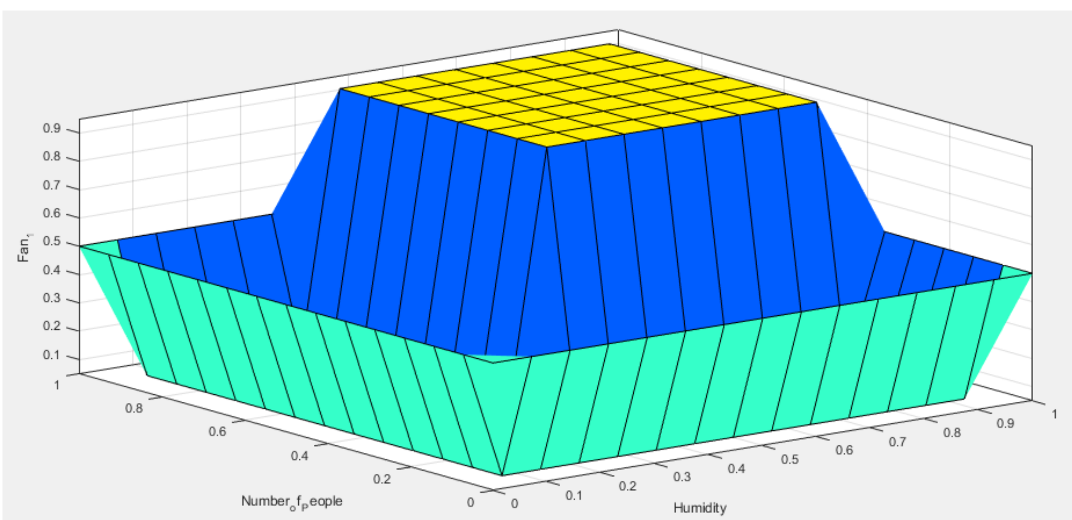


Figure 4.37: Relation Between Input Variables Humidity & Number-of-People and Output Variable Fan1

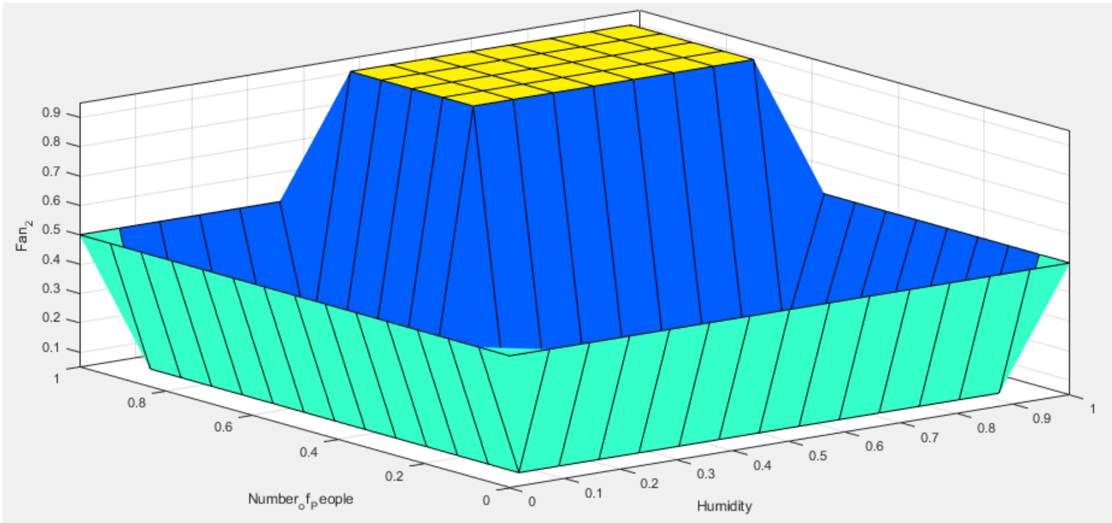


Figure 4.38: Relation Between Input Variables Humidity & Number-of-People and Output Variable Fan2

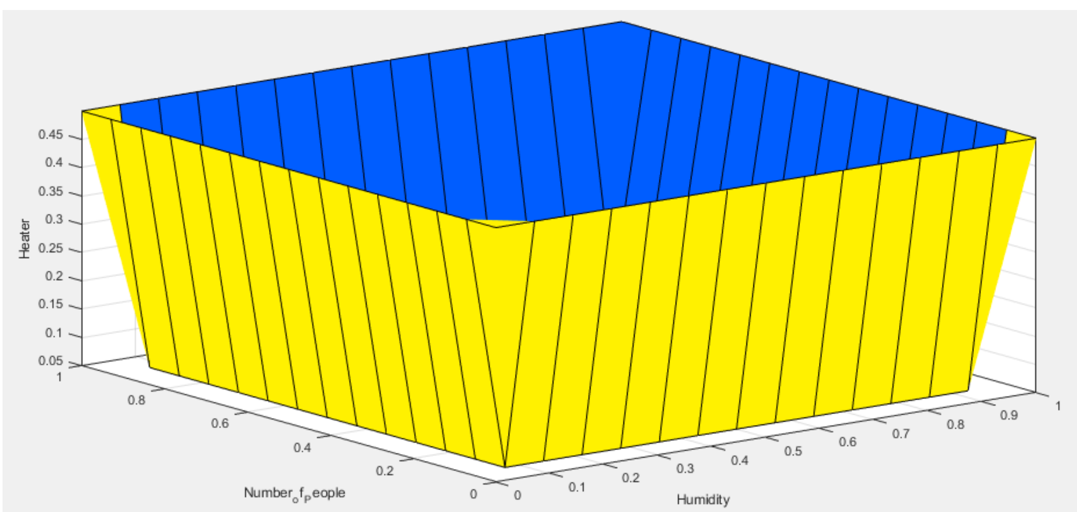


Figure 4.39: Relation Between Input Variables Humidity & Number-of-People and Output Variable Heater

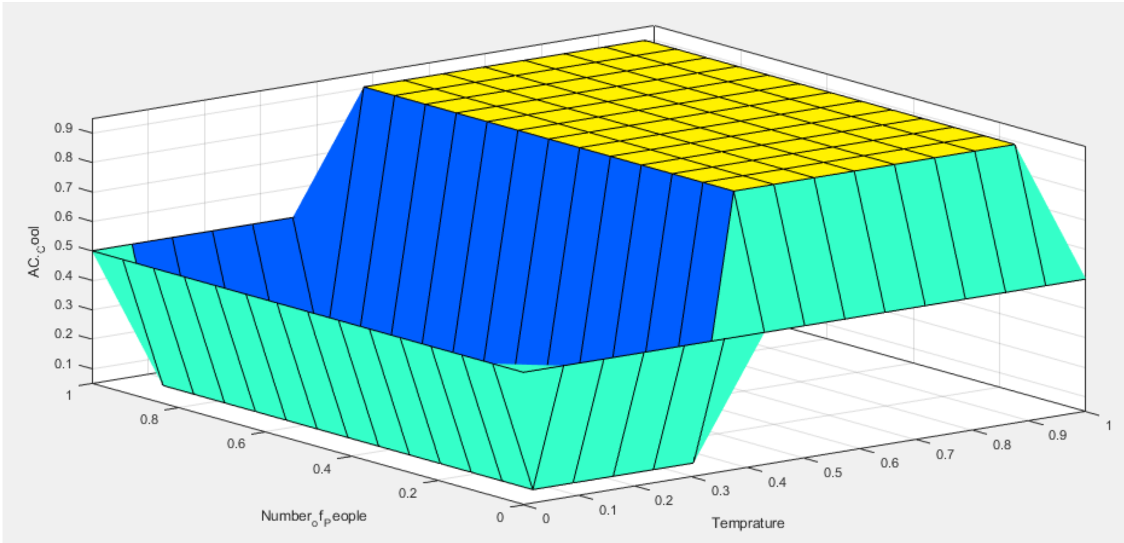


Figure 4.40: Relation Between Input Variables Temperature & Number-of-People and Output Variable AC-Cool

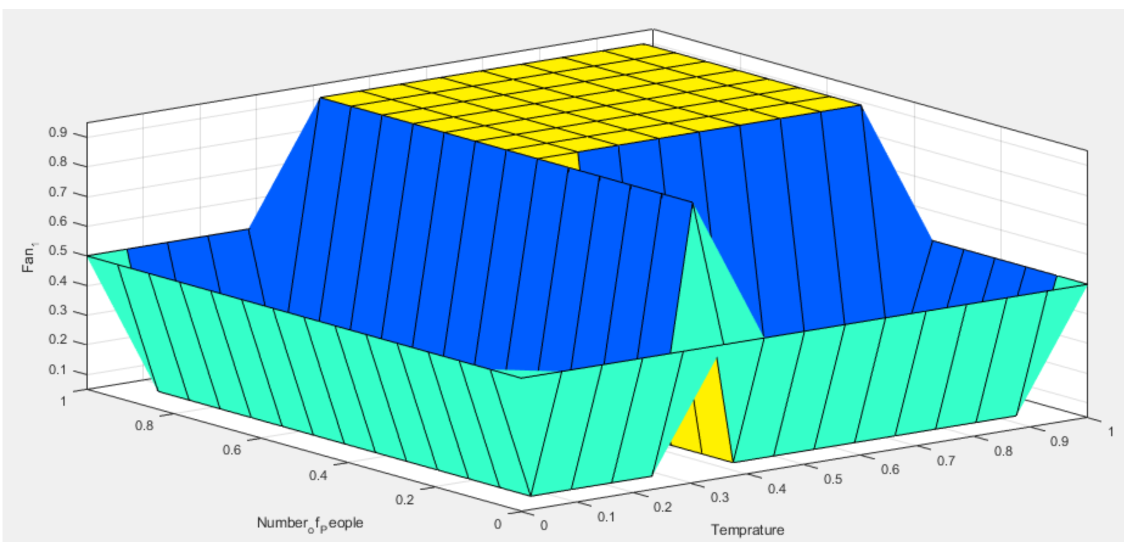


Figure 4.41: Relation Between Input Variables Temperature & Number-of-People and Output Variable Fan-1

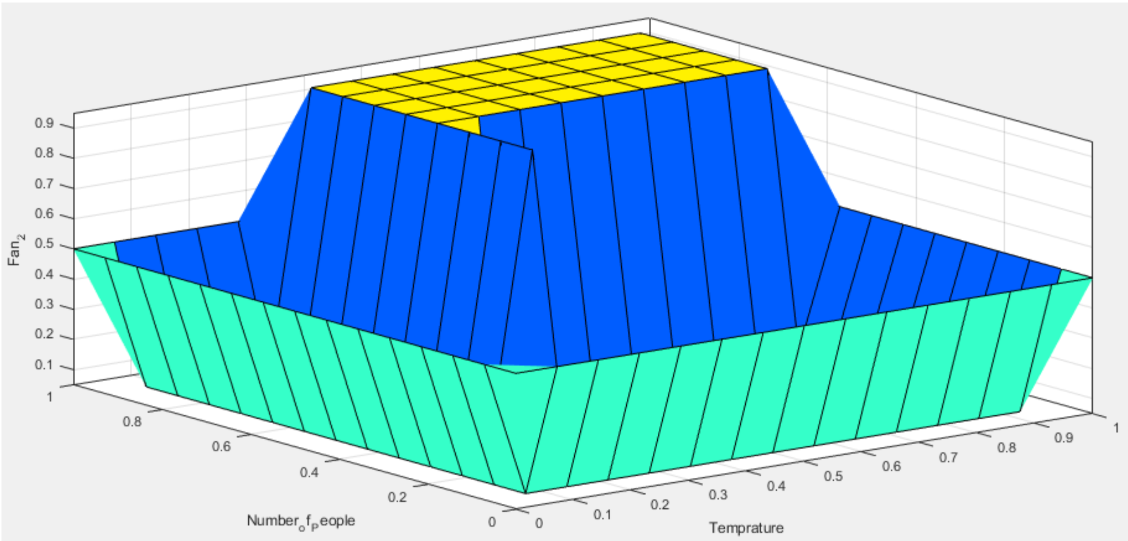


Figure 4.42: Relation Between Input Variables Temperature & Number-of-People and Output Variable Fan-2

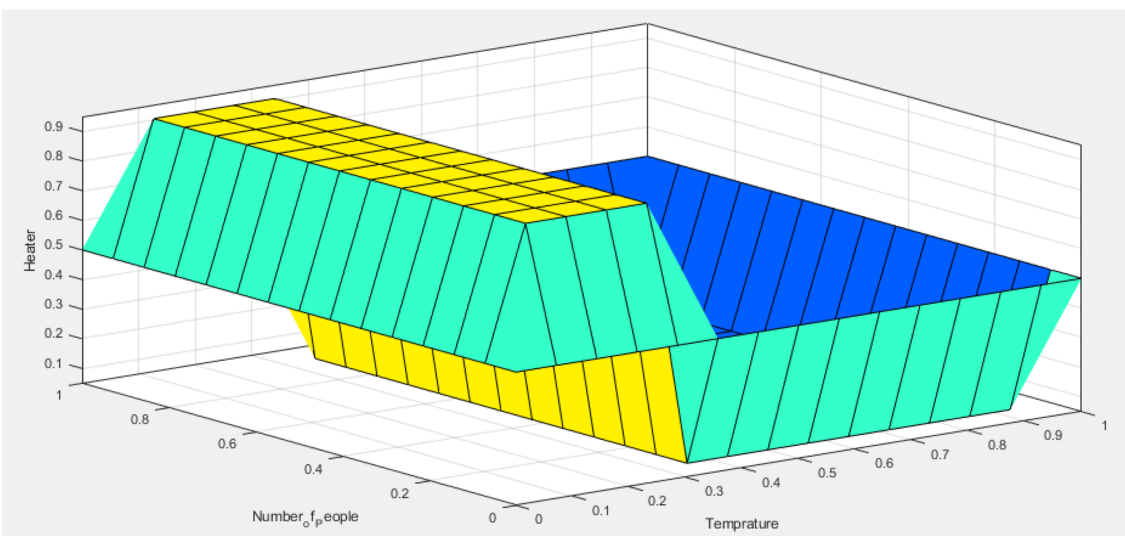


Figure 4.43: Relation Between Input Variables Temperature & Number-of-People and Output Variable Heater

## 4.4 Modeling of Bus Rapid Transit System

Let us consider the scenario of Bus Rapid Transport System. Bus Rapid Transport Systems (BRTS) are used in different developing and developed countries such as France, Germany, Turkey, India, and Pakistan. BRTS are an effective source of transportation in cities for avoiding traffic congestion and timely traveling. These systems are cheaper alternate of metropolitan trains because setup cost of BRTS is much less than that of metropolitan trains. Normally, there are rush hours when the number of passengers increases and sometimes the rush is so severe that it becomes difficult for passengers to timely and comfortably reach the destination. So, for understanding and analyzing the system we need to model it.

In this section we provide model of a generic BRT system for proof of concept. We modeled it at abstract level with less details. As ambient may be in hierarchical manner such as a person riding on a bus may contain a bag and in bag there may be laptop or mobile phone. So, in this case more details may be provided for a specific purpose. Now we are following the steps provided by the framework for the purpose of modeling.

### 4.4.1 Identification of Agents

We have three agents in our model i.e., Buses, People, Stations.

### 4.4.2 Classification of Agents

Buses: Buses are ambient

People: People are mobile agents

Stations: Stations are Static and Meta-agents

### 4.4.3 Determining the Relations of Agents

There are four stations, and the road is connecting these stations. People can ride on a bus. People can stand on a station. Bus stops on a station. Bus travels through road.

Figure 4.*44*: Casual Loop shows the casual loop of the system. Figure 4.*45*: Stock and Flow

Diagram shows the stock and flow diagram of the system. Figure 4.*46*: a) State Chart for Agent "Bus" b) State Chart for Agent "Passenger" has two parts, the first part a) shows the state-chart of the agent named as Bus and the second part b) shows the state-chart of the agent named as Passenger.

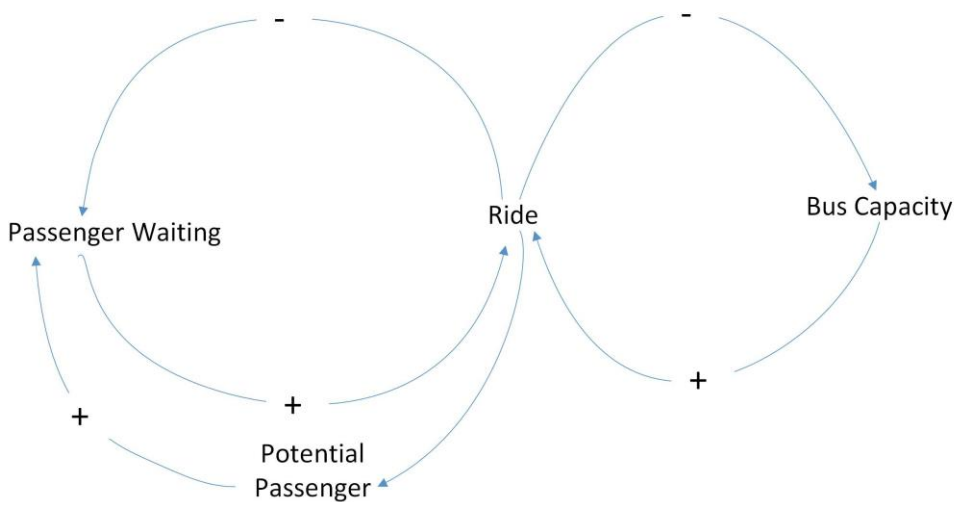


Figure 4.44: Casual Loop

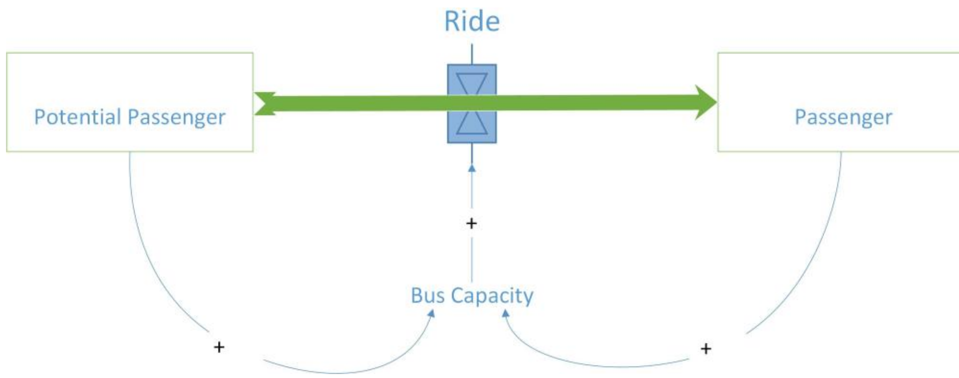


Figure 4.45: Stock and Flow Diagram

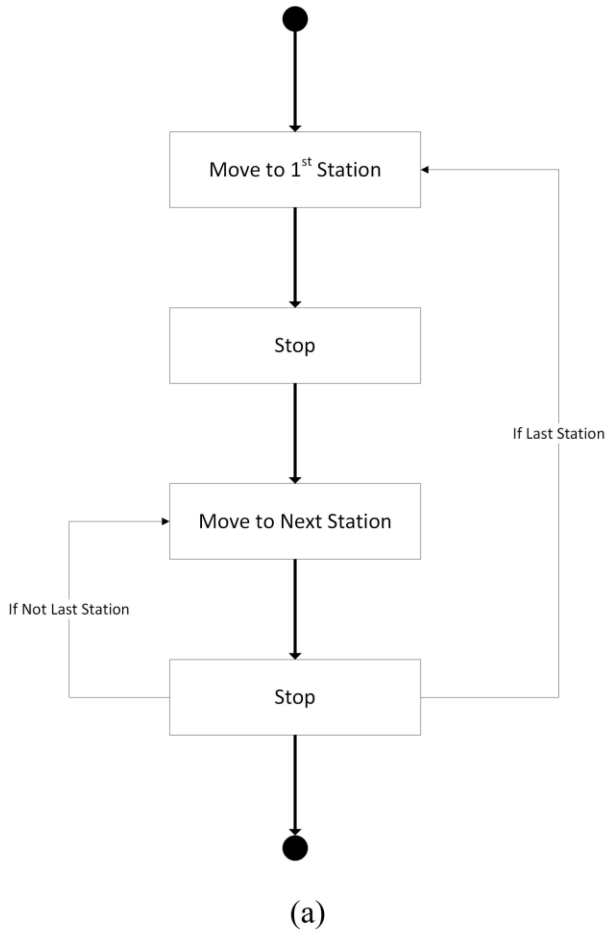


Figure 4.46: a) State Chart for Agent "Bus" b) State Chart for Agent "Passenger"

### 4.4.4 Formalization of Processes

The ambient-oriented model according to the above defined rules will be:

There are two ambient bus and passenger. The ambient bus is represented by *B* and the ambient passenger is represented by *X*. *P* is the process. *S* is the station. αis the location. mov means move to.

$PB \mid PX$

It shows both processes are parallel.

$P_B \Rightarrow (mov(\alpha == S_1))\ P_X \Rightarrow (mov(\alpha == S_1))$

The above shows bus moves to station one in first process and in second process passenger moves to station one.

$P_B \Rightarrow (in, :: X)$

It shows that the passenger standing on the station move in the bus.

$P_B \Rightarrow (mov(\alpha == S_2)\ P_X \Rightarrow (mov(\alpha == S_2))$

It shows bus moves to station two and similarly new passenger come to station two.

$P_B \Rightarrow (out, \uparrow X)$
$P_B \Rightarrow (in, :: X)$

In the above two line, the passenger which are inside the bus mean bus is parent of whom should get off whereas the passenger standing on the station who are the siblings of the bus should ride on the bus.

$P_B \Rightarrow (mov(\alpha == S_3)$
$P_X \Rightarrow (mov(\alpha == S_3))\ P_B \Rightarrow (out, \uparrow X)$
$P_B \Rightarrow (in, :: X)$
$P_B \Rightarrow (mov(\alpha == S_4)$
$P_B \Rightarrow (out, \uparrow X)$

In the above lines, similar to previous explanation has been presented. Let $F$ is the first station, $N$ is the last station and $O$ is constant. Process for bus $P_B$ will be modeled as shown below.

$$P_B \cong \begin{cases} mov(\alpha\ ==\ S_F\ ) \\ (in, ::X\ ) \\ mov(\alpha\ == S_{F+O}\ ) \\ (out, \uparrow\ X\ ) \\ (in, ::X\ ) \\ mov(\alpha\ == S_N) \\ (out, \uparrow\ X\ ) \end{cases} ) \qquad (4.0.10)$$

Let $S_{nearest}$ is the station nearest to location of passenger and the $S_{destination}$ is the destination of the passenger. The process X for passenger will be modeled as shown below:

$$P_X \cong \begin{cases} mov(\alpha\ ==\ Snearest\ ) \\ mov(in, ::\ B) \\ mov(\alpha\ ==\ S_{destination}\ \ ) \\ mov(out, \downarrow\ B) \end{cases} \qquad (4.0.11)$$

### 4.4.5 Formalization of Messages

Bus shares its current location to the next bus station and the bus station calculates the approximate reaching time and intimate to the passenger. So, this may be modeled as:

$P_B \Rightarrow (S_1 ::< bus_{-}id,\ \alpha >,\ 0)$

$P_X \Rightarrow (S1 :: (bus_{-}id,\ exp_{-}time), 0)$

The messages by the agent Station are as following:

$$P_X \cong \begin{cases} B ::\ (bus_{id}, \alpha), 0 \\ X ::\ <\ bus_{id}, exp_{time}\ >, 0x, \qquad x \geq 0 \end{cases} \qquad (4.0.12)$$

### 4.4.6 Formation of Rules

Assuming normal hours and routine service, when there is no rush, the rules will be:
The rules of the bus will be bus moves from source to station one. At station bus moves to bus stop and waits. After pickup bus moves to station two. At station two, bus first drops the passengers

and then picks up passengers. Then bus moves to station three. At station three, bus first drops passengers and then picks up passenger. Then the bus moves to station four which is the final station. At station four, bus drops passengers and ends. The rules of the passenger will be; the passenger comes to platform, waits for bus arrival, rides on bus, moves to desired station, gets off to platform and exits from platform.

### 4.4.7 Simulation of Model and Graphs

We used Anylogic simulator for simulation. Figure 4.*47*: Workflow of Model for BRTS shows the flowchart of the model along with 2D run. After running the values show the number of buses as well as passengers who crossed specific area. The rules are represented in this flowchart. Figure 4.48 shows the 3D representation of the model where one can see buses picking up passengers, moving, dropping of passengers, and exiting from the area. Whereas the passengers coming to a station, riding on the bus and exiting from the bus and leaving the station. With the 3-D model a graph is shown in the same figure. This graph has been created to analyze the waiting time of the people at first station. The Bus Rapid Transit Systems with isolated tracks are represented here in a model.

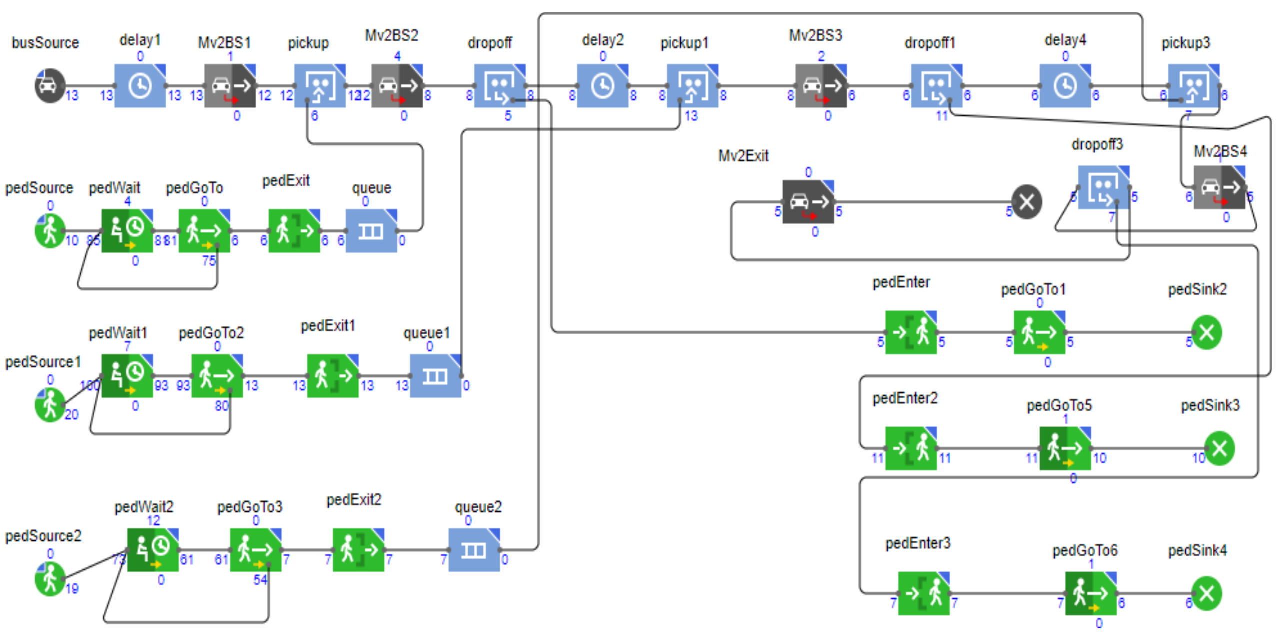


Figure 4.47: Workflow of Model for BRTS

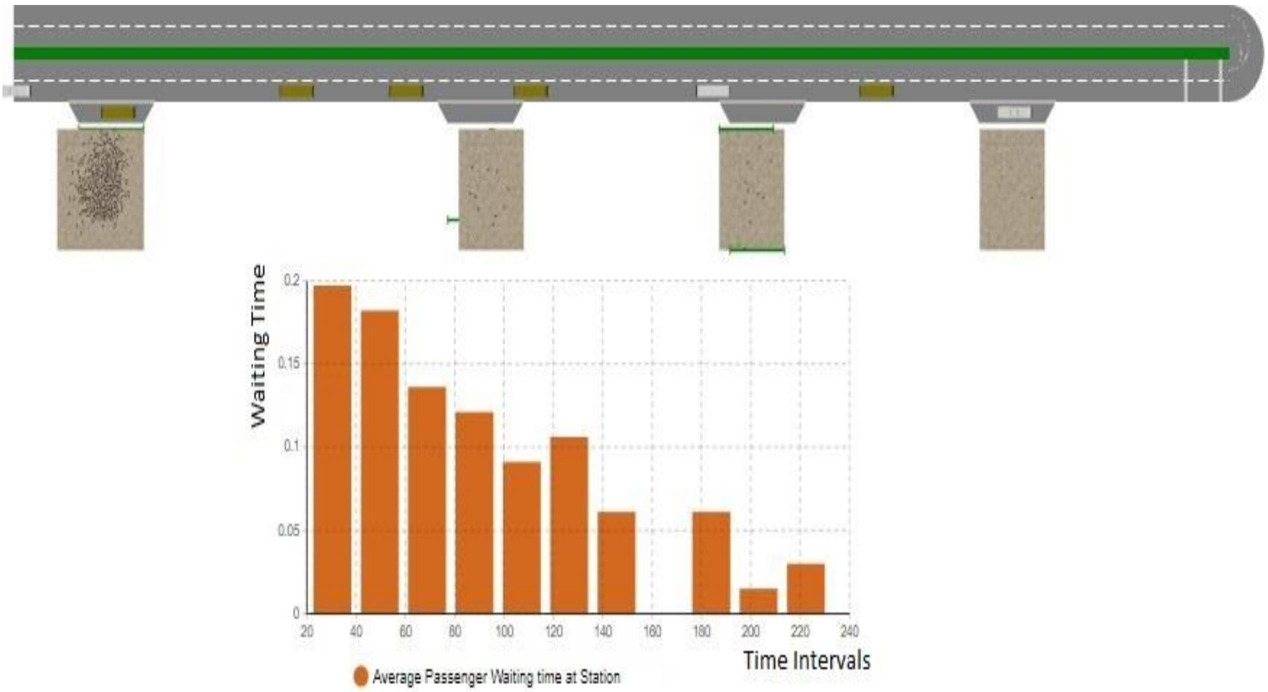


Figure 4.48: 3D-view of BRTS Simulation with Graph for Waiting Time

Here, we provide simulation of a specific model of Rawalpindi-Islamabad metro bus system. In Figure 4.*49*: The Main View of Model Including Flexible Slide Bars for Selecting Variables, the start screen of simulation has been displayed. In this figure at the top "Bus Rapid Transit System" is the name of this model. There are certain attributes that are flexible, and the user may select the values according to need. Frequency of Buses is the first attribute, and it has numeric values from 0 to 100. The user may select any value by dragging the bar. This number will add the number of buses in the system. Here, 0 buses will mean that there is no bus in the system. 10 buses means that the number of buses running in the simulation will be 10. The second attribute is Inter-arrival time of Buses which shows the time gap between different buses. The third flexible attribute is Stop-time of bus which represents the time that a bus stops at a station. The next flexible attribute is Bus Capacity which shows the total number of passengers that can ride on a bus at a time. The last flexible attribute is Bus Speed which shows the maximum speed a bus can achieve. In the left side of this image the values selected for attributes has been displayed. In the bottom of the image information about the model has been provided.

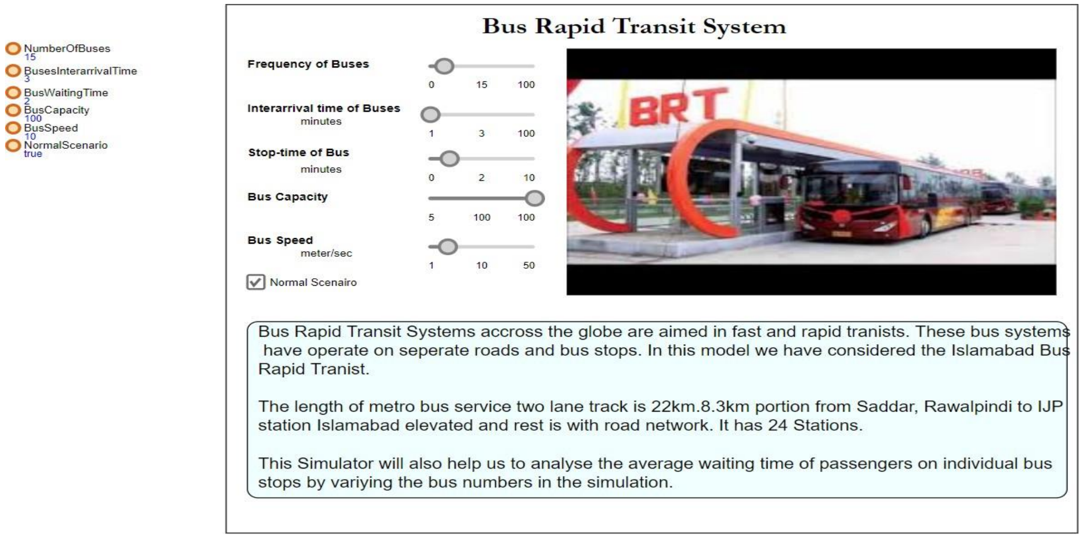


Figure 4.49: The Main View of Model Including Flexible Slide Bars for Selecting Variables

In Figure 4.*50*: Map of Islamabad for Simulation, the simulation 3-D simulation environment of model has been shown. The simulation environment is based on the map of route taken from Google. In the lower part of the image workflow of the model has been shown. In the bottom left the number of stops and number of buses is shown. The variable “I” on the map shows the number/identity of stations. The variable P shows the number of passengers at a station.

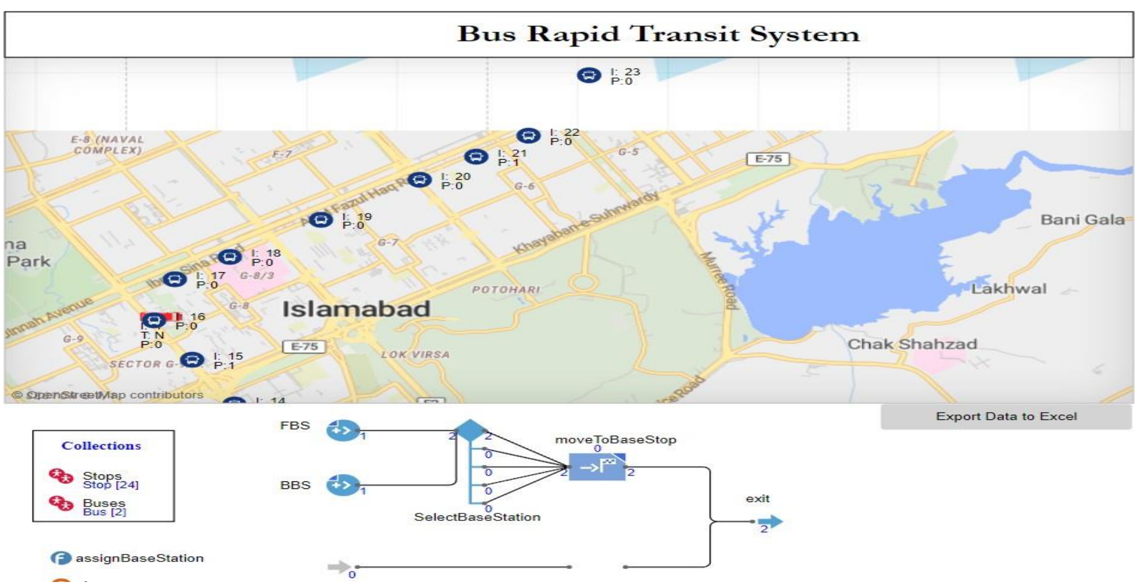


Figure 4.50: Map of Islamabad for Simulation

Figure 4.*51*: Graph at Station 0, shows the graph at station 0 where the blue line indicates the average waiting time in the direction from station 0 towards station 23 and the red line indicates the average waiting time from station 23 towards station 0. There is no red line in graph at station 0 because station 0 is the last station in this direction and no passenger is in waiting. Figure 4.*52*: Graph at Station 1, shows the average waiting time at station 1. At station 1 passenger can move in both directions therefore both the red and blue lines are present in this graph. Figure 4.*53*: Graph at Station 2, *Figure 4.54: Graph at Station 3*, Figure 4.*55*: Graph at Station 4, Figure 4.*56*: Graph at Station 10, Figure 4.*57*: Graph at Station 11, Figure 4.*58*: Graph at Station 12, Figure 4.*59*: Graph at Station 21, Figure 4.*60*: Graph at Station 22 and Figure 4.*61*: Graph at Station 23 show the average waiting time at station 2, 3, 4, 10, 11, 12, 21, 22 and 23 respectively. At station 23, in graph only red line is present because it is the last station in the direction from station 0 to 23. All these graphs are based on the set values of flexible attributes. These values are of attributes are number of buses is set at 12, inter-arrival time is set at 3, waiting time is 2 minutes, bus capacity is 100 and bus speed is set at 10 KM/H. Also, the normal scenario check box is selected.


Figure 4.51: Graph at Station 0

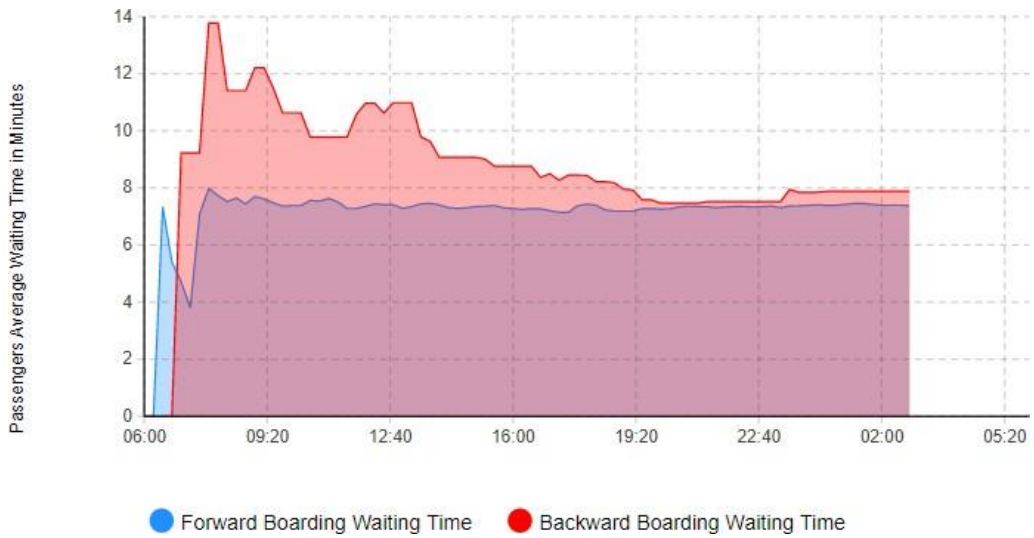


Figure 4.52: Graph at Station 1

After just increasing the number of buses change in waiting time can be seen in Figure 4.*62*: After Changing the Frequency at Station 1, Figure 4.*63*: After Changing the Frequency of Buses the Graph at Station 10 and Figure 4.*64*: After Changing the Frequency at Station 21 which show graph at station 1, 10 and 21 respectively. As compared to the previous graphs of respective stations, decrease in the average waiting time can be seen.

The previous graphs show that changing the number of buses cause a decrease in the average waiting time. Now we change some more variables and see how this effect in average waiting time. Now we increase the frequency of buses to 51 as shown in Figure 4.*65*: Change in the Frequency of Buses which was at first set at 15. The change in average waiting time can be noticed by graphs shown in Figure 4.*66*: Graph at Station 0 After Increasing the Number of Buses, Figure 4.*67*: Graph at Station 1 After Increasing the Number of Buses, Figure 4.*68*: Graph at Station 10 After Increasing the Number of Buses, Figure 4.*69*: Graph at Station 21 After Increasing the Number of Buses, Figure 4.*70*: Graph at Station 23 After Increasing the Number of Buses for stations 0, 1, 10, 21 and 23 respectively. There is a significant decrease in average waiting time compared to the previous values.

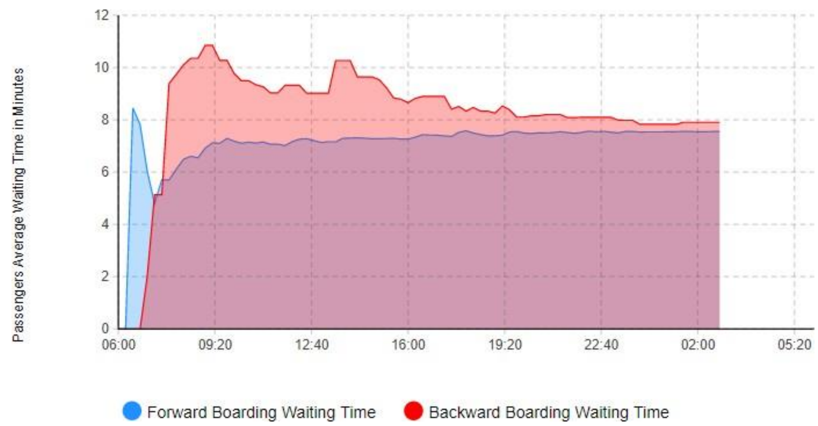


Figure 4.53: Graph at Station 2


*Figure 4.54: Graph at Station 3*

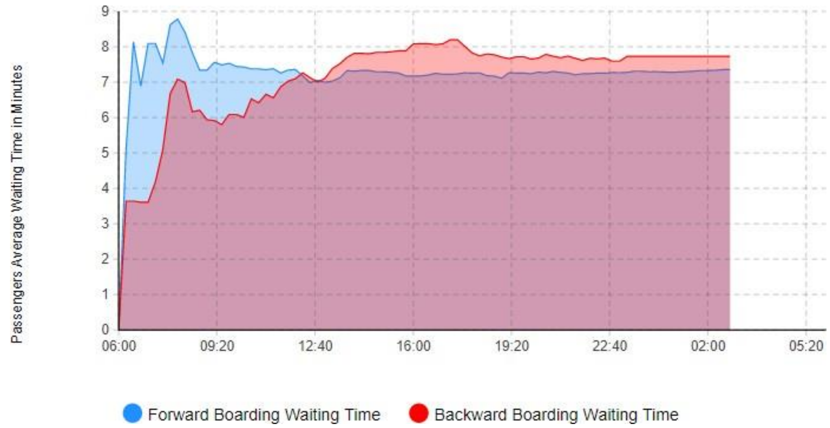


Figure 4.55: Graph at Station 4

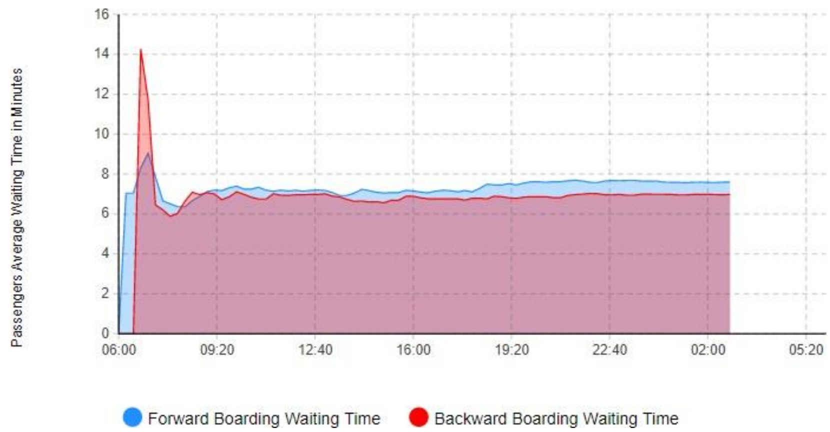


Figure 4.56: Graph at Station 10

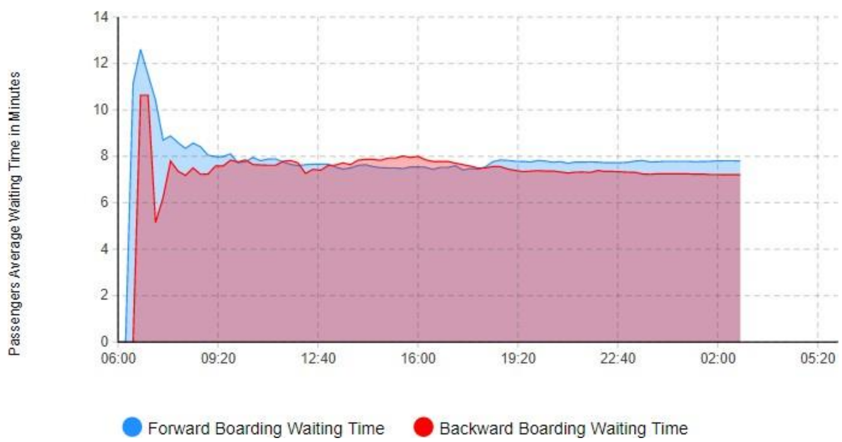


Figure 4.57: Graph at Station 11

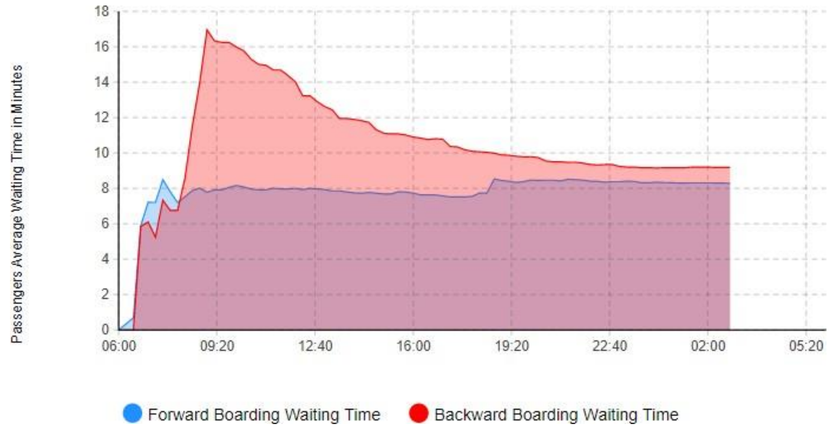


Figure 4.58: Graph at Station 12

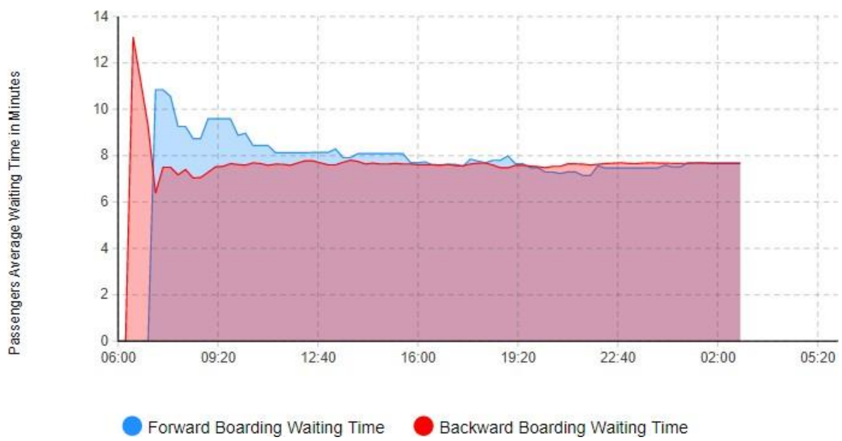


Figure 4.59: Graph at Station 21

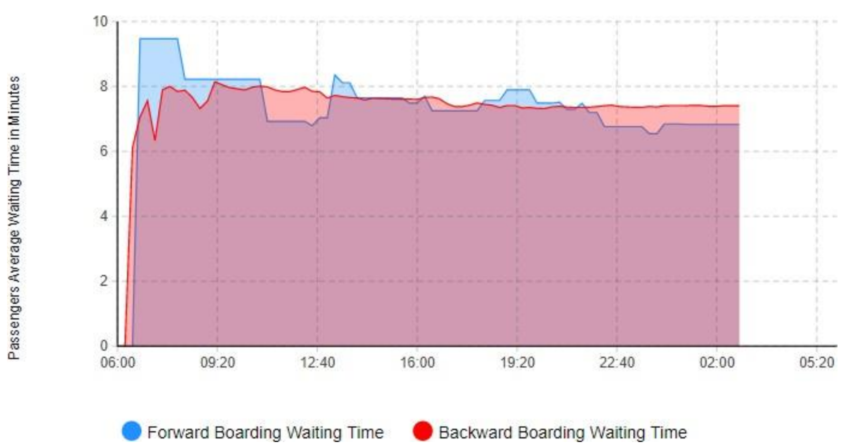


Figure 4.60: Graph at Station 22


Figure 4.61: Graph at Station 23


Figure 4.62: After Changing the Frequency at Station 1

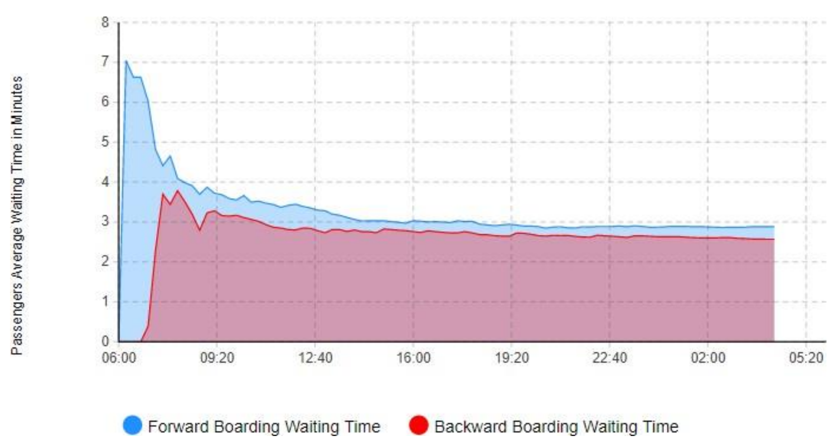


Figure 4.63: After Changing the Frequency of Buses the Graph at Station 10

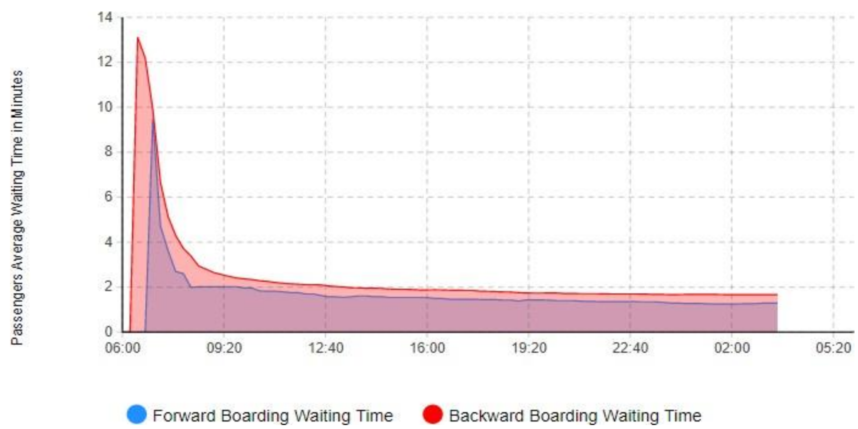


Figure 4.64: After Changing the Frequency at Station 21


Figure 4.65: Change in the Frequency of Buses


Figure 4.66: Graph at Station 0 After Increasing the Number of Buses

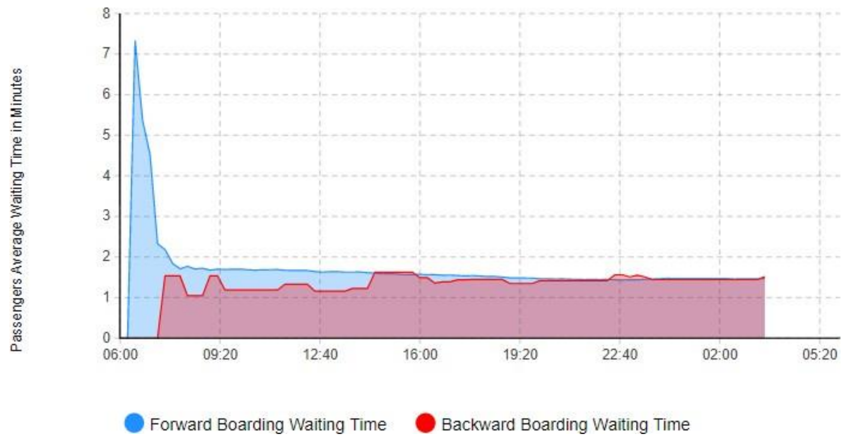


Figure 4.67: Graph at Station 1 After Increasing the Number of Buses

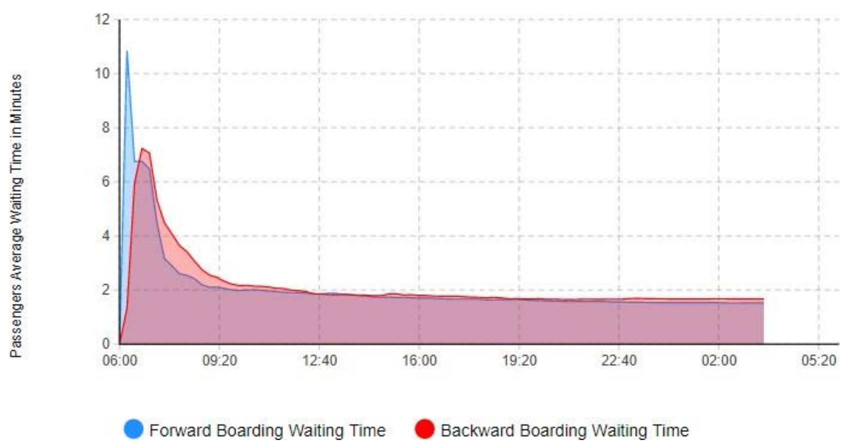


Figure 4.68: Graph at Station 10 After Increasing the Number of Buses

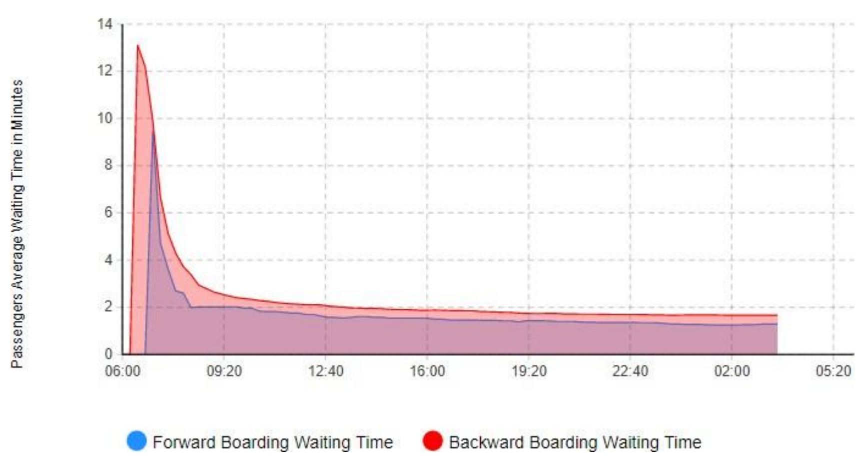


Figure 4.69: Graph at Station 21 After Increasing the Number of Buses

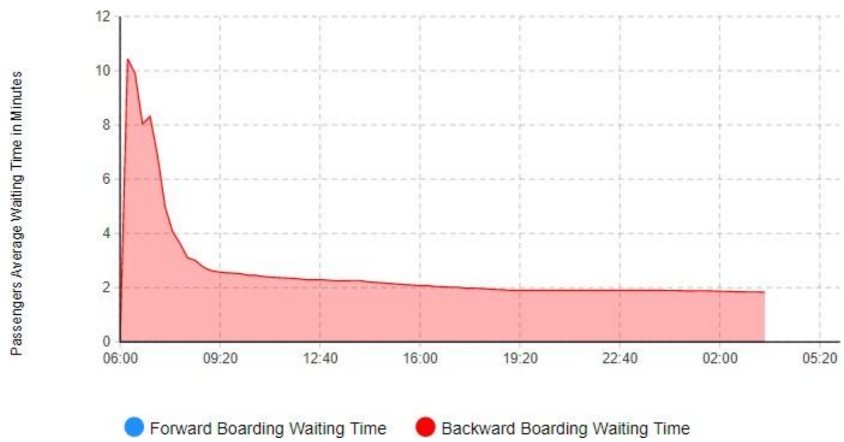


Figure 4.70: Graph at Station 23 After Increasing the Number of Buses

## 4.5 Discussion

In this section, we discuss the value-addition of the proposed framework for developing models of the systems that are composed of agents and ambient. In the absence of any comparative frameworks, here we examine how to design complex agent-based ambient-oriented systems in general. In particular, we also discuss how conventional modeling schemes fail to provide effective tools for building detailed models and simulation of agent-based ambient-oriented systems.

### 4.5.1 Problem in Modeling Agent-based Ambient-oriented Systems

The concept of agent and ambient functioning that are used in collaboration was provided by [241] where, first time the ambient was treated as a mobile entity. In this article the agents were considered as entities that are placed within an ambient. On the basis of this article, [227] provided the concept of ambient-oriented modeling approach. The proposed ambient-oriented modeling unlike its base article doesn’t distinguish between agent and ambient. In [136] the presented model treats any entity as an ambient whether it fulfills the definition of ambient or not. Agent-based modeling is mature and have well developed frameworks, software and libraries as discussed in previous sections. So, modeling agent-based ambient-oriented systems needs framework to properly distinguish agents and ambient, elaborating the process, simulating the model, representing their relations in model and representation of messages.

### 4.5.2 Our Framework for Modeling Agent-based Ambient-oriented Systems

Our first research question is that how complex systems having agents that contain other agents and have the ability to move within a limited location can be modeled? So, in other words we can write it as how can agent-based, ambient-oriented systems be modeled? The second research question is that how can one represent different agents of different levels based on their dependencies? So, we can write this question as how can we represent the inclusion relations of ambient and agents? The third question is that how can we add details like message sending or receiving by certain agents in link with the representation of dependencies? We can write it as; how can we represent messages along with relations of ambient and agents?

Ambient always own three properties i.e., inclusion, narrowness, and mobility. Inclusion means an ambient is a container which can contain other entities. Narrowness means that it should have certain boundaries. Mobility means that an ambient will always have the ability to change the location. When we analyze the means of transportation like airplanes, trains, buses and cars, they may be treated as ambient. An airplane is an ambient which moves from one airport to the other and also contains passengers, and crew staff. A bus is an ambient as it can move from one location to the other and passengers can ride on it. A train is an ambient as it moves from station to the station containing people and luggage. A car is an ambient as it can move from one place to the other including the driver and passengers. A ship is an ambient as it can move from one port to the other containing people and luggage. Similarly, the trucks are the ambient and may contain different agents.

We have discussed different types of agents in previous sections. All other agents who don’t fulfill the properties of ambient should be treated as agents. An agent may be an ambient at certain granularity level whereas at other level it may not. For example, while modeling the bus systems the human riding on bus will be an agent. Whereas in other case human may also contain some agents like mobile, smart watch etc., and have ability to move. An agent should have some role in the model. Depending on the availability of data one may select any of the agent-based modeling type. There are different frameworks/protocols available for agent-based modeling. These frameworks may be extended for adding the ambient features.

A model is based on some entities and the rules for processes. Usually the rules of agent-based modeling are written in code for simulations. In some simulation environments such as Anylogic the rules are written as flow diagrams and have visual representations or rules. In ambient-oriented modeling the rules are represented as processes. Processes contain messages as well. To use agent-based modeling incombination with ambient-oriented modeling, the rules should be defined in simulations and explanation should be provided in the form of ambient-oriented modeling expressions. Agent-based modeling is very effective for ad hoc systems and continuous time simulations. Since, ambient-oriented is majorly state-based and can be effectively used for event-based properties of systems. Our proposed framework will be useful for modeling complex systems that have both agents and ambient as their components.

With the emergence of new technologies, modern transport systems are rapidly changing. Internet based applications are facilitating for the transportation as well. Internet based transport systems like Uber and Careem are operating in multiple countries. Internet of Things based goods tracking applications are also proposed. SWVL is a service operating in different countries which provides internet based routed public transport. These types of systems need modeling for different purposes. Hence, our modeling approach will help in modeling such transport systems. It will help in modeling both urban as well as long distance transport systems. There are various agent-based simulation environments. Although we used Anylogic, however other tools may also be used. Some of the famous agent-based modeling tools are NetLogo, AgentCell, AgentFactory, Brahms, Urbansim and Swarm. However, addition of ambient properties to these tools will be helpful in representation of ambient and modeling agent-based ambient-oriented systems.

## 4.6 Unified Framework

A digital twin refers to a virtual representation of a physical system, product, or process, created using real-time data and other information. It allows organizations to simulate and analyze the behavior of their physical systems, processes, and products, helping them optimize operations and make informed decisions. Digital twins are widely used across a range of industries, including manufacturing, healthcare, and smart cities.

Manufacturing: In the manufacturing industry, digital twins are used to simulate and optimize production lines. For example, a digital twin of a production line can help identify bottlenecks in the production process and predict maintenance needs. This can result in reduced downtime, increased efficiency, and improved product quality.

Healthcare: In healthcare, digital twins can be used to model patients, enabling healthcare professionals to simulate treatment options and make informed decisions. For example, a digital twin of a patient with a complex medical condition can help healthcare professionals understand the impact of different treatments and make predictions about the patient's future health.

Smart Cities: In smart cities, digital twins can be used to simulate and optimize the performance of urban infrastructure and services, such as transportation systems and energy grids. For example, a digital twin of a city's transportation system can help city planners optimize routes, reduce congestion, and improve energy efficiency. Buildings: In the building industry, digital twins are used to model and optimize the performance of buildings. For example, a digital twin of a building can helps building owners and operators understand energy usage, predict maintenance needs, and improve indoor air quality.

A digital twin is a virtual representation of a physical object or system, and it can be used in the context of Internet of Things (IoT) systems to better understand and optimize the behavior and performance of these systems. The different modeling approaches, such as agent-based modeling (ABM), service-oriented modeling (SOM), contract-based modeling, network-based modeling, and fuzzy logic modeling, can be used to create digital twins for IoT systems.

For example, ABM can be used to model the behavior and interactions of individual devices or components within an IoT system, and a digital twin of an IoT system can be created by aggregating the individual digital twins of its components. SOM can be used to model the services provided by an IoT system and the relationships between these services, and the digital twin can be used to represent and monitor these services. Contract-based modeling can be used to specify the relationships and interactions between different components of an IoT system, and the digital twin can be used to represent and monitor these relationships. Network-based modeling can be used to model the communication and data exchange between different components of an IoT system, and the digital twin can be used to represent and monitor these interactions. Fuzzy logic modeling can be used to model the uncertainty and imprecision inherent in IoT systems, and the digital twin can be used to represent and monitor these uncertainties. By using these different modeling approaches, the digital twin of an IoT system can provide a comprehensive and detailed representation of the system, and it can be used to analyze, optimize, and control the system.

Additionally, the use of digital twins can help to reduce the gap between the physical and virtual worlds, and it can improve the understanding and management of complex IoT systems.

### 4.6.1 Case Study of Smart Public Transportation

Imagine a public transportation system that consists of a fleet of buses, each equipped with IoT devices to monitor and control various aspects of their operations. To model such a system, we can use a combination of various modeling approaches, including agent-based modeling, network-based modeling, contract-based modeling, ambient-oriented modeling, and service-oriented modeling.

Agent-based modeling can be used to represent the different agents involved in the system, such as the buses, passengers, and bus drivers. The buses can be modeled as mobile agents that can move from one place to another and interact with passengers and drivers. Passengers can be modeled as passive agents who have certain needs and preferences, such as the desire to reach their destination on time and in comfort. Drivers can be modeled as active agents who take actions based on their own goals and the information they receive from the environment.

Network-based modeling can be used to represent the connections between the agents and the environment. For example, the connections between the buses and the bus stops can be represented as edges in a graph, while the bus stops themselves can be represented as nodes. The connections between the passengers and the buses can also be represented in this way. By using network-based modeling, it is possible to analyze the flow of information and resources within the system and to identify bottlenecks and inefficiencies.

Contract-based modeling can be used to specify the obligations and responsibilities of the different agents. For example, the contracts between the passengers and the bus company can specify the level of service that the passengers can expect, such as the frequency of bus arrivals and departures, the quality of the buses, and the speed of the journeys. The contracts between the bus company and the bus drivers can specify the working conditions and the incentives for good performance. By using contract-based modeling, it is possible to ensure that the different agents are aligned in their objectives and that the system is fair and transparent.

Ambient-oriented modeling can be used to represent the context in which the agents operate. For example, the ambient of a bus can be modeled as the physical and social environment in which it operates, such as the route it takes, the weather conditions, and the behavior of the passengers. The ambient can also include information about the traffic conditions, the road conditions, and the locations of other vehicles and obstacles. By using ambient-oriented modeling, it is possible to capture the complexity of the real-world environment and to simulate the interactions between the agents and the environment.

In addition to ambient-oriented modeling, the bus system can also be modeled using aspect-oriented modeling. This means that the system can be divided into different aspects, such as scheduling, monitoring, and maintenance. Each aspect can be considered as a modular unit that can be developed, tested, and maintained independently, allowing for a more flexible and adaptable system. For example, the scheduling aspect can include the planning and management of the bus routes and schedules, taking into account factors such as traffic conditions and passenger demand. The monitoring aspect can include the real-time monitoring of the bus and its passengers, allowing for quick and effective response in case of emergencies. The maintenance aspect can include the management of the bus fleet, including the scheduling of regular maintenance and repairs, as well as the replacement of worn-out parts.

In addition to aspect-oriented modeling, the bus system can also be modeled using service-oriented modeling. This means that the system can be viewed as a set of services that can be offered to the passengers, such as online ticket booking, real-time tracking, and on-board entertainment. These services can be offered to the passengers through various channels, such as mobile applications, websites, and onboard displays. For example, the online ticket booking service can allow passengers to book their tickets in advance, eliminating the need to wait in long queues at the ticket counters. The real-time tracking service can allow passengers to track the location of their bus in real-time, helping them plan their journey accordingly. The on-board entertainment service can include features such as Wi-Fi, charging ports, and entertainment systems, providing a comfortable and convenient experience for the passengers.

Moreover, we can also add fuzzy logic modeling to the system, which will allow us to capture the uncertainty and vagueness of real-world data and decisions. For example, the bus driver’s decision-making process can be modeled using fuzzy logic, where the driver can make decisions based on a set of fuzzy rules that take into account various factors such as traffic conditions, road conditions, passenger preferences, and safety concerns. The deployment of fuzzy logic can also help to optimize the efficiency of the bus system by making more informed decisions based on real-world data.

In conclusion, the combination of different modeling approaches including agent-based modeling, network-based modeling, contract-based modeling, ambient-oriented modeling, service-oriented modeling, aspect-oriented modeling, and fuzzy logic modeling can provide a comprehensive and detailed simulation of the public transportation system. The simulation can help to optimize the design and operation of the public transportation system by considering all the different aspects and concerns involved, from the passengers and drivers to the weather conditions and traffic patterns. The simulation can also help to ensure that the public transportation system provides a high level of service to the passengers while also being efficient, secure, and sustainable.

### 4.6.2 Case Study of Smart Home

A Smart Home IoT system can be used as an example to show how different modeling approaches can be integrated. Consider a smart home system where different devices and sensors are connected to monitor and control various aspects of the home environment such as temperature, lighting, security, and appliances. The smart home system has to operate in real-time and handle multiple tasks simultaneously. To achieve this, the smart home system can use a combination of different modeling approaches.

Agent-based modeling: Agents can be used to represent different devices and sensors in the smart home system. The agents can be programmed to act autonomously and make decisions based on the data received from the sensors. For example, an agent for temperature control can be programmed to adjust the temperature of the room based on the data received from temperature sensors. The agents can also communicate with each other to exchange information and coordinate their actions. Ambient-oriented modeling: The smart home system can use ambient-oriented modeling to provide a more details about ambient in system. For example, there may be a wheelchair in the home. The wheelchair can be modeled as an ambient. Similarly, food serving trolly is also ambient.

Fuzzy logic modeling: Fuzzy logic can be used to handle uncertainty in the data received from the sensors. For example, the smart home system can use fuzzy logic to determine the temperature of the room when multiple temperature sensors are installed. The fuzzy logic system can use the data from all the sensors to calculate the average temperature of the room, taking into account the reliability of each sensor.

Contract-based modeling: Contracts can be used to define the interactions between the different devices and sensors in the smart home system. The contracts can specify the actions to be taken by the devices and sensors under different circumstances. For example, a contract between a temperature sensor and a temperature control agent can specify the actions to be taken when the temperature goes above or below a certain threshold.

Aspect-oriented modeling: The smart home system can use aspect-oriented modeling to separate the cross-cutting concerns from the main functionality of the system. For example, the security aspect of the smart home system can be separated from the temperature control aspect. This will make it easier to maintain and modify the system as the security aspect will not affect the temperature control aspect, and vice versa.

Network-based modeling: The smart home system can use network-based modeling to represent the communication between the different devices and sensors in the home. The devices and sensors can be represented as nodes in the network, and the communication between them can be represented as edges. For example, the temperature sensor can be represented as a node, and the communication between the temperature sensor and the temperature control agent can be represented as an edge.

In this scenario, the different modeling approaches can be used together to provide a robust, flexible, and scalable solution for the smart home system. The agent-based modeling provides the autonomy and decision-making capability for the devices and sensors, while the ambient-oriented modeling provides a natural and intuitive interface to the users. The fuzzy logic modeling handles uncertainty in the data, while the contract-based modeling provides a framework for defining the interactions between the devices and sensors. The aspect-oriented modeling separates the cross-cutting concerns, and the network-based modeling provides a representation of the communication between the devices and sensors.

In this scenario, let’s consider an IoT system for Smart Home automation. The system consists of multiple smart devices such as smart lights, smart thermostats, smart locks, and smart cameras. These devices are connected to the internet and can be controlled remotely through a smartphone app or a web interface.

Agent-Based Modeling: In this system, each smart device can be considered as an individual agent with its own set of behaviors and functions. The agents interact with each other to perform specific tasks such as turning on the lights when someone enters the room, adjusting the thermostat based on the ambient temperature, and automatically locking the doors when everyone leaves the house. The behavior of each agent can be modeled using an agent-based modeling approach.

Ambient-Oriented Modeling: In this system, the ambient that are a special type of agent should also be considered. For example, a smart food serving trolly is also the part of the home. The trolly can move to different places at home such as kitchen, dining room and drawing room.

Fuzzy Logic Modeling: In some cases, the decision-making process for controlling the smart devices may not be straightforward. For example, the lighting system may need to turn on the lights in the evening when it starts to get dark, but the brightness should not be too high as to disturb someone who is sleeping in the room. In these cases, fuzzy logic modeling can be used to model the uncertainty and imprecise knowledge involved in the decision-making process.

Contract-Based Modeling: The interactions between the smart devices can be modeled using contracts in a contract-based modeling approach. The contracts can define the expected behavior of each agent and the rules for communication between the agents. For example, the smart locks and the smart cameras can have a contract that defines the rules for automatically unlocking the doors and turning on the cameras when someone enters the house.

Aspect-Oriented Modeling: The aspect-oriented modeling approach can be used to model the cross-cutting concerns in the IoT system. For example, security is a concern that affects multiple smart devices in the system. An aspect-oriented modeling approach can be used to define the security requirements for the system and ensure that the smart devices implement the security features consistently.

Network-Based Modeling: The smart devices in the IoT system can be modeled as nodes in a network-based modeling approach. The connections between the devices can be modeled as edges. The network-based modeling approach can be used to analyze the interactions and communication between the devices and optimize the network for better performance and reliability.

Service-oriented modeling: The smart home IoT system can also benefit from service-oriented modeling, where each smart device provides a specific service to the system, such as temperature control, lighting control, or security control. The services can be modeled as independent entities that can be composed and reused to provide new functionality, making it easier to modify and extend the system.

By using a combination of these modeling approaches, the smart home automation system can be modeled and analyzed to ensure that it meets the requirements and performs optimally. The models can be used to simulate the system and test different scenarios before deploying it in the real world. This helps to identify potential issues and improve the design of the system before it is deployed.

# Chapter 5

# CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

In this dissertation we have provided a unified framework for modeling service-oriented IoT systems. The unified framework is composed of three sub frameworks of modeling IoT systems. We provided the procedure for selecting any of the frameworks with respect to requirements. The research began with identifying the case studies of IoT systems. We examined different uses of IoT. After identification of different applications of IoT, we separated those applications in four different groups. We examined previous works done with respect to each group. Here, we provided a separate framework for each group. Every framework uses a combination of modeling approaches. We considered a total of eight modeling approaches. We used different simulation and design tools at different levels.

### 5.1.1 Contributions

This thesis has four major contributions. The contributions of the thesis are:

#### 5.1.1.1 For Software Engineering Purpose

Internet of Things is an emerging area connecting various domains and aimed at targeting almost every aspect of life. Everything is going to be connected to the internet and everything is expected to be smart in near future. These smart systems are highly dependent on artificial intelligence and connections. The increasing number of connected devices and their collaboration is demanding more and more complex systems. These complex systems require proper planning and guidelines for development. New scenarios are emerging on daily basis. Due to these reasons, Internet of Things systems development needs complete and correct models. However, previous modeling approaches may not be sufficient to effectively model such systems. In this study we have proposed a framework for complex IoT systems modeling for Software Engineering purpose. Our framework provides an effective way to model different types of systems and subsystems i.e. continuous, discrete, ad hoc, general, service-oriented systems including decomposition and composition. We have used well known elements of four modeling approaches i.e., agent-based modeling, aspect-oriented modeling, contract-based modeling, and service-oriented modeling. We validated the framework by using it for an IoT system scenario. We used different tools like Netlogo, Attributed graph grammar (AGG), XML and WSDL.

#### 5.1.1.2 For Fuzzy-values Purpose

Modeling complex systems is used for different purposes in engineering. For communication among stakeholders from different backgrounds models are helpful in understanding viewpoints. In IoE we have different types of devices and people to interact with. In such heterogeneous environments models are used to represent systems at different viewpoint levels. Human agents interacting with a system more often produce fuzzy values. Our proposed framework provides a mechanism to model IoE systems which have fuzzy values. The agents which produce fuzzy values are termed as fuzzy agents. Our framework provides a more detailed model using concepts of different modeling approaches. Moreover, it is helpful in building consistent and complete models as new agents can be created at composition stage and model can be revised accordingly.

#### 5.1.1.3 For Ambient Purpose

We discussed agent-based modeling and ambient-oriented modeling in detail. And then we discussed agents and ambient from different aspects as an agent is the basic unit of agent-based modeling while the ambient is the basic unit of ambient-oriented modeling. After this discussion, we provide the similarities between the agent and ambient. After a deep discussion of the agents and the ambient, we conclude that the ambient may be treated as a type of agent. However, in case of agent-based modeling and ambient-oriented modeling, both the approaches promote totally different ways of modeling. Agent-based modeling promotes visual representations and minimizes the use of mathematics. Ambient-oriented modeling is based on calculus and also has very low support for visual modeling. We then used an example of BRTS to elaborate their integrated use. With the help of example modeling, it was clear that a need arises to use both the techniques for same system at different levels of modeling. We presented the required steps for modeling modern transportation ways using an integrated approach. This modeling approach will be helpful in model-based engineering of transport systems that use information and communication technologies.

### 5.1.2 Comparisons

A hybrid modeling framework uses agent-based modeling (ABM), Discrete-event Simulations (DES) and System Dynamics (SD) in combination to model teamwork and ABM builds the frame along with modeling the internals of agents by DES and SD [242]. For modeling teamwork in engineering environment, ABM approach has been provided [243]. A framework for modeling freights has been proposed that overcomes the limitations of existing approaches [244]. A framework known as MESA has been presented and this framework for agent-based modeling [245]. A framework for modeling and simulation of emergent behaviors is presented [246]. A framework for modeling and simulation is proposed that addresses incident management on three axes i.e. incident, domain and life-cycle phase [247]. A traffic simulation framework to reproduce urban freight movements, particularly concerning double-parked delivery operations has been proposed [248]. A modeling and simulation framework has been provided to support a holistic analysis of healthcare systems through stratification of the levels of abstraction into multiple perspectives and their integration in a common simulation framework [249]. A framework for modeling different aspects of transportation system has been proposed and has used agent-based modeling in this framework [250]. A framework for modeling complex systems has been provided and this framework used the combination of network-based and agent-based modeling in combination [4]. A framework for modeling the security of Internet of Things systems has also been proposed [251].

There are several other frameworks that provide the use of agent-based modeling and other traditional modeling approaches for different specific purposes. However, there was lack of modeling framework that: provides the use of agent-based and ambient-oriented modeling approach in combination, provides representation of messages and processes that are associated with agents in agent-based modeling, provides a mechanism to model IoT systems that create or use fuzzy data, provides a mechanism to model IoT systems from viewpoint of Software Engineering, and provides a unification mechanism.

### 5.1.3 Limitations

We have applied our framework on few scenarios. As the use of IoT is increasing, we expect emergence of new scenarios. Hence, there may appear new combinations of technologies which would demand for modeling form different aspect.

## 5.2 Future Work

As we provide a unified framework for modeling complex IoT systems. IoT is an emerging field and IoT systems are using different technologies as well. In future more technologies can be incorporated in IoT systems to get more benefits from emerging technologies. The incorporation of new technologies can emerge the need of new modeling perspectives. Also, the existing tools may not be able to model a complete system according to different perspectives. Hence, improvement in tools is also the need of the day.

### 5.2.1 Framework

In future we aim at extending the framework that includes ambient in IoT systems for implementation in relevant scenarios like Internet of Things supply-chain. We also aim at focusing on the concept of device-to-device service provision and modeling digital currency payment method for IoT. There will be set of agents on the basis of services they provide. These agents will be some IoT devices. Every agent will have some contracts. There will be rules for modeling. We also intend to use Blockchain usage in IoT environment as a case study. These will help in defining mechanism of Visual smart contracts management and control. Also, it will provide a way to model mechanisms to monitor Visual smart contracts.

### 5.2.2 Tools

However, in future we aim on working for a unified tool based on this framework. We used

WSDL for devices by using device as a portType. There is a need to customize WSDL to service description language for devices and provide alternate names to the tags such as portType to Device. Similarly, SOAP needs to be updated as per IoT because here a physical object provides service. Also, binding of simple object in web service differs from binding of a physical object in IoT service.

# REFERENCES


[1] Rolf H Weber. Internet of things need for a new legal environment. *Computer Law & Security Review*, 25(6):522–527, 2009.

[2] W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand. Using contract based component specifications for virtual integration testing and architecture design. In *Proceedings of Design, Automation Test in Europe Conference Exhibition*, 2011.

[3] Miao Yun and Bu Yuxin. Research on the architecture and key technology of internet of things applied on smart grid. In *International Conference on Advances in Energy Engineering*, 2011.

[4] Komal Batool and Muaz A. Niazi. Modeling the internet of things a hybrid modeling approach using complex networks and agent based models. *Complex Adapt Syst Model*, 5(4):19 pages, 2017.

[5] Glenn I Hawe, Graham Coates, Duncan T Wilson, and Roger S Crouch. Agent based simulation for large-scale emergency response a survey of usage and implementation. *ACM Computing Surveys*, 45(1):51, 2012.

[6] William N Robinson and Yi Ding. A survey of customization support in agent based business process simulation tools. *ACM Transactions on Modeling and Computer Simulation*, 20(3):29, 2010.

[7] John A. Paravantis. *From Game Theory To Complexity Science and Agent Based Modeling in World Politics*. Springer, Berlin, 2016.

[8] Matteo Richiardi. The future of agent based modelling. *Eastern Econ J*, 43:24 pages, 2017.

[9] Giuseppe Bruno, Andrea Genovese, and Antonino Sgalambro. An agent based framework for modeling and solving location problems. *Sociedad de Estadistica e Investigacion Operativa*, 18:81–96, 2010.

[10] Shanmuganathan Vasanthapriyan and Selvarajah Thuseethan. Prediction of human flow in disaster situations a multi agent based modelling and simulation. In *2nd International Symposium on Dependable Computing and Internet of Things*, 2015.

[11] Nadia Creignou, Odile Papini, Stefan Rummele, and Stefan Woltran. Belief merging within fragments of propositional logic. *ACM Transactions on Computational Logic*, 17(3):28 pages, 2016.

[12] Todorka Glushkova and Stanimir Stoyanov. Ambient oriented modeling of intelligent context aware systems. *Bulgarian Computer Science And Communications*, 7(1):53–61, 2018.

[13] S Sehili, Laurent Capocchi, Jean Francois Santucci, Stephane Lavirotte, and Jean Yves Tigli. Discrete event modeling and simulation for iot efficient design combining wcomp and devsimpy framework. In *5th International Conference on Simulation and Modeling Methodologies Technologies and Applications*, 2015.

[14] Werner Retschitzegger, Wieland Schwinger, and Elisabeth Kapsammer. A survey on uml based aspect-oriented design modeling. *ACM Computing Surveys*, 43(4):59, 2011.

[15] Christina Chavez, Alessandro Garcia, Uira Kulesza, Claudio SantAnna, and Carlos Lucena. Crosscutting interfaces for aspect-oriented modeling. Journal of the Brazilian Computer Society, 12(1):43–58, 2006.

[16] Jorge Fox. A formal foundation for aspect oriented software development. In Memoria del XIV Congreso Internacional de Computacion CIC, 2005.

[17] Jingjun Zhang, Yuejuan Chen, Yang Zhang, and Hui Li. Aspect oriented modeling and mapping driven by model driven architecture. In 2nd IEEE International Conference on Computer Science and Information Technology, 2009.

[18] Jingyong Liu and Lichen Zhang. Qos modeling for cyber-physical systems using aspectoriented approach. In Second International Conference on Networking and Distributed Computing, 2011.

[19] Armin Wasicek, Patricia Derler, and Edward A Lee. Aspect oriented modeling of attacks in automotive cyber physical systems. In ACM DAC 14 San Francisco CA USA, 2014.

[20] Tomas Cerny. Aspect oriented challenges in system integration with microservices soa and iot. Enterprise Information Systems, page 23 pages, 2018.

[21] Senthil Murugan Balakrishnan and Arun Kumar Sangaiah. Chapter 6 - aspect oriented modeling of missing data imputation for internet of things (iot) based healthcare infrastructure. In Arun Kumar Sangaiah, Michael Sheng, and Zhiyong Zhang, editors, Computational Intelligence for Multimedia Big Data on the Cloud with Engineering

Applications, Intelligent Data-Centric Systems, pages 135 – 145. Academic Press, 2018.

[22] Xu Wang, Xuan Zhang, Tong Li, Junhui Liu, and Qingyi Chen. Correctness of aspect oriented business process modeling. Business Process Management Journal, 24(2):537–566, 2018.

[23] Meriem Chibani, Brahim Belattar, and Abdelhabib Bourouis. Practical benefits of aspect oriented programming paradigm in discrete event simulation. Hindawi Modelling and Simulation in Engineering, 2014:16 pages, 2014.

[24] Aws A. Magableh. Systematic review of aspect oriented formal method. Int. J. Computer Applications in Technology, 56(2):132–140, 2017.

[25] D. Mouheb, D. Alhadidi, M. Nouh, M. Debbabi, L.Wang, and M. Pourzandi. Aspect oriented modeling framework for security hardening. Innovations Syst Softw Eng, page 27 pages, 2015.

[26] A. Schauerhuber, W. Schwinger, E. Kapsammer, W. Retschitzegger, M. Wimmer, and G. Kappel. A survey on aspect oriented modeling approaches. Technical report, Business Informatics Group Vienna University of Technology Vienna Austria, 2007.

[27] Hui Li, Jingjun Zhang, and Yuejuan Chen. Aspect oriented modeling in software architecture pattern based on uml. In The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010.

[28] Bowei Chen, Jun Wang, Ingemar J. Cox, and Mohan S. Kankanhalli. Multi keyword multi click advertisement option contracts for sponsored search. ACM Transactions on Intelligent Systems and Technology, 7(1):29 pages, 2015.

[29] Anna Ye Du, Sanjukta Das, Ram D Gopal, and R Ramesh. Risk hedging in storage grid markets do options add value to forwards. ACM Transactions on Management Information Systems, 2(2):23 pages, 2011.

[30] Tien Dung Cao, Tran Vu Pham, Quang Hieu Vu, Hong Linh Truong, Duc Hung Le, and Schahram Dustdar. Marsa a marketplace for realtime human sensing data. ACM Transactions on Internet Technology, 16(3):21 pages, 2016.

[31] T. Stephen Strickland, Christos Dimoulas, Asumu Takikawa, and Matthias Felleisen. Contracts for first class classes. ACM Transactions on Programming Languages and Systems, 35(3):58 pages, 2013.

[32] Christos Dimoulas and Matthias Felleisen. On contract satisfaction in a higher order world.

ACM Transactions on Programming Languages and Systems, 33(5):29 pages, 2011.

[33] Jooyong Yi, Dawei Qi, Shin Hwei Tan, and Abhik Roychoudhury. Software change contracts. ACM Transactions on Software Engineering and Methodology, 24(3):43 pages, 2015.

[34] Xin Zhang, Tomas Ward, and Seamus Mcloone. Comparison of predictive contract mechanisms from an information theory perspective. ACM Transactions on Multimedia Computing, Communications and Applications, 8(2):18 pages, 2012.

[35] Giuseppe Castagna, Nils Gesbert, and Luca Padovani. A theory of contracts for web services. ACM Transactions on Programming Languages and Systems, 31(5):61 pages, 2009.

[36] Thi Thieu Hoa Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. Contract based requirement modularization via synthesis of correct decompositions. ACM Transactions on Embedded Computing Systems, 15(2):26 pages, 2016.

[37] Roger Barga, David Lomet, German Shegalov, and Gerhard Weikum. Recovery guarantees for internet applications. ACM Transactions on Internet Technology, 4(3):289–328, 2004.

[38] Victoria Ungureanu. Using certified policies to regulate e commerce transactions. ACM Transactions on Internet Technology, 5(1):129–153, 2005.

[39] Iulia Dragomir, Iulian Ober, and Christian Percebois. Contract based modeling and verification of timed safety requirements within sysml. Software and Systems Modeling, 16(2):687–624, 2015.

[40] Huansheng Ning, Qingjuan Li, Dawei Wei, Hong Liu, and Tao Zhu. Cyberlogic paves the way from cyber philosophy to cyber science. Ieee Internet Of Things Journal, 4(3):783–790, 2017.

[41] R Jayachandran, S Denis Ashok, and S Narayanan. Fuzy logic based modeling and simulation approach for the estimation of the tire forces. In Procedia Engineering 64, pages 1109–1118, 2013.

[42] Amilcare Francesco Santamaria, Pierfrancesco Raimondo, Floriano De Rango, and Abdon Serianni. A two stages fuzzy logic approach for internet of things wearable devices. In IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC): Workshop Internet of Things for Ambient Assisted Living (IoTAAL), 2016.

[43] Abdulmonam Al Adhab, Hamad Altmimi, Majed Alhawashi, Hussain Alabduljabbar, Farah Harrathi, and Hammeed ALmubarek. Iot for remote elderly patient care based on fuzzy logic. In IEEE International Symposium on Networks, Computers and Communications (ISNCC), 2016.

[44] R Babuska and H B Verbruggen. An overview of fuzzy modeling for control. Control Eng. Practice, 4(11):1593–1606, 1996.

[45] Mohammad R Emami, I. Burhan Turksen, and Andrew A Goldenberg. Development of a systematic methodology of fuzzy logic modeling. Ieee Transactions On Fuzzy Systems, 6(3):346– 361, 1998.

[46] S Sankar and P Srinivasan. Fuzzy logic based energy aware routing protocol for internet of things. I.J. Intelligent Systems and Applications, 10:11–19, 2018.

[47] Y P Tsang, K L Choy, C H Wu, G T S Ho, Cathy H Y Lam, and P S Koo. An internet of things based risk monitoring system for managing cold supply chain risks. Industrial Management and Data Systems, 118(7):1432–1462, 2018.

[48] Jung Hyok Kwon, Minki Cha, Sol Bee Lee, and Eui Jik Kim. Variable categorized clustering algorithm using fuzzy logic for internet of things local networks. Multimedia Tools and Applications, 2017.

[49] Daniel Meana Llorian, Cristian Gonzalez Garcia, B Cristina Pelayo G Bustelo, Juan Manuel Cueva Lovelle, and Nestor Garcia Fernandez. Iofclime the fuzzy logic and the internet of things to control indoor temperature regarding the outdoor ambient conditions. Future Generation Computer Systems, page 10 pages, 2016.

[50] Navid Khademi, Mojtaba Rajabi, Afshin S Mohaymany, and Mahdi Samadzad. Day to day travel time perception modeling using an adaptive network based fuzzy inference system. EURO J Transp Logist, 5:25–52, 2016.

[51] Abhinav Shukla and Neelam Sahu. Development of fuzzy logic based internet of things technology for fire monitoring system residential fire. International Journal of Recent Scientific Research, 9(10):29089–29105, 2018.

[52] Muaz Ahmed Khan Niazi. Towards A Novel Unified Framework for Developing Formal Network and Validated Agent Based Simulation Models of Complex Adaptive Systems. PhD thesis, Computing Science and Mathematics School of Natural Sciences University of Stirling Scotland UK, 2011.

[53] Vladimir Boginski, Panos M Pardalos, and Alkis Vazacopoulos. Network based Models and Algorithms in Data Mining and Knowledge Discovery. Handbook Of Combinatorial Optimization. Kluwer Academic Publishers, 5 edition, 2004.

[54] Wang Guanghui, Wang Yufeiz, Liu Yijun, and Chi Yuxue. An overview of structurally complex network based modeling of public opinion in the we the media era. International Journal of Modern Physics B, 32:22 pages, 2018.

[55] Josh Czemeresand Kurt Buse and Gennady M Verkhivker. Atomistic simulations and network based modeling of the hsp90 cdc37 chaperone binding with cdk4 client protein a mechanism of chaperoning kinase clients by exploiting weak spots of intrinsically dynamic kinase domains. PLoS ONE, 12(12):34 pages, 2017.

[56] Ammar Naqvi, Huzefa Rangwala, Ali Keshavarzian, and Patrick Gillevet. Network based modeling of the human gut microbiome. Chemistry And Biodiversity, 7:1040–1050, 2010.

[57] Yingzhuo Wei, Shaowu Zhang, Chunhui Zhao, Feng Yang, and Quan Pan. Network based modeling for analyzing the human skin microbiome. In IEEE International Conference on Bioinformatics and Biomedicine Workshops, 2010.

[58] Daning Hu, J. Leon Zhao, Zhimin Hua, and Michael C. S. Wong. Network based modeling and analysis of systemic risk in banking systems. MIS Quarterly, 36(4):1269–1291, 2012.

[59] Nobuya Okami, Yasuo Aihara, Hiroyuki Akagawa, Koji Yamaguchi, Akitsugu Kawashima, Toshiyuki Yamamoto, and Yoshikazu Okada. Network based gene expression analysis of vascular wall of juvenile moyamoya disease. Childs Nerv Syst, 31:399–404, 2015.

[60] Zhenghui Sha, Yun Huang, Jiawei Sophia Fu, Mingxian Wang, Yan Fu, Noshir Contractor, and Wei Chen. A network based approach to modeling and predicting product coconsideration relations. Complexity, 2018:14 pages, 2018.

[61] Denghui Zhang, Zhengxu Zhao, Yiqi Zhou, and Yang Guo. A novel complex network based modeling method for heterogeneousproduct design. Cluster Computing, page 12 pages, 2017.

[62] Yaroslav V Zolotukhin, Oleg A Markelov, and Mikhail I Bogachev. A network based approach to the analysis of geomagnetic fluctuations. In IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2017.

[63] JH Powell. Powergraph a network based approach to modelling and managing corporate strategic conict and cooperation. Journal of the Operational Research Society, 50:669–683,

1999.

[64] Yan tao Wang, Jia guang Liu, and Yong sheng Chai. Network based negotiation model for components suppliers. In 11th International Conference on Computer Supported Cooperative Work in Design, 2007.

[65] Max Menenberg, Surya Pathak, Hari P Udyapuram, Srinagesh Gavirneni, and Sohini Roychowdhury. Topic modeling for management sciences a network based approach. In IEEE International Conference on Big Data, 2016.

[66] Altug Akay, Andrei Dragomir, and Bjorn Erik Erlandsson. Network based modeling and intelligent data mining of social media for improving care. IEEE Journal Of Biomedical And Health Informatics, 19(1):210–218, 2015.

[67] Chao Gao and Jiming Liu. Network based modeling for characterizing human collective behaviors during extreme events. IEEE Transactions On Systems, Man, And Cybernetics Systems, 47(1):171–183, 2017.

[68] Hui Cheng and Xuening Chu. A network based assessment approach for change impacts on complex product. J Intell Manuf, 23:1419–1431, 2012.

[69] Wei Chen, Babak Heydari, Anja M Maier, and Jitesh H Panchal. Network based modeling and analysis in design. Design Science, 4:4 pages, 2018.

[70] Huijuan Zhang and Miao Zhang. Complex network based growth and evolution model for internet of things. In IEEE 5th International Conference on Software Engineering and Service Science, 2014.

[71] https://www.techopedia.com/definition/28584/object-oriented-modeling-oom.

[72] Ruchi Shukla, W A Clarke, and T Marwala. Object oriented modeling framework of a kohonen network based character recognition system. In International Conference on Computer Communication and Informatics, 2012.

[73] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things a survey. Ieee communications surveys & tutorials, 16(1):414–454, 2014.

[74] Robert Kok, Tania R Lanphere, Bioresource Engineering, and McGill University. Object based modeling and simulation what why and how. In CSBE Workshop Modeling Material and Energy Flows Through Agro Ecosystems, 2006.

[75] Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. Towards a

development methodology for smart object oriented iot systems a metamodel approach. In IEEE International Conference on Systems Man and Cybernetics, 2015.

[76] Mauro Conti, Ali Dehghantanha, Katrin Franke, and Steve Watson. Internet of things security and forensics challenges and opportunities. Future Generation Computer Systems, 78:544–546, 2018.

[77] Chengen Wang, Zhuming Bi, and Li Da Xu. Iot and cloud computing in automation of assembly modeling systems. IEEE Transactions On Industrial Informatics, 10(2):1426–1434, 2014.

[78] Shifeng Fang, Lida Xu, Yunqiang Zhu, Yongqiang Liu, Zhihui Liu, Huan Pei, Jianwu Yan, and Huifang Zhang. An integrated information system for snowmelt flood early warning based on internet of things. Inf Syst Front, 17:321–335, 2015.

[79] Pavel Vrba, Vladimir Marik, Pierluigi Siano, Paulo Leitao, Gulnara Zhabelova, Valeriy Vyatkin, and Thomas Strasser. A review of agent and service oriented concepts applied to intelligent energy systems. IEEE Transactions On Industrial Informatics, 10(3):1890–1903, 2014.

[80] Ranjit Kumar Behera, K. Hemant Kumar Reddy, and Diptendu Sinha Royb. Modeling and assessing reliability of service oriented internet of things. International Journal of Computers and Applications, 2018.

[81] Jack C P Cheng, Kincho H Law, Hans Bjornsson, Albert Jones, and Ram Sriram. A service oriented framework for construction supply chain integration. Automation in Construction, 19:245–260, 2010.

[82] Jun Huang, Donggai Du, Qiang Duan, Yi Sun, Ying Yin, Tiantian Zhou, and Yanguang Zhang. Modeling and analysis on congestion control in the internet of things. In Adhoc and Sensor Networking Symposium, 2014.

[83] Boukaye Boubacar Traore, Bernard Kamsu Foguem, Fana Tangara, and Xavier Desforges. Service-oriented computing for intelligent train maintenance. Enterprise Information Systems, 2018.

[84] I Ling Yen, Farokh Bastani, San Yih Hwang, Wei Zhu, and Guang Zhou. From software services to iot services the modeling perspective. In International Conference on Serviceology, 2017.

[85] Florian Rademacher, Sabine Sachweh, and Albert Zundorf. Analysis of service oriented

modeling approaches for viewpoint specific model driven development of microservice architecture. CoRRRademacher2018, abs/1804.09946, 2018.

[86] Andrea Delgado, Francisco Ruiz, Ignacio Garcia Rodriguez de Guzman, and Mario Piattini. Towards an ontology for service oriented modeling supporting business processes. In Fourth International Conference on Research Challenges in Information Science, 2010.

[87] Yuran Jin and Shoufeng Ji. Mapping hotspots and emerging trends of business model innovation under networking in internet of things. EURASIP Journal on Wireless Communications and Networking, 96:12 pages, 2018.

[88] Hyojik Lee, Onechul Na andYanghoon Kim, and Hangbae Chang. A study on designing public safety service for internet of things environment. Wireless Pers Commun, 93:447–459, 2017.

[89] Leon Lim, Pierrick Marie, Denis Conan, Sophie Chabridon, Thierry Desprats, and Atif Manzoor. Enhancing context data distribution for the internet of things using qoc-awareness and attribute based access control. Ann. Telecommun, 71:121–132, 2016.

[90] Samer Machara, Sophie Chabridon, and Chantal Taconet. Trust based context contract models for the internet of things. In 10th IEEE International Conference on Ubiquitous Intelligence Computing, 2013.

[91] Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. Integration of agent based and cloud computing for the smart objects oriented iot. In 18th IEEE International Conference on Computer Supported Cooperative Work in Design, 2014.

[92] Yu Zhang, Yuxinghan, and Jiangtao Wen. Smer a secure method of exchanging resources in heterogeneous internet of things. Front. Comput. Sci., page 12 pages, 2018.

[93] Claudio Savaglio, Giancarlo Fortino, and Mengchu Zhou. Towards interoperable cognitive and autonomic iot systems an agent based approach. In IEEE 3rd World Forum on Internet of Things, 2016.

[94] Pablo Pico Valencia, Juan A Holgado Terriza, Deiver Herrera Sanchez, and Jose Sampietro. Towards the internet of agents an analysis of the internet of things from the intelligence and autonomy perspective. Ingenieria e Investigacion, 38(1):121–129, 2018.

[95] Shingo Yamaguchi, Shoki Tsugawa, and Kazuya Nakahori. An analysis system of iot services based on agent oriented petri net pn. In International Conference on Consumer Electronics Taiwan, 2016.

[96] Giancarlo Fortino, Wilma Russo, and Claudio Savaglio. Agent oriented modeling and simulation of iot networks. In Federated Conference on Computer Science and Information Systems, 2016.

[97] Nicholas J. Kaminski, Maria Murphy, and Nicola Marchetti. Agent based modeling of an iot network. In IEEE International Symposium on Systems Engineering, 2016.

[98] Pablo Pico Valencia and Juan A. Holgado Terriza. Semantic agent contracts for internet of agents. In IEEE WIC ACM International Conference on Web Intelligence Workshops, 2016.

[99] Gabriele D Angelo, Stefano Ferretti, and Vittorio Ghini. Modeling the internet of things a simulation perspective. In International Conference on High Performance Computing & Simulation, 2017.

[100] Alexander Kuzmin. Blockchain based structures for a secure and operate iot. In IEEE Conference on Internet of Things Business Models Users and Networks, 2017.

[101] Oscar Novo. Blockchain meets iot an architecture for scalable access management in iot. IEEE Internet Of Things Journal, 5(2):1184–1195, 2018.

[102] Jan Kramer, Jan Martijn E. M. van der Werf, Johan Stokking, and Marcela Ruiz. A blockchain based micro economy platform for distributed infrastructure initiatives. In IEEE International Conference on Software Architecture, 2018.

[103] Ricardo Neisse, Gianmarco Baldini, Gary Steri, Yutaka Miyake, Shinsaku Kiyomoto, and Abdur Rahim Biswas. An agent based framework for informed consent in the internet of things. In IEEE 2nd World Forum on Internet of Things, 2015.

[104] Florin Bogdan Balint and Hong Linh Truong. On supporting contract aware iot dataspace services. In 5th IEEE International Conference on Mobile Cloud Computing Services and Engineering, 2017.

[105] C. Savaglio, G. Fortino, M. Ganzha, M. Paprzycki, C. Badica, and Ivanovic M. Agent-Based Computing in the Internet of Things a Survey. Springer, Cham, 2018.

[106] Torsten Spieldenner, Sergiy Byelozyorov, Michael Guldner, and Philipp Slusallek. Fives an aspect oriented virtual environment server. In International Conference on Cyberworlds, 2017.

[107] Senthil Murugan Balakrishnan and Arun Kumar Sangaiah. Mifim middleware solution for service centric anomaly in future internet models. Future Generation Computer Systems,

74:449– 465, 2017.

[108] Senthil Murugan Balakrishnan and Arun Kumar Sangaiah. Aspect oriented middleware for internet of things a state of the art survey of service discovery approaches. International Journal of Intelligent Engineering and Systems, 8(4):16–28, 2015.

[109] Igor Kotenko, Igor Saenko, and Sergey Ageev. Countermeasure security risk management in the internet of things based on fuzzy logic inference. In IEEE Trustcom, 2015.

[110] Mengru Tu, Ming K Lim, and Ming Fang Yang. Iot based production logistics and supply chain system modeling iot-based manufacturing supply chain. Industrial Management & Data Systems, 118(1):65–95, 2018.

[111] Giancarlo Fortino, Wilma Russo, Claudio Savaglio, Mirko Viroli, and MengChu Zhou. Modeling opportunistic iot services in open iot ecosystems. In 17th Workshop From Objects to Agents WOA, 2017.

[112] James M. Tien. Internet of things real time decision making and artificial intelligence. Ann. Data. Sci., 4(2):149–178, 2017.

[113] JunPing Wang, ShiHui Duan, and YouKang Shi. Multi objects scalable coordinated learning in internet of things. Pers Ubiquit Comput, 19:1133–1144, 2015.

[114] Suparna De, Payam Barnaghi, Martin Bauer, and Stefan Meissner. Service modelling for the internet of things. In Federated Conference on Computer Science and Information Systems, 2011.

[115] Jose Creissac Campos, Camille Fayollas, Marcelo Goncalves, Celia Martinie, David Navarre, Philippe Palanque, and Miguel Pinto. A more intelligent test case generation approach through task models manipulation. Proc ACM Human Computer Interaction, 1:20 pages, 2017.

[116] Shancang Li, George Oikonomou, Theo Tryfonas, Thomas M Chen, and Li Da Xu. A distributed consensus algorithm for decision making in service oriented internet of things. IEEE Transactions On Industrial Informatics, 10(2):1461–1468, 2014.

[117] Yu Zhang and JiangtaoWen. The iot electric business model using blockchain technology for the internet of things. Peer-to-Peer Netw. Appl., 10:983–994, 2017.

[118] Gabriele D Angelo, Stefano Ferretti, and Vittorio Ghini. Simulation of the internet of things. In International Conference on High Performance Computing & Simulation, 2016.

[119] Pablo Chamoso, Fernando De la Prieta, Francisco De Paz, and Juan M. Corchado. Swarm

agent based architecture suitable for internet of things and smartcities. In 12th Int. Conference on Advances in Intelligent Systems and Computing, 2015.

[120] I Ling Yen, Farokh Bastani, Wei Zhu, Hessam Moeini, San Yih Hwang, and Yuqun Zhang. Service oriented iot modeling and its deviation from software services. In IEEE Symposium on Service-Oriented System Engineering, 2018.

[121] Wenwu Tang, Shaowen Wang, David A Bennett, and Yan Liu. Agent based modeling within a cyberinfrastructure environment a service oriented computing approach. International Journal of Geographical Information Science, 25(9):1323–1346, 2011.

[122] James Odell, H Van Dyke Parunak, and Mitchell Fleischer. Modeling agents and their environment the communication environment. Journal of Object Technology, 2(1):39–52, 2003.

[123] David Chelberg, LonnieWelch, Arvind Lakshmikumar, Matthew Gillen, and Qiang Zhou. Meta reasoning for a distributed agent architecture. In IEEE 33rd Southeastern Symposium on System Theory, pages 377–381, 2001.

[124] Amy H Auchincloss and Leandro Martin Totaro Garcia. Brief introductory guide to agent based modeling and an illustration from urban health research. Cadernos de saude publica, 31:65–78, 2015.

[125] Joris Borgdorff, Mariusz Mamonski, Bartosz Bosak, Derek Groen, Mohamed Ben Belgacem, Krzysztof Kurowski, and Alfons G Hoekstra. Multiscale computing with the multiscale modeling library and runtime environment. In International Conference on Computational Science ICCS, volume 18 of Procedia Computer Science, pages 1097–1105, 2013.

[126] Dierk Raabe, Matthias Scheffler, Kurt Kremer, Walter Thiel, Jorg Neugebauer, and Martin Jansen. Multi scale modeling in materials science and engineering. Technical report, 2009.

[127] Alfons Hoekstra, Bastien Chopard, and Peter Coveney. Multiscale modelling and simulation a position paper. Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences, 372, 2014.

[128] Derek Groen, Jaroslaw Knap, Philipp Neumann, Diana Suleimenova, Lourens Veen, and Kenneth Leiter. Mastering the scales a survey on the benefits of multiscale computing software. Philosophical Transactions of the Royal Society A, 377, 2019.

[129] Alexey V Verkhovtsev, Ilia A Solovyov, and Andrey V Solovyov. Advances in multiscale

modeling for novel and emerging technologies. The European Physical Journal D, 75:1–18, 2021.

[130] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling reasons techniques tools. In 22nd International Conference of the System Dynamics Society, 2004.

[131] Mazlina A Majid, Mohammed Fakhreldin, and Kamal Z Zuhairi. Comparing discrete event and agent based simulation in modelling human behaviour at airport check in counter. In International Conference on Human-Computer Interaction Springer Cham, pages 510–522, 2016.

[132] Nora Metzner. A comparison of agent based and discrete event simulation for assessing airport terminal resilience. Transportation Research Procedia, 43:209–218, 2019.

[133] Robert Maidstone. Discrete event simulation system dynamics and agent based simulation discussion and comparison. System, 1(6):1–6, 2012.

[134] Benjamin Dubiel Omer Tsimhoni. Integrating agent based modeling into a discrete event simulation. In Winter Simulation Conference, 2005.

[135] Muhammad Raees, Tamim Ahmed Khan, Khurrum Mustafa Abbasi, Afzal Ahmed, Samina Fazilat, and Inaam Ahmed. Context aware services using manets for long distance vehicular systems a cognitive agent-based model. Scientific Programming, 2021:12 pages, 2021.

[136] Todorka Glushkova, Stanimir Stoyanov, Ivan Popchev, and Stoyan Cheresharov. Ambient oriented modelling in a virtual educational space. Compt. Rend. Acad. bulg. Sci, 71(3):398–406, 2018.

[137] Stanimir Stoyanov, Todorka Glushkova, Asya Stoyanova Doycheva, and Vanya Ivanova. A reference architecture supporting smart city applications. In In International Conference on Business Information Systems, pages 463–474. Springer, Cham, 2019.

[138] Todorka Glushkova, Stanimir Stoyanov, Asya Stoyanovab Doycheva, Vanya Ivanova, and Lyubka Doukovska. Ambient an environment for ambient oriented modeling. International Journal of Computing, 18(3):331–340, 2019.

[139] Serter Iyigunlu, Clinton Fookes, and Prasad Yarlagadda. Agent-based modelling of aircraft boarding methods. In 4th IEEE International Conference On Simulation And Modeling Methodologies Technologies And Applications SIMULTECH, pages 148–154, 2014.

[140] Martin Molina, Sergio Carrasco, and Jorge Martin. Agent based modeling and simulation

for the design of the future european air traffic management system the experience of cassiopeia. In International Conference on Practical Applications of Agents and Multi Agent Systems, pages 22–33. Springer, Cham, 2014.

[141] Christian Bongiorno, Salvatore Micciche, and Rosario N. Mantegna. An empirically grounded agent based model for modeling directs, conflict detection and resolution operations in air traffic management. PLoS one, 12(4):e0175036, 2017.

[142] Camelia Delcea, Liviu Adrian Cotfas, and Ramona Paun. Agent based evaluation of the airplane boarding strategies efficiency and sustainability. Sustainability, 10(6):1879, 2018.

[143] Dominik Grether, S Furbas, and Kai Nagel. Agent-based modelling and simulation of air transport technology. In Procedia Computer Science, volume 19, pages 821–828, 2013.

[144] Colin JR Sheppard, Andrew Harris, and Anand R Gopal. Cost effective siting of electric vehicle charging infrastructure with agent based modeling. IEEE Transactions on Transportation Electrification, 2(2):174–189, 2016.

[145] Wei Yang, Yue Xiang, Junyong Liu, and Chenghong Gu. Agent based modeling for scale evolution of plug in electric vehicles and charging demand. IEEE Transactions on Power Systems, 33(2):1915–1925, 2017.

[146] Joschka Bischoff, Francisco J Marquez Fernandez, Gabriel Domingues Olavarria, Michal Maciejewski, and Kai Nagel. Impacts of vehicle fleet electrification in sweden a simulation based assessment of long distance trips. In 6th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems MTITS, pages 1–7, 2019.

[147] Adedamola Adepetu, Srinivasan Keshav, and Vijay Arya. An agent based electric vehicle ecosystem model san francisco case study. Transport Policy, 46:109–122, 2016.

[148] Junjie Hu, Hugo Morais, Morten Lind, and Henrik W Bindner. Multi agent based modeling for electric vehicle integration in a distribution network operation. Electric Power Systems Research, 136:341–351, 2016.

[149] Marco Pagani, Wolfgang Korosec, Ndaona Chokani, and Reza S Abhari. User behaviour and electric vehicle charging infrastructure an agent based model assessment. Applied Energy, 254:113680, 2019.

[150] Chris Silvia and Rachel M Krause. Assessing the impact of policy interventions on the adoption of plug in electric vehicles an agent based model. Energy Policy, 96:105–118, 2016.

[151] Ingo Wolf, Tobias Schröder, Jochen Neumann, and Gerhard de Haan. Changing minds about electric cars: An empirically grounded agent-based modeling approach. Technological forecasting and social change, 94:269–285, 2015.

[152] Jun Liu, Kara M Kockelman, Patrick M Boesch, and Francesco Ciari. Tracking a system of shared autonomous vehicles across the austin, texas network using agent-based simulation. Transportation, 44(6):1261–1278, 2017.

[153] Poulami Dalapati, Abhijeet Padhy, Bhawana Mishra, Animesh Dutta, and Swapan Bhattacharya. Real time collision handling in railway transport network an agent based modeling and simulation approach. Transportation Letters, 11(8):458–468, 2019.

[154] Qiling Zou, Daniel S Fernandes, and Suren Chen. Agent based evacuation simulation from subway train and platform. Journal of Transportation Safety and Security, pages 1–22, 2019.

[155] Sony Sulaksono Wibowo and Piotr Olszewski. Modeling walking accessibility to public transport terminals case study of singapore mass rapid transit. Journal of the Eastern Asia Society for Transportation Studies, 6:147–156, 2005.

[156] James P Warren and Adriana Ortegon Sanchez. Designing and modeling havanas future bus rapid transit. Urban Design and Planning, 169(2):104–119, 2015.

[157] Jiaqing Wu, Rui Song, Youan Wang, Feng Chen, and Shubin Li. Modeling the coordinated operation between bus rapid transit and bus. Mathematical Problems in Engineering, 2015:7 pages, 2015.

[158] Wanjing Ma and Xiaoguang Yang. A passive transit signal priority approach for bus rapid transit system. In Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference Seattle WA USA, 2007.

[159] Ioakeim G Georgoudas and Georgios Ch Sirakoulis. An anticipative crowd management system preventing clogging in exits during pedestrian evacuation processes. IEEE Systems Journal, 5(1):129–141, 2011.

[160] Ning Wang, Yangzhou Chen, and Liguo Zhang. Design of multi agent based distributed scheduling system for bus rapid transit. In IEEE Third International Conference on Intelligent Human Machine Systems and Cybernetics, 2011.

[161] Fabian Marki, David Charpar, and Kay W Axhausen. Agent based model for continuous activity planning with an open planning horizon. Transportation, February:18, 2014.

[162] Emily Zechman Berglund. Using agent based modeling for water resources planning and

management. Journal of Water Resources Planning and Management, May:040150251–0401502517, 2015.

[163] Inocencio Rodriguez Gonzalez, Gerard E D Souza, and Zarina Ismailova. Agent based modeling an application to natural resource mangement. Journal of Environmental Protection, 9:991–1019, 2018.

[164] Romain Franceschini, Simon Van Mierlo, and Hans Vangheluwe. Towards adaptive abstraction in agent based simulation. In Proceedings of the 2019 Winter Simulation Conference, 2019.

[165] Eric Bonabeau. Agent based modeling methods and techniques for simulating human systems. In Proceedings of the national academy of sciences, volume 99 of 3, pages 7280–7287, 2002.

[166] Robert Siegfried, Axel Lehmann, Rachid El Abdouni Khayari, and Tobias Kiesling. A reference model for agent based modeling and simulation. In Proceedings of the 2009 spring simulation multiconference, pages 1–8, 2009.

[167] Christian Derksen, Cherif Branki, and Rainer Unland. A framework for agent based simulations of hybrid energy infrastructures. In Proceedings of the federated conference on computer science and information systems, pages 1293–1299, 2012.

[168] Mizar Luca Federici, Stefano Redaelli, and Giuseppe Vizzari. Models abstractins and phases in multi agent based simulation. In WOA, 2006.

[169] Sameera Abar, Georios K Theodoropoulos, and Pierre Lemarinier. Agent based modelling and simulation tools a review of the state of art software. Computer Science Review, 24:13–33, 2017.

[170] Todorka Glushkova, Maria Miteva, Asya Stoyanova Doycheva, Vanya Ivanova, and Stanimir Stoyanov. Implementation of a personal internet of thing tourist guide. American Journal of Computation Communication and Control, 5(2):39–51, 2018.

[171] Todorka Glushkova, Stanimir Stoyanov, Ivan Popchev, and Lyubka Doukovska. Ambient oriented modeling in an intelligent agriculture infrastructure. In IEEE 10th International Conference on Intelligent Systems, 2020.

[172] Surekha Paneerselvam. Internet of Things for Industry 4.0 Design, Challenges and Solutions: Role of AI and Bio-Inspired Computing in Decision Making, chapter 8, pages 115–136. Springer Nature Switzerland AG, 2020.

[173] Rajanpreet Kaur Chahal, Neeraj Kumar, and Shalini Batra. Trust management in social internet of things a taxonomy open issues and challenges. Computer Communications, 2019.

[174] R. Santhana Krishnan, E. Golden Julie, Y. Harold Robinson, Raghvendra Kumar S. Raja, Pham Huy Thong, and Le Hoang Son. Fuzzy logic based smart irrigation system using internet of things. Journal of Cleaner Production, 2019.

[175] Ahmadreza Montazerolghaem and Mohammad Hossein Yaghmaee. Load-balanced and qosaware software defined internet of things. IEEE INTERNET OF THINGS JOURNAL, page 15, 2020.

[176] Dariusz Mrozek, Mateusz Milik, Bozena Malysiak Mrozek, Krzysztof Tokarz, Adam Duszenko, and Stanislaw Kozielski. Fuzzy Intelligence in Monitoring Older Adults with Wearables, pages 288–301. Lecture Notes in Computer Science. Springer Nature Switzerland, 2020.

[177] Kawser Wazed Nafi, Tonny Shekha Kar, Md. Amjad Hossain, and M.M.A Hashem. An advanced certain trust model using fuzzy logic and probabilistic logic theory. International Journal of Advanced Computer Science and Applications, 3:164–173, 2012.

[178] Francesco Restuccia, Salvatore DOro, and Tommaso Melodia. Securing the internet of things in the age of machine learning and software defined networking. IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 1, JANUARY 2018, 5(6):4829–4842, 2018.

[179] Tasmima Noushiba Mahbub, S M Salim Reza, Dilshad Ara Hossain, Mehedi Hasan Raju, Md Murshedul Arifeen, and Afida Ayob. Anfis based authentication performance evaluation for enhancing security in internet of things. In ACM ICCA Dhaka, Bangladesh, 2020.

[180] Wendy YAnez, Redowan Mahmud, Rami Bahsoon, Yuqun Zhang, and Rajkumar Buyya. Data allocation mechanism for internet-of-things systems with blockchain. IEEE INTERNET OF THINGS JOURNAL, 7(4):3509–3522, 2020.

[181] Ahmad Khalil, Nader Mbarek, and Olivier Togni. Fuzzy logic based security trust evaluation for iot environments. In IEEE ACS 16th International Conference on Computer Systems and Applications AICCSA, 2019.

[182] Farhad Lotfi and Ali Soleimani. A model to stabilize e loyalty to healthcare services in the context of the internet of things by using fuzzy ahp method. In International Conference on

Computation, Automation and Knowledge Management ICCAKM Amity University, 2020.

[183] Vikram Puri, Magesh Chandramouli, Chung Van Le, and Trinh Hiep Hoa. Internet of things and fuzzy logic based hybrid approach for the prediction of smart farming system. In International Conference on Computer Science, Engineering and Applications ICCSEA, 2020.

[184] Jaideep Kaur and Kamaljit Kaur. A fuzzy approach for an iot based automated employee performance appraisal. CMC-Computers, Materials & Continua, 53(1):23–36, 2017.

[185] A A Bolgov, S A Ermakov, L V Parinova, and V N Kostrova. Internet of things networks predictive risk assessment method and security management, volume 862 of IOP Conf. Series: Materials Science and Engineering, page 5. IOP Publishing, 2020.

[186] Teiseer Alzubaidi and Osman Nuri Ucan. Efficient monitoring and control system for hybrid smart grids using fuzzy logic and iot. Aurum Journal Of Engineering Systems And Architecture, 4(1):93–102, 2020.

[187] Qurat ul Ain, Sohail Iqbal, Safdar Abbas Khan, Asad Waqar Malik, Iftikhar Ahmad, and Nadeem Javaid. Iot operating system based fuzzy inference system for home energy management system in smart buildings. Sensors, 18:30 pages, 2018.

[188] Ahmed Mohamed Elmashtoly and ChoongKoo Chang. Prognostics health management system for power transformer with iec61850 and internet of things. Journal of Electrical Engineering & Technology, 15:673–683, 2020.

[189] Daniel Meana Llorian, Cristian Gonzalez Garcia, B. Cristina Pelayo G Bustelo, Juan Manuel Cueva Lovelle, and Nestor Garcia Fernandez. Iofclime the fuzzy logic and the internet of things to control indoor temperature regarding the outdoor ambient conditions. Future Generation Computer Systems, 76:275–284, 2017.

[190] Solomon Gebreyohannes, Ali Karimoddini, Abdollah Homaifar, and Albert Esterline. Formal Verification of a Fuzzy Rule Based Classifier Using the Prototype Verification System, volume 831 of NAFIPS CCIS, pages 1–12. Springer International Publishing AG, 2018.

[191] Jia Guo, Ing-Ray Chen, and Jeffrey J.P. Tsai. A survey of trust computation models for service management in internet of things systems. Computer Communications, 97:1–14, 2017.

[192] Joel Colloc, Relwende A Yameogo, Peter F Summons, Ying Shen, Mira Park, and Jay E

Aronson. Epice an emotion fuzzy vectorial space for time modeling in medical decision. In ACM IML 17 Liverpool, UK, 2017.

[193] Deepika Agrawal and Sudhakar Pandey. Load balanced fuzzy based unequal clustering for wireless sensor networks assisted internet of things. Engineering Reports, page 21, 2020.

[194] Dong Chen, Guiran Chang, Dawei Sun, Jiajia Li, Jie Jia, and Xingwei Wang. Trm iot a trust management model based on fuzzy reputation for internet of things. Comput. Sci. Inf. Syst., 8:1207–1228, 2011.

[195] Lokesh B Bhajantri and Prashant M Baluragi. Context aware data perception in cognitive internet of things cognitive agent approach. International Journal of Hyperconnectivity and the Internet of Things, 4(2):24, 2020.

[196] Hany F. Atlam, Robert J. Walters, Gary B. Wills, and Joshua Daniel. Fuzzy logic with expert judgment to implement an adaptive risk based access control model for iot. Mobile Networks and Applications, page 13, 2019.

[197] Sayedahmed HAM. Intelligent agent approaches for internet of things challenges a review. International Robotics & Automation Journal, 4(1):51–52, 2018.

[198] Xiaoping Jiang, Hao Ding, Hongling Shi, and Chenghua Li. Novel qos optimization paradigm for iot systems with fuzzy logic and visual information mining integration. Neural Computing and Applications, page 17, 2019.

[199] Diego M Jimenez Bravo, Alvaro Lozano Murciego, Daniel H de la Iglesia, Juan F De Paz, and Gabriel Villarrubia Gonzalez. Central heating cost optimization for smart homes with fuzzy logic and a multi agent architecture. Applied Sciences, 10:1–26, 2020.

[200] Hasan Omar Al Sakran. Intelligent traffic information system based on integration of internet of things and agent technology. International Journal of Advanced Computer Science and Applications, 6(2):37–43, 2015.

[201] Ghada Besbes, Hajer Baazaoui Zghal, and Yann Pollet. Semantic based collaborative decisional system integrating fuzzy reasoning in an iot context. In International Conference on Information Systems Development Toulon France, 2019.

[202] Sofia Kouah and Ilham Kitouni. Internet of things agents diagnosis architecture application to healthcare iot system. In 3rd Edition of the International Conference on Advanced Aspects of Software Engineering ICAASE18 Constantine Algeria, 2018.

[203] Hongzhuo Qi. Fuzzy logic hybridized artificial intelligence for computing and networking

on internet of things platform. Peer to Peer Networking and Applications, page 11 pages, 2020.

[204] Venus Mohammadi, Amir Masoud Rahmani, Aso Mohammed Darwesh, and Amir Sahafi. Trust based recommendation systems in internet of things a systematic literature review. Hum. Cent. Comput. Inf. Sci., 9(21):61 pages, 2019.

[205] Kouah Sofia and Kitouni Ilham. Multi-layer agent-based architecture for internet of things systems. Journal of Information Technology Research, 11(4):32–52, 2018.

[206] Sondes Titi, Hadda Ben Elhadj, and Lamia Chaari Fourati. A fuzzy ontology-based diabetes monitoring system using internet of things. In ICOST LNCS 12157, pages 287–295, 2020.

[207] Nikolaos L Tsakiridis, Themistoklis Diamantopoulos, Andreas L Symeonidis, John B Theocharis, Athanasios Iossifides, Periklis Chatzimisios, George Pratos, and Dimitris Kouvas. Versatile internet of things for agriculture an explainable ai approach. In AIAI IFIP AICT 584, pages 180–191, 2020.

[208] Jiafu Wan, Jiapeng Li, Qingsong Hua, Antonio Celesti, and Zhongren Wang. Intelligent equipment design assisted by cognitive internet of things and industrial big data. Neural Computing and Applications, 32:4463–4472, 2020.

[209] Syed Rameem Zahra and Mohammad Ahsan Chishti. Fuzzy logic and fog based secure architecture for internet of things flfsiot. Journal of Ambient Intelligence and Humanized Computing, page 25 pages, 2020.

[210] J. Carbo, J. M. Molina, and J. Davila. Trust management through fuzzy reputation. International Journal of Cooperative Information Systems, 12(1):135–155, 2003.

[211] R.M. Dijkman, B. Sprenkels, T. Peetersa, and A. Janssen. Business models for the internet of things. International Journal of Information Management, 35:672–678, 2015.

[212] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things vision applications and research challenges. Ad Hoc Networks, 10:1497–1516, 2012.

[213] Xueqin Jia, Jing Wang, and Qing He. Iot business model and extended technical requirements. In In proceedings of ICCTA2011, 2011.

[214] Sheeraz A. Alvi, Bilal Afzal, Ghalib A. Shah, Luigi Atzori, and Waqar Mahmood. Internet of multimedia things vision and challenges. Ad Hoc Networks, 33:87–111, 2015.

[215] Jussi Kiljander, Alfredo D'elia, Francesco Morandi, Pasi Hyttinen, Janne Takalo Mattila,

Arto Ylisaukko Oja, Juha Pekka Soininen, and Tullio Salmon Cinotti. Semantic interoperability architecture for pervasive computing and internet of things. IEEE Access, 2:856–873, 2014.

[216] Luis Guijarro, Vicent Pla, Jose R. Vidal, and Maurizio Naldi. Maximum profit two sided pricing in service platforms based on wireless sensor networks. IEEE Wireless Communications Letters, 5:8–11, 2016.

[217] Amirhossein Ghanbari, Andres Laya, Jesus Alonso-Zarate, and Jan Markendahl. Business development in the internet of things a matter of vertical cooperation. IEEE Communications Magazine, 0:135–141, 2017.

[218] Kai Kang, Zhibo Pang, Li Da Xu, Liya Ma, and Cong Wang. An interactive trust model for application market of the internet of things. IEEE Transactions On Industrial Informatics, 10:1516–1526, 2014.

[219] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. Computer Networks, 57:2266–2279, 2013.

[220] H. Van Dyke Parunak, Robert Savit, and Rick L. Riolo. Agent Based Modeling vs Equation Based Modeling A Case Study and Users Guide. Springer Berlin Heidelberg, 1998.

[221] Charles M. Macal and Michael J. North. Tutorial on agent-based modeling and simulation. In

2005 Winter Simulation Conference, 2005.

[222] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. Ieee Access Journal, 4:2292–2303, 2016.

[223] Christopher K. Frantz and Mariusz Nowostawski. From institutions to code towards automated generation of smart contracts. In IEEE 1st International Workshops on Foundations and Applications of Self Systems, 2016.

[224] H.V. Asha, Shantharam Nayak, and Annamma Abraham. Formalization of soa concepts with mathematical foundation. International Journal of Electrical and Computer Engineering (IJECE), 10(4):3883–3888, 2020.

[225] Khurrum Mustafa Abbasi, Tamim Ahmed Khan, and Irfan ul Haq. Hierarchical modeling of complex internet of things systems using conceptual modeling approaches. IEEE Access, 7:102772–102791, 2019.

[226] Tamim Ahmed Khan and Reiko Heckel. On model-based regression testing of web-services using dependency analysis of visual contracts. In Dimitra Giannakopoulou and Fernando Orejas, editors, Fundamental Approaches to Software Engineering, pages 341–355, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[227] Stanimir Stoyanov, Todorka Glushkova, and Ivan Popchev. Modeling of intelligent context aware systems. Engineering Sciences Bulgarian, 3:5–21, 2017.

[228] Nelson Alfonso Gomez Cruz, Isabella Loaiza Saa, and Francisco Fernando Ortega Hurtado. Agent based simulation in management and ororganization studies a survey. European Journal of Management and Business Economics, 26(3):313–328, 2017.

[229] He Ding, Yan Guoqiang, and Ma Yuan. Study of bus rapid transit system evaluation based on gray extension matterelement model. In Fourth International Conference on Intelligent Computation Technology and Automation, 2011.

[230] X S Dong, Gang Xiong, Dong Fan, F H Zhu, and Li Xie3. Bus rapid transit brt parallel system based on acp approach. In Proceedings of the 10th World Congress on Intelligent Control and Automation China, 2012.

[231] Vernon Joseph Racehorse, Guohui Zhang, Aaron Sussman, Afshin Jian, and Timothy Parker. Bus rapid transit system deployment for high quality and cost effective transit service a comprehensive review and comparative analysis. IET Intelligent Transport Systems, 9(2):175–183, 2015.

[232] George Papageorgiou, Athanasios Maimaris, and Petros Ioannou. Analysis and evaluation of intelligent bus rapid transit systems in cyprus. In IEEE 18th International Conference on Intelligent Transportation Systems, 2015.

[233] Graham Currie. The demand performance of bus rapid transit. Journal of Public Transportation, 8(1):41–55, 2005.

[234] Dirk Helbing and Stefano Balietti. Social Self Organization, chapter How to do agent based simulations in the future from modeling social mechanisms to emergent phenomena and interactive systems design, pages 25–70. Springer, Berlin, 2013.

[235] Gianluca Manzo. The potential and limitations of agent based simulation an introduction.
Revue francaise de sociologie, 55(4):433–462, 2014.

[236] Shi-Wan Lin, Mark Crawford, and Stephen Mellor. The industrial internet of things volume g1: Reference architecture. Technical report, Industrial Internet Consortium, 2017.

[237] Jens Gayko, WEI Sha, Martin Hankel, Bosch Rexroth, Frank Schewe, Phoenix Contact, Michael Hoffmeister, and Festo. Alignment report for reference architectural model for industrie 4.0/ intelligent manufacturing system architecture. Technical report, Sino-German Industrie 4.0/Intelligent Manufacturing Standardisation Sub-Working Group, 2018.

[238] Bruno Costa, Paulo F Pires, and Flavia C Delicato. Modeling soa based iot applications with soaml4iot. In IEEE 5th World Forum on Internet of Things WF-IoT, pages 496–501, 2019.

[239] Bruno Costa, Paulo F Pires, and Flavia C Delicato. Modeling iot applications with sysml4iot. In IEEE 42th Euromicro Conference on Software Engineering and Advanced Applications, pages 157–164, 2016.

[240] Marcelo Pitanga Alves, Flávia C. Delicato, and Paulo F. Pires. Iota-md: a model-driven approach for applying qos attributes in the development of the iot systems. In In Proceedings of the Symposium on Applied Computing, pages 1773–1780, 2017.

[241] Luca Cardelli and Andrew D Gordon. Mobile ambients. Theoretical Computer Science, 240:177–213, 2000.

[242] Joachim Block. Hybrid agent based modeling habm a framework for combining agent based modeling and simulation discrete event simulation and system dynamics. In 2017 Operations Research Proceedings 2017 Springer Cham, pages 603–608, 2018.

[243] Richard M Crowder, Mark A Robinson, Helen P N Hughes, and Yee Wai Sim. The development of an agent based modeling framework for simulating engineering team work. Ieee transactions on systems man and cybernetics part a systems and humans, 42(6):1425–1439, 2012.

[244] Rinaldo A Cavalcante and Matthew J Roordaa. Freight market interactions simulation fremis an agent based modeling framework. Procedia Computer Science, 19:867–873, 2013.

[245] David Masad and Jacqueline Kazil. Mesa an agent based modeling framework. In Proc of the 14th python in science conf, 2015.

[246] Daniel Foguelman, Philipp Henning, Adelinde Uhrmacher, and Rodrigo Castro. Eb devs a formal framework for modeling and simulation of emergent behavior in dynamic complex systems. Journal of Computational Science, 53:1–25, 2021.

[247] Sanjay Jain and Charles R McLean. An integrating framework for modeling and simulation for incident management. Journal of homeland security and emergency management, 3:1–24, 2006.

[248] Michele D Simonia and Christian G Claudel. A simulation framework for modeling urban freight operations impacts on traffic networks. Simulation modelling practice and theory, 86:36–54, 2018.

[249] Mamadou Kaba Traore, Gregory Zacharewicz, Raphael Duboz, and Bernard Zeigler. Modeling and simulation framework for value-based healthcare systems. Simulation Transactions of the Society for Modeling and Simulation International, 95(6):481–497, 2019.

[250] Joshua Auld, Michael Hope, Hubert Ley, Vadim Sokolov, Bo Xua, and Kuilin Zhang. Polaris agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transportation research part c emerging technologies, 64:101–116, 2016.

[251] Mengmeng Ge and Dong Seong Kim. A framework for modeling and assessing security of the internet of things. In IEEE 21st International Conference on Parallel and Distributed Systems, pages 776–781, 2015.