Doctoral Dissertation – Academic Year 2022

A City-Scale Networked Sensing Platform with Spatio-Temporal Information Processing

A dissertation for the degree of Ph.D.
in Media and Governance

Graduate School of Media and Governance
Keio University

Takafumi Kawasaki

Abstract of Doctoral Thesis Academic Year 2022

A City-Scale Networked Sensing Platform with Spatio-Temporal Information Processing

All data generated in cities can be represented in a spatio-temporal manner. This study expresses this data in three dimensions by combining the time and spatial axes, defined as spatio-temporal information. This study assumes that data in a city should be simultaneously delivered to multiple targets and extends the data delivery platform based on the publish/subscribe messaging model (pub/sub-model) with spatio-temporal information processing. All city information is collected from embedded sensors across the city and the intelligent devices people carry with them. Monitoring the in-city status and solving problems are important to improve people's quality of life. Solving city problems using the information collected makes a city smart. This study takes its concept and systematically summarizes sensing technologies to support it. Moreover, as an example of Smart Sensing, this study considers the road condition detection method using public cars. This study aims to extend data delivery platforms for cities. First, it defines time and spatial elements and develops the middleware required for extending Internet of Things (IoT) platforms. The delivery frequency depends on the sender at pub/sub-model because it is delivered by push-type. Hence, 1) data may be delivered beyond the receiver's request, and 2) data is delivered in pieces when receiving data from multiple senders, making synchronous handling difficult. This study developed two extended middleware for IoT platforms: a smart data streamer (SDS) and dynamic topic optimization system (DTOS). The SDS can configure the receiver's receive frequency parameter and deliver data synchronously to the receiver over the receive frequency. Next, users must order topics that they want to receive data about in advance through the pub/sub-model. Nevertheless, the target changes when the receiver moves. DTOS realizes the dynamic target changing using spatial information.

These two systems define delivery control based on time and spatial information, as well as the information elements to be considered by the sender and receiver for this purpose. This study demonstrates how to combine spatial and temporal information to express them. Moreover, this study demonstrates the feasibility of spatio-temporal information processing by showing the various combinations that realize various clusterings.

Keywords
Smart City, Internet-of-Things, Smart Sensing, Middleware, Spatio-Temporal Information

Graduate School of Media and Governance, Keio University
Takafumi Kawasaki

博士論文要旨 2022 年度 (令和 4 年度)

時空間情報処理を用いた都市規模なネットワークドセンシング基盤

街で生じる全てのデータが時空間上に表現できる．本研究ではこの時空間を時間軸と空間軸の組み合わせによって三次元で表現する．これを「時空間情報」と定義する．本研究では，街の情報は受信を希望する複数の対象に同時に配送されるべきとし，パブリッシュ/サブスクライブメッセージングモデル（以下,「パブサブモデル」）に基づくデータ配信プラットフォームを時空間情報処理によって拡張する．街のインフラへのセンサを埋め込みや，人々が日常的にスマートデバイスを持つようになったことで，街のあらゆる情報が取得できるようになりつつある．人々の生活の質を上げるためには，街全体のモニタリングや，街に潜む課題を解決することが重要である．この街の課題を情報の力で解決することを「街のスマート化」と呼ぶ．本研究では街のスマート化をコンセプトとして，それを支えるセンシング技術を体系的にまとめる．スマートセンシングの一例として，筆者が行った公用車を用いた道路の状態検出手法を示す．これらの背景を踏まえ，街を対象としたデータ配信基盤の拡張を目指す．まず，時間情報の要素と空間情報の要素を定義し，それぞれを用いたプラットフォーム拡張ミドルウェアを開発した．パブサブモデルではプッシュ型でデータが配送されるため，その送信頻度は送信者に依存する．そのため，1) 受信者の要望以上のデータが配信される可能性があること，2) 複数の送信者からのデータを受信する際にバラバラにデータが配送されるため，同期的に扱うのが困難となる．本研究では「SDS (Smart Data Streamer)」と「DTOS (Dynamic Topic Optimization System)」の二つの IoT プラットフォーム拡張システムを開発した．SDS は受信者が受信したい周期を SDS 上に保存できるようにし，その周期に応じて受信者が希望したトピック全ての情報を同時に配信するように制御する．次に，パブサブモデルでは予め受信したい情報のトピックを指定する必要があるが，受信者が移動する場合，その対象は動的に変わる．DTOS は空間情報を用いて動的なトピック切り替えを実現する．

この 2 つのシステムによって時間情報と空間情報による配信制御と，そのために送信者と受信者が考慮すべき情報要素を定義する．これらを組み合わせることによって時空間情報が表現でき，その組み合わせ方によって様々なクラスタリングが表現できることを示し，時空間情報処理の実現可能性を示す.

キーワード

スマートシティ，Internet-of-Things，スマートセンシング，ミドルウェア，時空間情報

慶應義塾大学大学院 政策・メディア研究科

河崎 隆文

Acknowledgments

First, I would like to express great appreciation to my supervisor Professor Jin Nakazawa. Without his long-time sincere guidance, this dissertation and my life as a researcher not have started. Second, I also appreciate the assistance of Professor Osamu Nakamura, Associate Professor Tadashi Okoshi, and Associate Professor Takeshi Iwamoto. They taught me their insightful comments and advice, which gave me a fresh perspective. I want to tell Professor Jin Nakazawa for graciously accepting my admission.

After graduating from a master's course, I left a research road because I belonged to a company and worked there. Nevertheless, when I told him I wanted to return to academia, he accepted it. He gave me many advice and comments on my research plan. Then, my research life started again.

However, shortly after being admitted as a Ph.D. student, the large-scale pandemic of COVID-19 began worldwide. The pandemic's impact was vast; my daily life has changed significantly. It forced me to change to remote work because everyone could not go to college by the pandemic. I felt lonely while proceeding with my research work in this situation. I think that many students also felt the same as me. Professor Jin Nakazawa and Associate Professor Tadashi Okoshi have considered and practiced various communication methods via the network. Therefore, I was able to proceed with my research smoothly. Professor Jin Nakazawa gave much advice in considering my research area. Associate Professor Tadashi Okoshi gave many hints to expand my idea from his different insights. Moreover, Professor Osamu Nakamura gave comments and advice to summarize my study. His guidance helped me because he listened and thought about my research from a bird's eye view. Their comments, advice, and hints are constructive to my research. I want to thank Associate Professor Takeshi Iwamoto, a professional in ubiquitous computing at Toyama Prefectural University. I studied for my undergraduate days at Toyama Prefectural University. Associate Professor Takeshi Iwamoto has supported me since then. He gave me some chances to consider my research life. If he did not give me opportunities to communicate with Tokada Laboratory members, I would never be here now. In addition, after graduating the university, he supports me as always.

I also thank Professor Hideyuki Tokuda, Professor Kazunori Takashio, Associate Professor Takuro Yonezawa, and Associate Professor Yin Chen. They gave me a lot of advice and comments for my master's course and various opportunities to grow.

Moreover, I have greatly benefited from Yuka Matsuo, Akira Tsuge, Tomotaka Ito, Sachiko Endo, Dr. Wataru Sasaki, Naohiro Isokawa, Kazuhiro Imura, and all of the laboratory members in Nakazawa and Okoshi Laboratory. In addition, thanks to Shinnosuke Date, he and I sometimes talked about our mutual research status. Moreover, I want to gratitude people who cooperated in collaborative projects. Especially Georgios Palaiokrassas, a doctor of Engineering at National Technical University of Athens, communicate to collaborate on our systems. I'm weak at listening and speaking in English, but he did his best to listen and talk plainly to me. Therefore, our systems realized integration, and the project was a success.

Besides, I want to express my gratitude to my friends. Mainly, I want to thank Masaki Ogawa, Shinsuke Takagi, and Yoshiki Komachi. They often communicated with me after they graduated the university. They listened to my concerns and consulted about my private life. In addition, I want to thank Mitsuru Sakuda, Koki Sugita, Issei Kimura, Takahiro Kitano, Junpei Kawai, Hiroyuki Kuwabara, Yuki Yashima, and Tomoya Fuchigami. They sometimes talked and played with me. They helped and encouraged me; If they were not my friends, I would have had a tough time living a Ph.D. student's life.

Moreover, I would like to thank the relatives and people of my company. I retired from the company early, so my term had been short. Nevertheless, after retiring there, they called out to me, so thank Seitaro Senda, Takuya Kimura, and Yusaku Funane. Mainly, I want to thank Hisashi Morita, who talks and sometimes drinks and gives various advice. In addition, I want to thank Takahiko Sasaki, Ryo Takasan, Arata Deguchi, Katsuhiro Shirosaki, and Masanao Nakada. They also play with me for a long time; their memories sometimes refresh me.

When I look back, many people have supported my student's life. If I study with my power only, I never am here now. My Ph.D. student life was under challenging social situations because of the pandemic. Still, I could feel that I was supported by many people, both in my private and research life. I will never forget this experience.

Last, I would like to best gratitude to my mother. I think she has concerned me since I left my parent's home. In addition, I'm abruptly retiring from the company and entering the university again; I think she was surprised by my choice. Nevertheless, she always looked out for me. Therefore, I could continue my research.

Finally, I would like to thank all the people who always encouraged my research life; and my family for unconditionally supporting me.

August 4, 2022
Takafumi Kawasaki

Contents

List of Figures

List of Tables

Preface

This paper indicates the study of a city-scale networked sensing platform with spatio-temporal information processing. This study aims at the in-city data applications. The author focused on the middleware system for it. Currently, there is an increasing amount of data worldwide by developing sensing technologies. In the background, the platform delivering data to users is essential. Current IoT platforms are implemented based on a push messaging model to deliver much of the data to several people. Nevertheless, the push messaging model's sending timing depends on senders, and it does not consider the physical space data characteristics. Therefore, the author suggests spatio-temporal information processing. In addition, it implemented data delivery middlewares that can understand data characteristics in the physical space.

The author has participated in various projects in the laboratory, and those activities relate to this study. First, it indicates the position of each participating project in this study and its activities in this chapter. Next, it describes each approach to realizing spatio-temporal data delivery for cities. Last, it describes the future direction of this study.

Background

At current, the number of intelligent devices, such as smartphones, is increasing worldwide. These devices have various sensors for device parameters such as accelerator and gyro, health monitoring sensors, such as heart rate, and more. Their real space information is essential to improving their quality of life. It causes everything evolves Smart. Smart means reducing waste while increasing information and opportunities, improving various situations and functions, Etc. Information technologies can satisfy them. For example, smart cities, sensors have been embedded there to collect environmental information, urban information, and human activities in cities. Municipalities can monitor the situation in their city currently and analyze there using stored past data and predict the future. In this way, real space information is helpful for people.

In addition, data is also generated on social networking services (Twitter, Facebook), personal blogs, and various web services. Engineers can get these virtual sensors by using crawling and scraping technologies. Their data has information that the physical sensor cannot detect, including the events of an area. Moreover, it can discover and use embedded information on the Internet. It guides users to use information easily because they do not need to prepare new sensors for their purpose.

However, almost all of the data is managed by each organization. It only opens to itself, so the data is only used in limited situations. Nevertheless, the data has the possibility to can use for several applications and services, so there can expect to generate new values currently from existing data by combining other data.

In this background, it means that the amount of data on the real/virtual space is growing explosively, so there needs to think about how to handle a large amount of data. For example, all of the data delivers freely, and that data stream constructs chaos. It is very hard for users (real space resources) and networks (cyberspace resources). Thus, a suitable data stream needs to be realized that satisfies the requirement for the background. This study focuses on the IoT (Internet-of-Things) platform, aiming to realize efficient and suitable data delivery on IoT platforms.

Research Goal

From the background, the platform will be necessary to deliver data to many users flexibly in the future. It has requirements for various using situations such as smart devices for human activities, environment monitoring in smart cities, mobility monitoring of autonomous cars, Etc. There are several messaging/connecting protocols. This study will choose the optimized protocol to satisfy those requirements.

However, it is essential to define the appropriate protocol and implement the extensible functions for IoT platforms of the next generations. This study focuses on the structure of the physical space data, and it develops the processing functions which can treat the physical space data and cyberspace flexibly for IoT platforms in this study. In addition, it defines spatio-temporal information to process the data delivery to meet conditions of various use cases and suggest processing patterns; finally, it implements an extensible system that has functions by using those elements. This paper first describes the independent elements, spatio-temporal information as the complex element, splitting the time and spatial elements. Therefore, it will describe the spatio-temporal information processing systems as the final deliverable after it describes the definition of each element, its usage, and its extensible system.

Approaches of This Research

The author has participated in some projects in the laboratory, and their activity relates to making a city smart and spatio-temporal data delivery. This section describes each activity and the position of this study. First, it describes the smart sensing technology for cities. It indicates the data sender in cities, and it describes the possibility of increasing data senders and data itself drastically in the future. Next, it describes the spatio-temporal information data delivery middleware to extend IoT platforms. It indicates the broker in a city and describes the necessity of extended IoT platforms for the concept of next-generation cities (Connected-smart-cities). Last, it describes other developed systems in projects for realizing Connected-smart-cities. It indicates developed systems for federation and integration among other IoT platforms. To achieve connected-smart-cities, we need to consider how heterogeneous smart city platforms work together and how to control the scope of openness. It indicates examples to realize it by the developed systems.

Effective Data Sensing Methods for Cities

The author has participated ClouT project from 2014 to 2015. The project goal was to improve citizens' quality of life in a city by providing responsive services by cloud and IoT. The researcher/engineers, including the author, studied and developed various sensing systems in this project. In addition, this project was practiced not only in Japan but also in Europe. Hence, the implemented common platforms and applications do not depend on the specific country. The author studied a sensing method for roads condition using public cars. Inspecting roads condition has a large burden for the municipalities. For instance, there are 1.2 million kilometers of roads in Japan, and 0.9 million kilometers are paved. Pave roads wear away daily because of rain, snow, traffic, and disaster. Maintaining paved roads entrust each municipality. Workers belonging to the city must check the entire city's road condition by cruising and visually themselves. The author studied the system to inspect the road condition automatically by a camera and public cars. In the ClouT project, the author focused on the traffic mark condition because it is especially fragile. It has implemented the system to inspect it by deep learning and image, measured the costs of computer resources to process it, and optimized the model [1, 2, 3].

In addition, the author has participated joint research with Edogawa since 2021. The joint research aims to inspect fatal roads and detect illegal dumping automatically. The author roled the project management and used the knowledge of the previous study to realize the road inspection automatically in Edogawa [4, 5]. These activities indicate new sensing methods for cities, and it suggests the possibility that there will be an increase in various data drastically in cities.

Spatio-Temporal Information Processing Middleware to Extend IoT Platforms

The author has participated in the SSM project. SSM project aimed to develop a Mobility Risk Information Generation Platform. This project studied a method of large-scale data collection in the physical space to sort and circulate them for mobility by collaborating with some universities and companies. The author constructed the platform to collect large-scale data from some resources and circulate them to several users [6, 7]. SOXFire, as an IoT platform developed in the laboratory belonging to the author, can circulate data to several users.

In this project, the author has implemented a data circulation system from other systems using Node-RED, and it can view data flow intuitively. Moreover, in the case of mobility, users move to some locations. IoT platforms need to send data optimally to their locations. Nevertheless, users must continuously connect to an IoT platform when using a push messaging model. The data does not optimize the user's location, generating waste data sending. Besides, users must call the IoT platform when they want to receive data, but the frequency depends on each user, so they have a possibility that they miss the data that they want. Therefore, this study implemented the system to understand spatio-temporal information, and it can control to data sent by them. The defined elements and implemented system indicate the new way of IoT platform understands the physical space's information.

Other Developed Systems for Realizing Connected-Smart-Cities

Last, the author participated in two projects about smart cities, M-Sec, and heterogeneous smart city infrastructure programmable federation demonstration of wide-area data utilization (NEDO-project). M-Sec project aims to research, develop, deploy and demonstrate multi-layered security technologies to ensure hyper-Connected Smart Cities. This project developed some applications and technologies. SOXFire has been used as an IoT Platform on the Japanese side. Smile City Report was developed to sense and share city information by citizens and increase their communication and circulate data using SOXFire. In addition, Smile City Report uses not only Japanese but also Europe users. The Europe side also developed Marketplace as an IoT platform for themselves. They developed the platform developed by blockchain technologies. SOXFire and Marketplace developed different protocols. Then, the author developed SOXCollaborator to integrate with them. SOXCollaborator can store all or specific data from SOXFire. In addition, it defined the necessary parts of sending data format to integrate them [8, 9, 10].

Next, NEDO-project aimed the developed programable federation technologies for smart cities. There are also different IoT platforms, and it considers locations. The location means IoT platforms are installed in each municipality because their data includes citizens' privacy in the city or the city's privacy information. The author developed the system to flexibly control access to other municipality's IoT platforms. Moreover, The author developed Node-RED has implemented the system to realize visual data and the status of the city flow and FIWARE as an IoT platform. The project demonstrated changing the access control Synerex (A developed IoT platform in Nagoya University) to FIWARE [11].

These activities indicate the possibility of different IoT platforms' integration and federation. These technologies are important so that people will realize Connected-smart-city in the future.

Thesis Configuration

It indicates thesis configuration at last of this chapter. This paper is constructed in five chapters. The contents of each chapter are listed below.

Chapter 1: Basic Concepts and Taxonomy of Enabling Technologies for Smart Cities

This chapter describes the basic concepts and taxonomy of enabling technologies for this study. First, it describes smart cities. It indicates the sensing technologies for realizing its Internet-of-Things, Embedded data on Internet, Participatory Sensing, and Smart Sensing for supporting smart cities. About Smart Sensing is discussed, including the author's actual study. Next, It indicates how data is handled based on backgrounds that will change from the past. Last, it discusses the middleware's importance in sending and processing data and the necessary features when using the middleware to support smart cities.

Chapter 2: Spatio-Temporal Information Processing

This chapter describes how data should be spatio-temporal. First, it indicates the three dimensions for expressed spatio-temporal information. Last, it indicates applications with spatio-temporal information processing. Next, it defines the necessary elements to construct the spatio-temporal information.

Chapter 3: Development of Spatio-Temporal Information Processing Middleware to Extend IoT Platforms

This chapter describes the actually developed spatio-temporal information processing systems. First, it describes Smart Data Streamer (SDS). This system is to control data delivery using time elements. Second, it describes the Dynamic Topic Optimization System (DTOS). This system is to control data delivery using spatial elements. The system uses elements defined in chapter 2. Last, the speculative data broadcast system is an example of spatio-temporal information processing. In addition, it suggests the spatio-temporal abstraction by combining time and spatial elements defined in this study.

Chapter 4: Evaluations of Spatio-Temporal Data Delivery

This chapter describes the evaluation of spatio-temporal data delivery. The primary evaluation is constructed in two sections. First, it evaluates the performance of SDS. Second, it evaluates the performance of DTOS. After that, it discusses the performance and the limit of each system.

Chapter 5: What Will Spatio-Temporal Information Processing for City-Scale Sensor Data Delivery Lead in The Future?

This chapter describes the review and future direction of this study. This study indicates the possibility what abstract spatio-temporal information expression. It will guide the development of smart city platforms that understand the physical space's features. Then, smart cities can collaborate in cyberspace. The author calls it connected-smart-cities. Last, it discusses the necessary technologies for realizing connected-smart-cities.

Chapter 1

Basic Concepts and Taxonomy of Enabling Technologies for Smart Cities

This chapter describes the concept of this study. This study develops middlewares to extend IoT platforms and will realize to control several data delivery. First, the chapter explains what cities evolve into smart cities, including examples of smart cities.

After that, it indicates a taxonomy of sensing technologies, Internet-of-Things, Embedded data on the Internet, Participatory Sensing, and Smart Sensing. Moreover, it describes the actual practice of Smart Sensing using public cars. The result indicates the possibility that the number of in-city data will increase, showing the change in how to handle the data in the future. Last, it describes the platform's importance in supporting smart cities, indicating a taxonomy of middleware.

1.1 Basic Concepts of Smart Cities

A city is constructed by various factors, including buildings, citizens, and basic infrastructures such as roads, sewers, and electricity. The quality of the city depends on the quality of them. Currently, cities evolve smart cities using information technologies to increase the city's quality. In this study, smart means the city's various things and problems/tasks to improve and solve by sensors and information technologies. It expresses solving some subjects in the city and improving people's life without physical powers. Therefore, information technologies are essential for it.

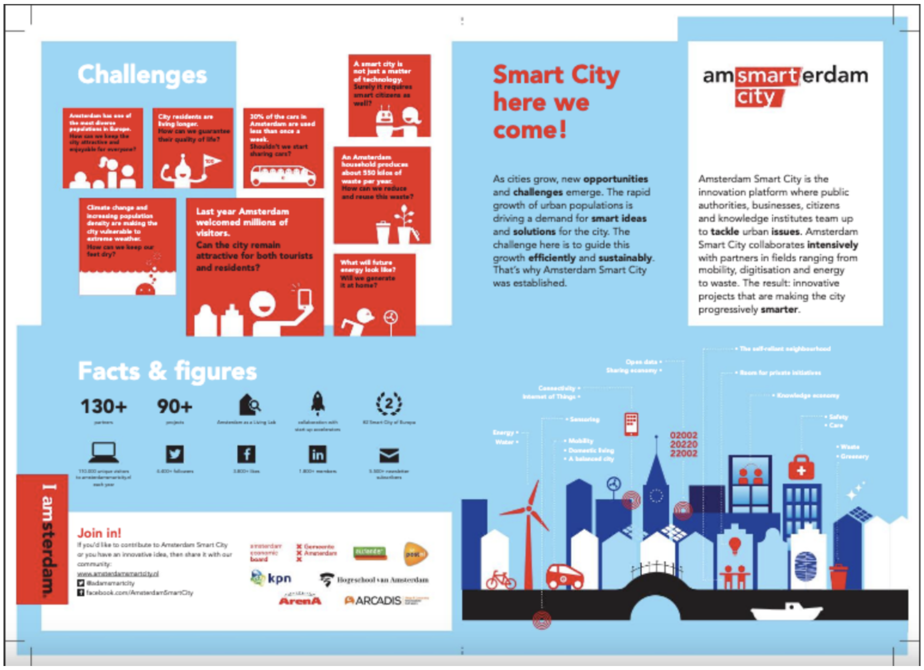


Figure 1.1: Image of Amsterdam Smart City Project[12]

They try to several challenges to solve the problem of happened growth of the urban population and to realize efficiency and sustainability in Amsterdam.

Initially, To make the city smart aimed to achieve green gas reduction targets, as shown in Figure 1.1. Amsterdam Smart City Project[12] is a typical example. They installed Smart Meter to make visible the power consumption to achieve the goal. Smart Meter provides an automatic meter reading function and sends the data to the server. People can estimate and confirm the electricity usage in their own house monthly. Like this, the smart city contributes to the health of the environment.

However, this activity is not only currently. There are many sensors in the city through this activity, including roads, parking, and building. Currently, a smart city's goal is to achieve green gas reduction targets. Countries and their municipalities aim to solve various problems of the environment and their city by using information technologies. First, to scatter sensors in the city is to maintain the city resources automatically. For example, in Smart Santander[13, 14], there are installing as many sensors as over 20,000 in Santander in Spain. The municipality manages the status of various city resources, including the parking system, garbage boxes, and street lights. These activities are others too. Another example, in Japan, TOYOTA MOTOR CORPORATION starts TOYOTA Woven city[15] as a new city concept. This city is a large demonstration experiment of the future smart city. There are embedded sensors in the city and Mobility as a Service, autonomous cars, and Smart Home. This activity will indicate the figure of a future city. The following description shows examples of smart cities.

CityVerve Project: CityVerve is a project in Manchester, England (Figure 1.2). With IoT technologies, the project aims to improve city services across health, transport, social care, and more [16, 17]. For example, combining sensors embedded in light fixtures, the wireless outdoor infrastructure, and other devices in the city created a sensing platform to support energy reduction and monitor the quality of the environment at some locations. This project also embedded sensors into several objects on roads and buildings, and they aim to visualize their city.

Smart Santander: Smart Santander is a project in Santander, Spain (Figure 1.3). This project aims to realize the sensorization of the city [13][19][14]. Santander has embedded several various sensors into the city. The sensors are always connected by wireless mesh, mobile phone networks, and broadband networks. They can find all kinds of information on the streets, including energy power, water quality, air quality, and the status of infrastructures. Therefore, Santander can monitor the city's status in real-time, and the city can predict the value of its sensors.

TOYOTA Woven City: TOYOTA Woven City is a project in Shizuoka, Japan. It is an advanced project especially. This city is a testbed to try some future technologies about human life. For example, the city has three different roads. The first one is used to drive autonomous cars and high-speed vehicles. The second one can drive personal mobility such as Segway and walking. The third one is for walking people only. Besides, they promote the demonstration of realizing carbon neutrality by collaborating with other companies. There will be installed hydrogen stations, and they will try to use the hydrogen gas for cars and more. Moreover, this project challenges various fields, including healthcare, smart home, agriculture, food, education, Etc. This project will indicate the figure of a future smart city.

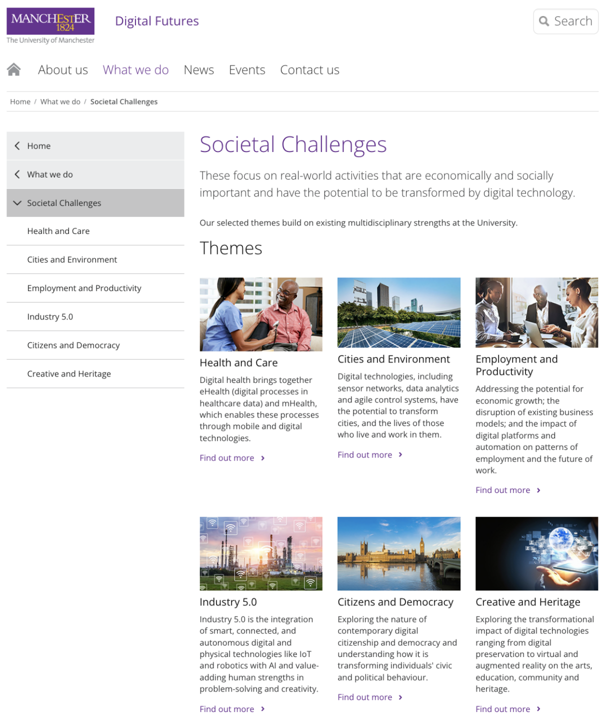


Figure 1.2: Website of CityVerve Project[18]

The project aims the improve the quality of the city in Manchester by using IoT. This page indicates challenge contents.

1.2 A Taxonomy of Sensing Technologies to Support Smart Cities

Making smart cities needs some technologies. Currently, data have been generated in some situations. It is essential to know when and how data are generated to consider a platform. This section indicates sensing technologies and their features to support smart cities.

1.2.1 Internet-of-Things

When it made the Internet, it used to connect to computers. People can get information from other places via servers, and they can send data to other people by networking services, including e-mail and other tools. Currently, the devices have generated various

Figure 1.3: Image of Smart Santander[13]

This image shows the actual sensor information in Santander. Pins indicate the information in Santander. Each pin has information on street lights, parking, environment, Etc.

data, such as video, music, and photos.

From the result, most of the available devices of current can connect Internet. Smartphones and tablet devices are famous examples. People can use these devices anytime and anywhere. These devices provide some services to help their life. Moreover, connecting to the Internet is not only the mobile phones but almost machines and items. For example, in our living life, a smart refrigerator can detect materials in it and suggest the recipe that uses them to people. This function is realized by installing a camera and a microcomputer in the refrigerator. Besides this, almost objects are installed with various sensors and a microcomputer. As a result, almost all objects can connect to the Internet, and it calls Internet-of-Things (IoT). IoT discovers and provides new values to what some objects have, and it contributes to developing a variety of human life. Engineers and researchers must require to create evolutional technologies and discover new value and information from IoT devices in the future world.

1.2.2 Embedded Data on Internet

There are several embedded data on the Internet. The data can separate mainly into two kinds. The first case is that senders send data to social networking services. In this case,

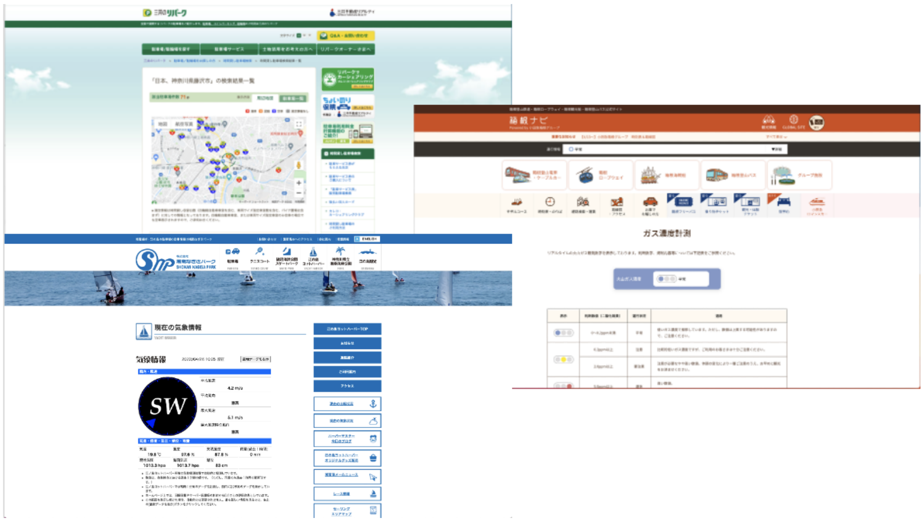


Figure 1.4: Examples of embedded data on Internet[20, 21, 22]

Many companies and organizations open their collected data. The data has some value, but it only opens their websites. Therefore, the usage is limited.

senders send information that is difficult to collect by a physical sensor. For example, a person tweeted, ‘I met a famous entertainer on the street in Tokyo!’ when he found the entertainer on the street. Besides, the post data by people include emotion. This data have some value and do not collect physical sensors. Like this data on Internet, it is called embedded data.

The second case is that senders open data on their websites, but physical sensors collect the data. Figure 1.4 shows embedded data examples.

It is many when a sender is a private company. For example, their parking status is currently collected by physical sensors, and users can confirm it on websites or collaborated applications. Nevertheless, many private companies do not open raw data, and the case is not only parking status. These data lost opportunities to use other situations, also called embedded data. In the case of the first, developers can use data when the API of the service is opened. Besides, developers do not use data easily when the API is not opened or in the second case.

However, some technologies collect data from websites, including crawling and scraping. Nakazawa et al. [23] developed the collecting system from websites. It can understand the website’s structure and scrape target information from there. In addi-

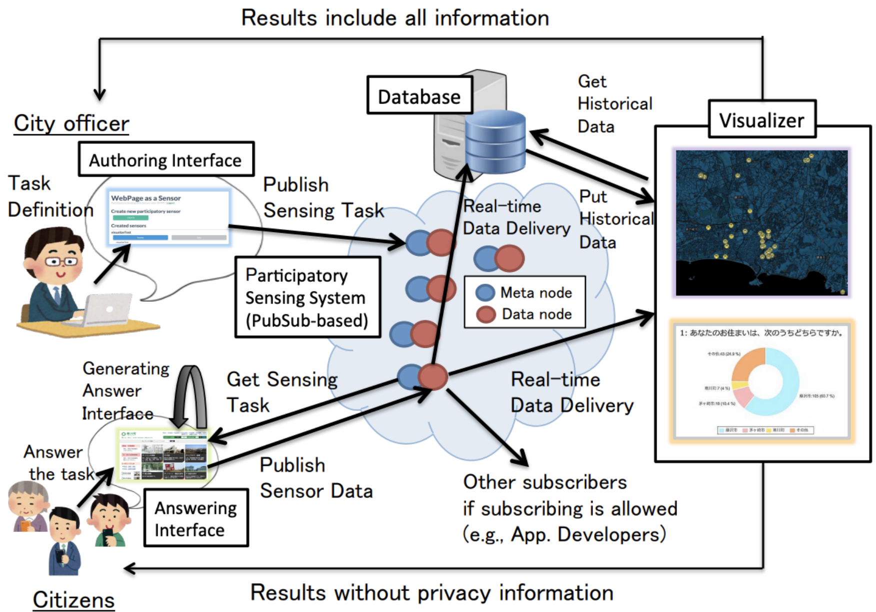


Figure 1.5: Overview of MinaQn[24]

The system can suggest tasks for citizens, and citizens answer the task at any time, and the municipality workers can confirm all answers.

tion, the system transfers the data to other applications via an IoT platform. Therefore, embedded data on Internet are also vital data resources.

1.2.3 Smart Sensing

It means the sensing method uses smart sensors. The sensor is embedded in microprocessors and wireless communication at least[25]. Currently, smart sensors are generally spreading IoT worldwide. There are studies on smart sensing, and many relate to the quality of cities and citizens' lives in the long term. [26, 27, 28, 29] The studies can realize monitoring some information about cities, human activities, and environmental status.

Moreover, there constructs a large and intelligent sensor network by improving smart sensors' specs. Kocakulak et al. indicated an overview of wireless sensor networks towards IoT[30]. They described the definition of wireless sensor networks and their design of it. The paper indicated that the wireless sensor network has many advantages in IoT. For example, the developer does not need complex infrastructure, and the network spends low energy. In addition, wireless sensor networks are compatible with other devices, increasing their usage. It indicates the possibility that IoT's wireless sensor networks are large-effectiveness for monitoring several situations.

1.2.4 Participatory Sensing

This sensing method is similar to the embedded data from social networking services described in the previous section. Participatory sensing handles people as sensors [31]. The world population is increasing, and people usually have smart devices, smartphones, tablets, and wearable devices. Primarily smartphones are used to contact colleagues when working, friends when playing together, or family. The device generally has a camera and chat capabilities and call functionality. In addition, developers can create applications for the device. Participatory sensing uses a person with a smartphone as a sensor from the background. Actually, there are some studies on participatory sensing currently.

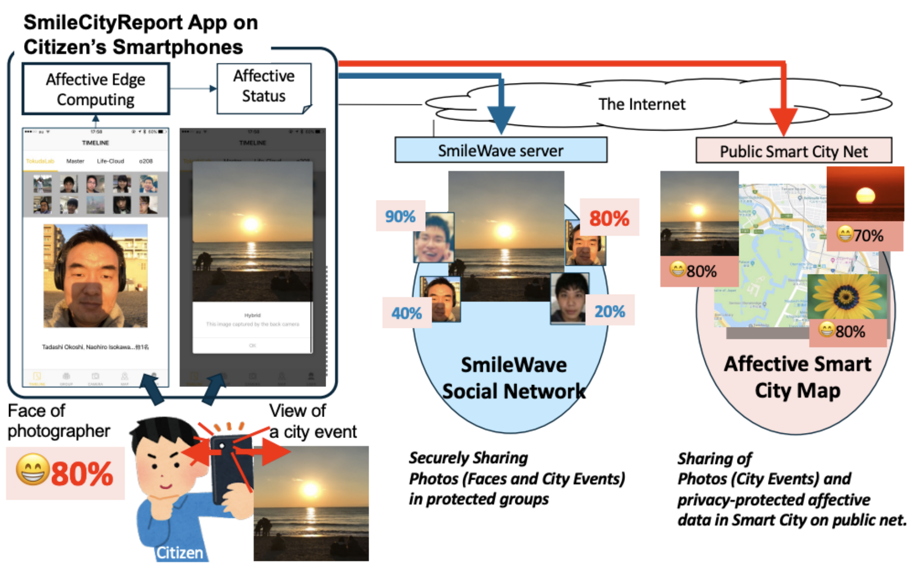


Figure 1.6: Overview of SmileCityReport[32]

Users can collect and share captured views of a city and its affective status.

Sakamura et al. have developed some systems of participatory sensing[33, 34, 24, 35]. For example, they developed a web framework to collect the city's information from both municipality workers and citizens, as shown in Figure 1.5. Municipality workers can create participatory tasks about the city and select the task to open for citizens. Citizens confirm the opened tasks at any time on the website. Computers and mobile devices can access the website, and they answer the task at any time. When a citizen answers the task, the answer content is published to many clients. In addition, the system stores all answers in a database, and municipality workers can confirm them. This study indicated that participatory sensing could catch up with citizens' opinions that do not collect citizen surveys only.

Next, Sasaki et al. have developed Smile City Report [32] as a smartphone app-based participatory sensing. This study aims to collect and share information on the events with users, as shown in Figure 1.6. Then, the reporter simultaneously captures both the city events and emotion-related status using two cameras. The authors confirmed the effectiveness of reporters' more activity that the system provides the reporter's emotion-related status by the double-sided photo shooting feature.

1.3 Practice of In-city Smart Sensing: Realizing Effective Smart Sensing Using Public Cars

This section describes the actual practice of the author. This study developed a method of detecting and collecting traffic markings' quality using garbage trucks as an example of smart sensing.

It has written about the smart sensing of Smart City in the previous section. However, many of their methods made embedded sensors in their city or depended on people's activity. Municipalities need to manage several objects in their city. They want to collect data from wide coverage and high frequency, such as the road and its equipment. In this case, the sensor embedding method must generate high management costs. Besides, in the case of the participatory sensing method, the collecting area and frequency depend on the living people in the city.

In the case of road inspection, the inspection cost is a large burden for municipalities because it has a vast management area and requires high inspection frequency. In addition, road conditions are usually inspected visually, by city staff, only once per several years. To solve this problem have been several studies have proposed various

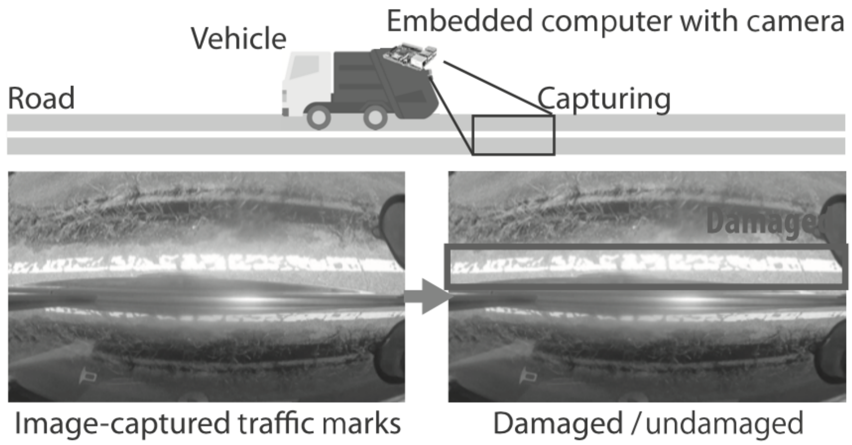


Figure 1.7: Overview of the system to detect road marking damage using a public car

The system proposes the method to detect road marking damage automatically using a public car with an embedded computer with a camera.

approaches to automated road inspection [36, 37, 38, 39, 40, 41, 42]. These studies use accelerator sensors attached to vehicles to detect road roughness automatically to reduce road inspection costs. However, it is difficult to detect road marking damage because they are painted thinly.

Then, This research focuses on the public car because it drives regularly and wide area in the city. This section describes suggestions for the new method to collect and detect road marking using public cars. Figure 1.7 shows the system overview.

Problems of Related Works

There have been several studies aimed at facilitating road inspections. Takahashi et al. focused on the bicycle[40]. The number of sharing bicycles is increasing currently, so they aimed to do monitoring road conditions using a smartphone and a bicycle. They defined the four types of cracks on the road. After that, they realized to inspect road roughness by collaborating on their definition and got the acceleration data from the smartphone.

Besides, Zhao et al. suggested a Dynamic Response Intelligent Monitoring System.[42] The system also uses a smartphone to inspect automatically. They analyze the frequency domain to detect unevenness. The affinity of those studies evaluates flatness.

However, road markings are painted very thin. If it has been damaged, the road does not affect it. Therefore, this research focuses on the road markings that it is difficult to detect for acceleration sensor, and it resolves using the photo taken by a drive recorder.

Automatic Traffic Mark Inspection with Public Cars

This section describes the requirements of automotive sensing for traffic mark inspection. First, we introduce our automotive sensing infrastructure in Fujisawa, Japan. Then, we determine the requirements to adapt the existing automotive sensing infrastructure to traffic mark inspection.

Public Cars as A City Sensing Infrastructure

As part of a smart city project, in collaboration with the city office of Fujisawa City in Japan, we designed and implemented an automotive sensing platform called Cruisers [43, 44] (as shown in Figure 1.8). Then, we deployed the platform using the city ’ s garbage collection trucks to evaluate the applicability of automotive sensing in a realistic urban setting. In Japan, garbage collection is conducted door-to-door by trucks, typically on weekdays; therefore, Cruisers were expected to ensure that most of the city ’ s areas, excluding those uninhabited, were visited at least once each day when garbage is collected collection occurred. Each garbage truck had a sensing module, a micro-computer, and an acellular communication module. The sensing module contained several basic sensors such as a global positioning system (GPS), temperature, accelerometer, gyroscope, and environmental sensors such as particulate matter (PM) 2.5 and nitrogen oxides (NOX).

Requirements

Camera-based road marking inspection has to fulfill the following requirements in order to apply to a sensing system using public cars, which has limitations related to network speed and computational resources.

1. Analysis at edge-side to reduce network communication: Usually, automotive sensing only uses low network capacity, such as a 3G link. It is better to analyze captured images at the edge-side and only send the analysis result through the network because the size of a captured image by a camera is much bigger

Figure 1.8: Images of Cruisers[43]

Cruisers developed in an automotive platform using public cars. The authors use garbage trucks as public cars to collect environmental information in Fujisawa. This system realized wide coverage using several garbage trucks.

than the data captured by simple sensors such as accelerometers and GPS. This requirement leads to the additional requirements listed below.

2. Working with limited computational resources: In addition to network limitations, the computational resources of public cars are also limited. A computer attached to a vehicle is powered by the vechicle's battery. It, in turn, requires low computer specifications. Also, it is better to use an inexpensive computer to keep the cost low. Therefore, we must not anticipate that a high-specification computer will be used in the analysis. Images are usually analyzed using machine-learning techniques. A method using a lightweight learning model and analysis technique is needed to reduce the computer ' s memory and CPU requirements.

3. Balancing high speed and high accuracy of analysis: computational resources are also limited; however, we must analyze traffic marks at high speed and accuracy. The camera would continuously capture images of traffic marks at rates such as 15 frames per second (fps). It is necessary to analyze images within 66 ms to achieve real-time sensing in the case of 15 fps. With real-time processing, road markings have to be analyzed with high accuracy. A camera captures images of road markings mounted to a car, whose speed would be from 0 km/h to 60 km/h

(in the case of garbage trucks in Fujisawa). The analysis should have been able to cope with various images captured under variable car speeds.

This study explored an appropriate and practical analysis method for road marking inspection. The method was required to have the ability to cope with limited computational and network resources.

Approach

This section describes a method to detect road markings damage. First, it introduces the collection process of road markings. Next, it refers to the damage detection model by using CNN.

Data to be Analyzed

This study assumes that the vehicle-mounted camera collects images depicting road marking conditions. In the experiment, a generic car was used, with the camera attached to the side, to capture images of the road surface only, as shown in Figure 1.9. Images were captured when the travel speeds were within the legal speed limit, between 0 km/h and 60 km/h. The captured images are blurred as the traveling speed increases, as shown in Figure 1.10

Detection Model: C's Model

First, we defined damaged traffic marks as the traffic mark image where a damaged part is depicted; thereby, the traffic mark condition was classified as either damaged or undamaged (as shown in Figure 1.11). We regard this problem as a binary classification task. In our study, a CNN model was used to solve this problem. The CNN model was reported to perform well in image correlation tasks. For example, engineers (or researchers) hold an annual classification accuracy competition where approximately one million images are classified into 1000 classes during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [45]–[46]. In this competition, the CNN model became the mainstream classification method after 2012. We regarded CNN models as adequate for solving correlation tasks related to the images considered in this study. Therefore, our work proposed the C ’ s model to extend the CNN model.

CNN is a particular model designed to provide good solutions to image-related problems. CNN is constructed with multiple convolutional layers and a few fully connected layers in almost any case relevant to a classification task. At the convolutional

Figure 1.9: Actual captured image of a road marking

This image shows the actual captured image by a camera. The camera has been attached to the side of a car. The image indicates that its quality satisfies to detect the road marking damage.

Figure 1.10: Road markings captured by different travel speeds

It indicates that the change of captured road markings depends on travel speeds. The figure shows the sample of actual images taken at various travel speeds.

layer, the input image of width n_i, height n_j, c color channels $\mathbf{h} \in R^{n_i \times n_j \times c}$, and output channel d is first convolved with a set of local filters $\mathbf{f} \in \mathbb{R}^{n_i \times n_j \times c \times d}$. For each location/pixel (i, j) of $\mathbf{x}$, the convolutional calculation is defined as:

$$\mathbf{z}_{i,j} = \sum_{i'}^{n'_i} \sum_{j'}^{n'_j} \Phi(\mathbf{f}^T_{i',j'} \mathbf{x}_{i+i',j+j'}) \tag{1.1}$$

where $\mathbf{f}_{i',j'} \in \mathbb{R}^{c \times d}$, $x_{i+i',j+j'} \in \mathbb{R}^c$, $\mathbf{h}_{i,j} \in \mathbb{R}^d$. Φ a nonlinear activation function with a parameterized set of parameters. Φ a nonlinear activation function with a parameterized set of parameters. At this time, it used the rectified linear unit (ReLU) [47]. When the final output was evaluated, the sigmoid function was used. The convolution in Eq. (1) was followed by local max pooling. The pooling layer had two contributions: First, the convolutional layer reduced the output channel dimensionality. In addition, neighboring feature activations were summarized by activating this spatial max pooling and leading to local translation invariance. The final feature map of the last convolutional layer was flattened to a vector representation h. This vector $\tilde{h}$ was fed through a few fully connected nonlinear layers until the output was produced.

Finally, the optimization of these functions was assigned to the optimizer. At this point, we used Adam[48]. The initial values were set at a learning rate of 0.001, and the momentum was set at 0.9. The C ' s model was developed by combining the above. In this study, models have some requirements, for example, regarding size;

Experiment Result and Discussion

This study prepared a dataset with images captured by a car with a camera at several travel speeds. Figure 1.11 shows a part of the dataset, and Table 1.1 shows the environment condition to train and test. The size of its is 224×224 pixels, and the number of images is 25,426 to train, and the number of images is 6,781 to test.

The detection accuracy was evaluated by comparing some different models. This study used Linear SVM[49, 50], Random Forest[51, 52, 53, 54], and C's model to evaluate. First, the result of SVM and Random Forest is shown in Table 1.2. The result indicated that all scores of Random Forest are better than Linear SVM. In addition, it confirmed the accuracy when the training and test datasets separated each travel speed. The result indicates that the accuracy score is highest when the travel speed of the training dataset and test dataset are the same. (all accuracy is 90 % over.)

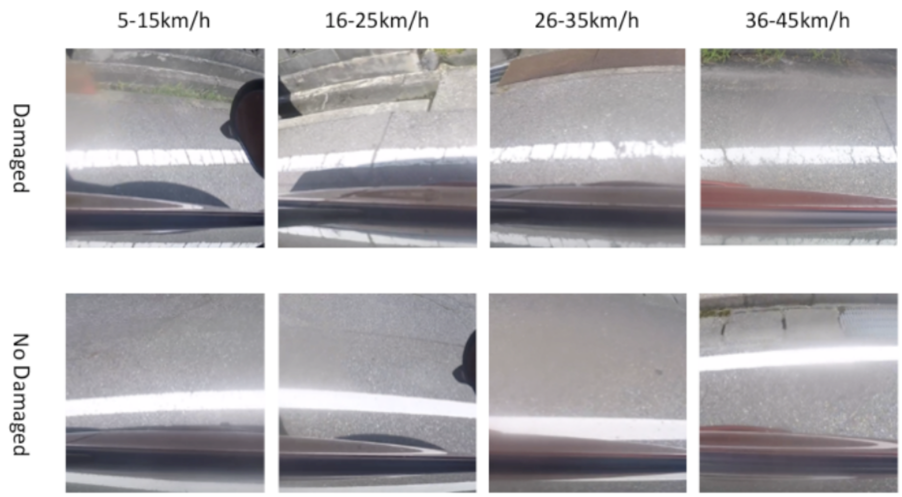


Figure 1.11: Dataset of C's

This shows a part of dataset to train and test. It includes images captured at several travel speeds to comfirm the effect by increasing travel spped.

All estimated road markings damage scores by Random Forest were better than Linear SVM.

Next, it describes the structure of C's model. Figure 1.12 shows it.

The dataset uses the same Linear SVM and Random Forest. This study evaluated two patterns, C's model and combining C's model and Dropout[55] because the result indicates the overfitting when doing a first training and test. The result is shown in Table 1.3. It indicated a high score better than Linear SVM and Random Forest.

Table 1.1: Environment conditions of C's

	Name	Spec
CPU	Intel Core i7	3GHz
Memory	-	64GB 3200MHz
GPU	TITANX	12GB
Language	python	2.7.12
	Keras (Backend: tensorflow)	tensorflow 0.12.0

Table 1.2: Detection Accuracy using Linear SVM and Random Forest
All estimated road markings damage scores by Random Forest were better than Linear SVM.

Algorithm	Accuracy(%)	AUC	Recall	Precision	f1
Linear SVM	80.0	0.79	0.85	0.80	0.83
Random Forest	84.0	0.83	0.91	0.83	0.87

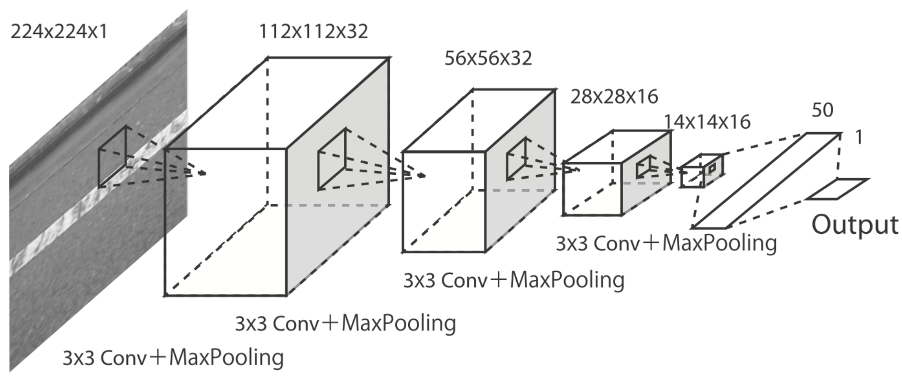


Figure 1.12: Image of C's model structure

C's model is constructed based on CNN. The structure has four convolutional layers and two fully connected layers.

Moreover, the score improved when it added Dropout to the model. This evaluation indicated the effectiveness of C's Model in detecting road marking damage because all scores are the highest in this evaluation.

Last, the experiment confirmed that C's model performs on microcomputers. The reason is that the size of the CNN model is more extensive than other algorithms. When the system uses a microcomputer, the machine resource is limited. Therefore, it is important to confirm the performance. It evaluated the detection accuracy and the processing speed by changing the structure of C's model, and it used a raspberry pi3 as a microcomputer. The result is shown in Figure 1.13. This image indicates various model structures of C's. Two layers mean that two layers construe the model. (16,8) indicates the number of channels in each layer. Therefore, two layers (16,8) means that two layers construct the model, and each layer has 16 channels and eight channels.

It indicated that the processing speed is within 0.5 seconds when using the most complex structure. The accuracy and processing speed depend on the number of layers

Table 1.3: Detection accuracy using C's Model
The accuracy is highest in this evaluation, and all scores improved when added Dropout.

Algorithm	Accuracy(%)	AUC	Recall	Precision	f1
C's Model(CNN)	90.8	0.90	0.93	0.91	0.92
C's Model + Dropout	93.0	0.93	0.94	0.95	0.94

and channels. The processing speed can improve by reducing them, but it causes a down in the accuracy. The model indicated the best performance with two layers (16,8). The processing speed is within 0.1 seconds, and the accuracy is over 92 %.

Therefore, it is essential to suit parameters that use situations. Nevertheless, C's model can perform on microcomputers because the accuracy and processing are very high.

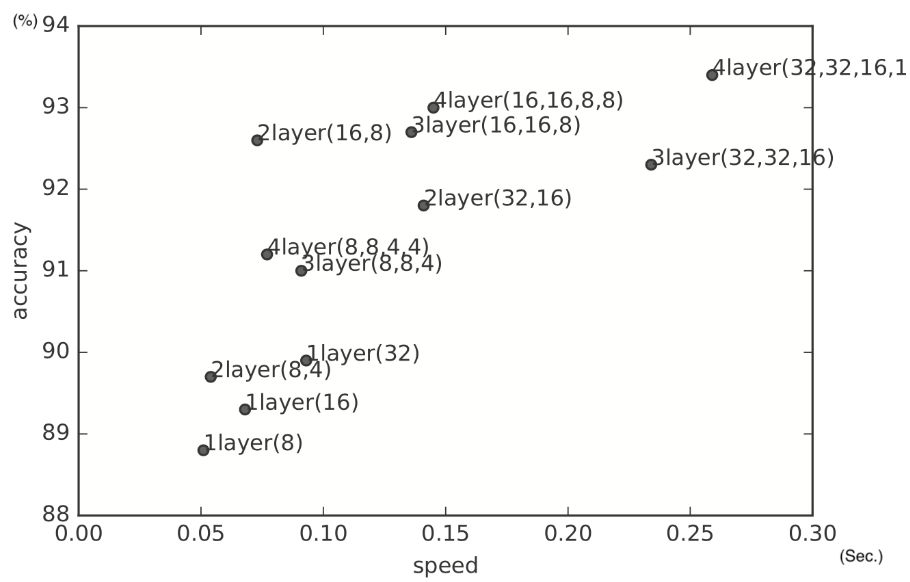


Figure 1.13: Result of changing C's architecture

This result indicates that accuracy and processing speed changes the number of layers. In addition, not changing the number of layers and reducing channels indicated the best performance on raspberry pi.

1.4 Individual Data to Clustered Data

The previous section described various sensing technologies and the actual practice of smart sensing. Those indicate that the amount of data in a city will increase. Then, engineers and researchers need to consider handling those several data effectively. Processing data generate or extract average information. There are some kinds of processing. The most straightforward method is using the raw value as information. Nevertheless, the data can generate more various information through complex processing. It can generate or extract average, predicted value, context-aware data, and more. To realize complex processing needs how to cluster from independent data, and clustering uses features that data has. For example, two different sensors can combine when they have a common target. When a user wants to know the temperature correctly, the system can suggest the value by calculating some sensor values. Heterogeneous data linkage can generate new values never seen before.

1.5 Data That Across Spatio-Temporal

Data in real space has special characteristics. Figure 1.14 shows the data in real space. First, each layer describes spatial information. All sensors and devices are on the layer and have a point. This point means the object's location. For example, (a) a car that people drive often changes its location. (b) people are interested in the various objects, the weather of their location, their hobby shop, and more, but the critical point is that their information spots are close to the user because, if the spot is away from the user, he cannot go there. These user's interest point calls Points of interest (POI). Location is very critical information to indicate users' POI.

Secondly, it explains each layer's overlapping in Figure 1.14. The overlapping describes time information. All data will always occur at some point. This point is time, and all data has this information. In addition, (c) each data from a sensor/device is continuous. For example, environmental information on a location is not independent, as the weather cannot change from sunny to rainy. Therefore, we can analyze the current status and the value trend and estimate and predict the future value. (c) shows the example of a traffic jam in a location. When we go through the same point at the same time every day, we can predict the traffic status because we store its information. Therefore, the time information is critical to cluster and estimate the data.

From the above, this research defines spatio-temporal information as a combination

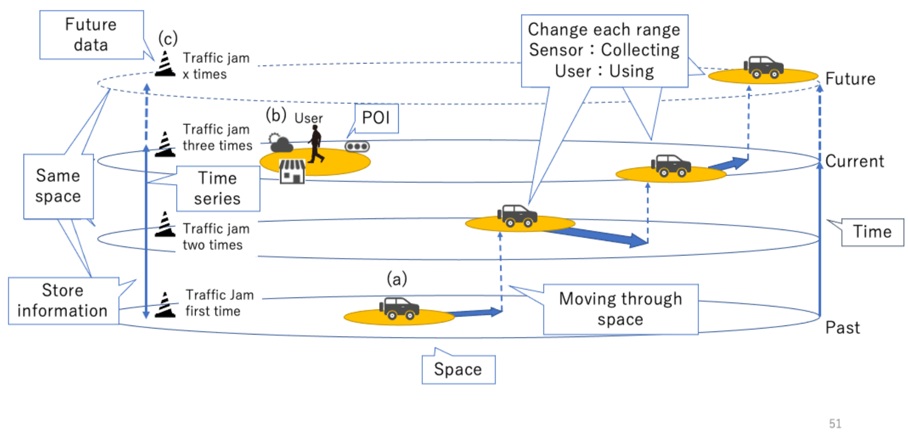


Figure 1.14: Image of data that across spatio-temporal

All data can express in real space. When data are generated continuously, systems can analyze information using it at one position. Besides, the case of it focuses on spatial, and it can cluster data using spatial features.

of time and space.

1.6 A Taxonomy of Necessary Middleware to Support Smart Cities

Data can be easily stored and cached on the computer by developing database systems and improving the computer's specs. Applications use raw and processed data. The methods of data processing are currently various kinds. For example, the temperature sensor senses the temperature value continuously. It detects value every minute or every second. These values are not independent, and we can make this helpful information by analyzing them, for instance, average, max, minimum, and more. Like this example, applications and systems use the information that is various figures.

1.6.1 Data Delivery Platforms

The number of devices connecting Internet will increase in the future. When it increases in the personal, home, and city, the IoT platform is important.

First, the fixed networking service provides high speed and a large scale by developing an optical communications line. In addition, the mobile networking services also will develop by 5th generation mobile networks. Therefore, people can connect many devices to the Internet without the limited number of devices they have.

On the other hand, the amount of data and information in cyberspace. When it increases, people must choose it that they want to use. The existed general server provides the connecting service between a computer and a server. It means the server communicates to computers independently. Hence, when some computers want to same data from the server, each computer must send a request to it, and the server must send data to each computer. When using IoT devices, this communication method has a large waste cost because all devices must send the same request. If the server can search the requested data of each device they want, they can send data to all target devices without their requests. Therefore, different platforms from the existed general server are required in the IoT world. This study aims at expanding to the IoT platforms for Smart City.

1.6.2 Requirements of IoT Platforms

When using IoT platforms for a city, platform placement is essential. Currently, some services use spatio-temporal information. For example, Google map provides the map, navigation, route suggestion, traffic jam information, Etc. Cloud computing supports those services. Cloud is all systems and services provided via the network. It changed infrastructures' form. The traditional system is constructed per private company. Nevertheless, cloud computing realized the abstraction of the infrastructure. In cloud computing, the data is aggregated in the cloud environment. Hence, companies that have large data centers can collect many data. They develop the system by analyzing big data. Moreover, it provides scalability for platforms. On the other hand, when the use case is smart cities, more things must be required. When the data is about a city, it is used for the city's citizens and systems. Hence, the platform must handle data there.

In addition, in the case of smart cities, many events, data, and opportunities are generated for citizens. Especially applications to improve citizens' life often handle the same data among some citizens. For example, the timeline information of buses there does not depend on per citizen. In that case, the platform can handle data publishing to them integrative. Moreover, it can generate new information value by collaborating several data.

Besides, there are an unspecified number of clients, each independent in smart cities. Therefore, data providers and clients must not depend on each other. Even if a data provider cannot provide data, clients must do without it. It can solve this problem using IoT platforms. This study focuses on how to use IoT platforms for smart cities.

The requirements of IoT platforms for smart cities are various, depending on the situation used. This study assumes that IoT platforms are used for several complex information for many clients. When the role of IoT platforms becomes large, the system load becomes large too. It indicates that the IoT platform loses quickness because of the processing time per process. On the other hand, IoT platforms acquire varieties of how to use in the situation without real-time processing. Section 2 indicates the use case.

Figure 1.15 shows the relationship between latency and independence. The horizontal axis of Figure 1.15 means the latency requirement, and the vertical axis of Figure 1.15 means the independence requirement. The most left box requires the hardest latency constraint but does not require independence. For example, a vehicle that needs traffic signals' status is very close. The most quickly data sending is when the target traffic signal and the vehicle connect directly, and the vehicle will stop following the data. The car's performance depends on the traffic signal. Moreover, this study does not consider hard real-time control because the system does not control according to time limitations. It means the system developed in this study does not guarantee safety when the system does not finish processes within the latency requirement. Therefore, the systems are not good at hardware operation.

On the other hand, the most right box requires the easiest latency constraint but requires high independence. For example, users want to book various services, taxis, hotels, events, restaurants, etc. In that case, users want information from some services. In addition, the network latency does not affect users because this case does not require immediacy better than hardware operation. However, if the client application depends on target services, clients are damaged when the target service is down. Therefore, it must require high independence to prevent this problem. This study focuses on the right area of Figure 1.15 because the systems do not have requirements regarding latency. This study implements the method of speculative data publishing to control soft real-time processing. It describes section 3. Hence, platforms for cities are required to comply with the following requirements.

- Data must be properly and at the same time delivered to an unspecified number

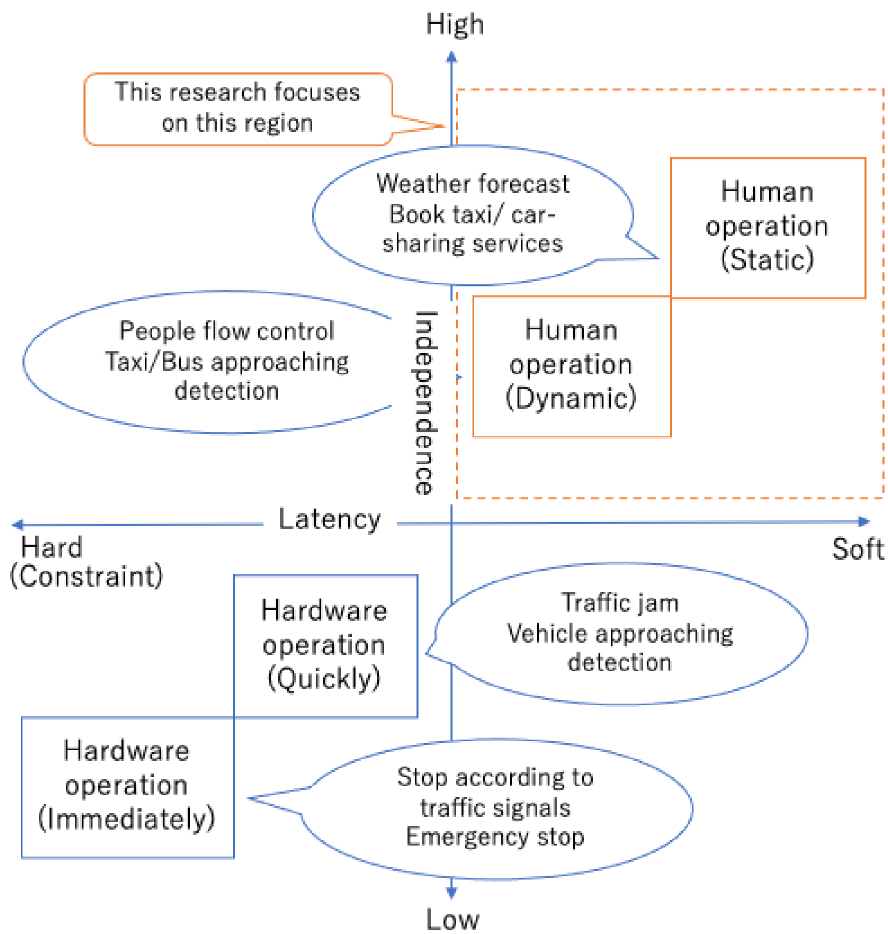


Figure 1.15: Relationship between latency and independence

Different latency requirements for different cases. For example, real-time processing of hardware wants hard requirements about latency. On the other hand, latency and independence among stakeholders have a relationship. This study indicates it uses IoT platforms. Moreover, it has determined this study's target: they require soft constraint and high independence.

of users who want to receive it.

- A sender need not be aware of the recipient unless the target is limited.
- Platforms can be anywhere in physical space.

The third point is important because sending time may be delayed when a sender, a platform, and clients are far away. Therefore, it is must need speculative data delivery to solve this problem. In addition, it is a must to define spatio-temporal information to realize speculative data delivery.

Besides, the previous section described IoT and smart cities, including the number of sensors and devices that will increase. In addition, described the IoT platform is important in the future. This section describes the data processing for many sensor data and the highly complex data delivery for people.

1.6.3 Existed Effective Control for Data Delivery

There are studies about effective control for data delivery. The methods of data transmission are various. This section indicates the existed studies about it.

The spatial information is used to optimize data delivery. First, at pull-type data delivery, there developed applications and studies using spatial information [56, 57, 58]. For example, Yahoo News[59] has realized the optimized information suggestion per user using users' location. Nevertheless, these services cannot provide automatic data receiving completely without the user's operation because of the feature of pull-type. Hence, the user may miss if the sender sends new data at a high frequency.

On the other hand, it describes existed studies using push-type. Location-aware pub/sub-systems use spatial information. The systems send data to users who correspond to an event's location. It assumes that the position of a user is static. Therefore, the purpose is to ensure that dynamically generated data is not missed. Therefore, these systems do not consider that the user updates his/her position. Besides, there are continuous spatial queries. These studies were developed by extending content-based filtering. Content-based filtering [60, 61, 62, 63] has been developed to realize data delivery without ordering a specific topic. The method uses keywords or values instead of topic designations. Therefore, the filter can express conditional expression. For example, when a keyword is weather, and the value is sunny, the system sends data only if the data has weather and sunny.

Continuous spatial queries use keywords as the user's interest.[64, 65, 66, 67, 68, 69] The systems publish queries continuously by keywords and user's location information. Moreover, if the user does not move, the query can continuously publish without updating his/her location.

This method requires specifying specifically that the user is interested in something. This is effective when a target is event information such as sightseeing and special sales. However, in the case of the data generated by sensors, the data is generated continuously and at a high frequency. Hence, the systems that use the data must receive it continuously. Furthermore, in this case, it is important to get the data from the correct topic, not from a specific value. The reason is that the clarity of the sensor disappears if the focus is only on the value.

1.6.4 Existed Spatio-temporal Data Processing

Various collaborations realize data applications. For example, geographic information systems (GIS) are used to manage spatial information systems. GIS is technology to process, manage, and visual for spatial information. GIS data is defined as the information that indicates the specific position or area in the space, the information of the relevant phenomena about it, or the simple location information. GIS data is specialized to handle spatial information. Hence, the related works using GIS data are many [70].

The first example uses GIS to visualize and analyze the spatial characteristics of a target phenomenon. McLafferty challenged understanding the spatial organization of health care using GIS [71]. When organizers consider convenient health care services, geographic variation in a population is essential information. The population per area differs along with many factors, including age, gender, culture, and more. The review paper described the challenges, including visualization of the population in an area, evaluating to assess travel along with transportation networks, including the network distance [72] and travel times [73] by calculating GIS data. It indicated able to analyze multiple aspects of the elements needed to support health care.

Moreover, For example, in recent years, the pandemic of COVID-19 has been a major problem worldwide. The reason is that the virus has strong infectivity. Mollalo et al. [74] have analyzed the COVID-19 incident rate in the continental United States. GIS data have visualized the incident rate, and it is expressed in 35 explanatory variables. The major categories are socioeconomic, behavioral, environmental, topographic, and demographic factors. They indicated it on geographically weighted regression (GWR) and multiscale GWR. They have estimated the correct model to analyze the COVID-19 incident rate. The study is cited related works [75, 76]. These researches also expressed the various European regions using GIS data and the spatial regression model. It has an impact, indicating the effectiveness of the GIS data application in analyzing COVID-19.

Another example, GIS is used to estimate various situations. Schroder et al. uses GIS to implment 3D-Routing-Model [77]. Currently, reducing CO_2-emissions is a global problem. They aimed the optimize the driving route to reduce CO_2-emissions. The map is expressed in three dimensions using GIS data. It considers elevation, slope direction, and positive or negative road inclination attributes. Hence, the routing model can optimize the driving route considering the road in real-world characteristics.

As these related works studies have shown, spatial information processing using GIS data is being done in various ways.

On the other hand, time processing means analyzing the previous X. X indicates the various targets, for example, measurements, experiences, behaviors, and more. Time information is a continuous element. There are many and many using time applications currently. This reason is that the time element is always necessary for analysis. For example, it is purchase behavior analysis. In sales strategy, users' purchase behavior analysis is important. A user always purchases to satisfy the user's requirements. [78, 79, 80, 81] The purchase history contains the trend of the user's purchase potential. Hence, a sales company found the user's purchase behavior by analyzing the history.

Bhagat et al.[81] presented the approach they developed for modeling repeat purchase recommendations using a user's purchase history. They focused on a user's repeat purchase behavior. Their study analyzed large-scale purchase data that e-commerce websites collected. They defined some models, including the aggregate time distribution model and the repeat customer probability model. Their developed system indicated the effectiveness of increasing users' repeat purchases in their demonstration. As in this study, the time element uses to express history.

1.7 Summary

This chapter has described the basic concept of this study. First, it explained what makes the city smart. (smart cities) In this study, smart means it is able to change, solve, and improve subjects in a city using information power. Various sensing technologies are important to realize. This chapter introduced a part of sensing studies, IoT, Participatory Sensing, and Smart Sensing. Moreover, it indicated the actual practice of Smart Sensing. The practice focuses on public cars because they must drive citywide to do their daily routine. This study suggested and developed a method to detect road damage. The result implied that edge computing and public cars could collect citywide data.

From the background, the amount of data in a city will increase; then, the platform for data delivery will be more important than now. This chapter described data focused on a spatio-temporal layer, and it indicated that in-city data can is expressed in it. Last, it described the middleware system and the related studies. The next chapter explains the spatio-temporal information in this study.

Chapter 2

Spatio-Temporal Information Processing

This chapter describes the definition of spatio-temporal information and the basic elements to construct it. All sensor exists in real space, so they have common information. It is time and spatial information. When considering using some data combinations, it is necessary to cluster among various types of sensors. A simple clustering example is what gathers the same type of sensors. In this case, comparing, calculating, and analyzing them is easy. As described, the common elements are necessary to cluster the multiple sensors. This study handles and clusters multiple sensors using spatio-temporal information as a common element.

First, it converts the figure of real space to three dimensions using the time axis and spatial axes. This study defines the three-dimensional space spatio-temporal information. Moreover, all applications of using spatio-temporal information processing are described in the three dimensions defined in this study. In addition, it indicates the vital point to solving each problem.

Next, it describes the definition of basic elements and indicates how to use each element.

2.1 Use Cases of Spatio-Temporal Information Processing in Smart Cities

Spatio-temporal information processing is helpful in various situations. This section describes some examples of use cases that this study targets. The use cases have some

Figure 2.1: Usecase of MaaS

Drivers send route information to an IoT platform. IoT platform publishes it speculatively using spatio-temporal information. Users can receive the carpool services flexibly by it.

common requirements. First, the use cases aim to integrate or collaborate various data from several publishers. When the platform processes spatio-temporal information, it can flexibly control data delivery. Hence, it publishes some data at the same time from different senders. It will lead to integration and collaboration among some data.

Second, some users or applications in a situation use the standard template for receiving data. It shows that some clients aim for the information for the same purpose. Therefore, a data delivery platform can send the same data using the standard template. When the excepted case, templates depend on per client. It means the use case is difficult to share with some clients because it depends on the interest or privacy of each client. This section describes example use cases on these requirements.

Applications for Mobility as a Service (MaaS):

In recent years, the Ministry of Land, Infrastructure, Transport, and Tourism has allowed carpool services in Japan. Carpool services are helpful for people who do not drive cars. For example, in rural, there does not develop the public transportation enough to move all people there. Hence, there is an increase in elderly drivers. It causes traffic accidents, but they need cars because they cannot make a living if they do not drive them.

Carpool services let resolve this problem. In Toyama Prefecture, Asahi city's municipality provides an official carpool service for citizens. The service realizes the carpool services through the cooperation of the municipality, transportation companies, and citizen drivers [82]. At current Carpool services, users register the destination, the point of departure, and the estimated use time. The system matches the users near their destination before they want to use it. However, when it is a little errand, for instance, shopping for daily necessities, daily hospital visits, Etc., it is tiresome to register plans. Then, if the users know the routing plan of taxis and know a taxi will come through near them, they can register the request to the taxi then.

The spatio-temporal information processing help this service. The information users must register the same. It is their location and destination. The platform must send data of some clients to taxis following spatial information. It also needs to send the user's request to the taxi in advance, as the taxi needs to know in advance where the desired user is. (Figure 2.1). Then, the platform must realize speculative data delivery.

Applications for Disaster Prevention: In the real world, there are often happened various disasters. When large-scale disasters happen, people cannot make a living safely. The reason depends on some factors. For example, people cannot go outside to shop for daily necessities when heavy snow, typhoon, or earthquake. Moreover, recently, the large-scale COVID-19 pandemic has attacked all over the world. Some governance ordered lockdown for people then. In these situations, people encounter food and daily necessities shortages because they cannot go outside freely.

Besides, older people living alone have difficulty solving problems that require a workforce. They could not remove snow when heavy snow attacked their house and area. Therefore, the government must prioritize removing the area's snow. Then, efficient data delivery is important for people and the government. Governments can determine the route of removing snow using the data. In addition, this uses to help each other in neighborhoods. For instance, people living in the same apartment can communicate under heavy snow or typhoon because they do not need to go outside and move within the building only. Therefore, when they are in an emergency, they communicate and help each other using the platform and spatio-temporal information processing.

Applications for People Flow Control in Large-Scale Event Situations

It is not good to gather many people within a small range currently because it has a risk that causes infection of COVID-19. When it holds live concerts by famous mu-

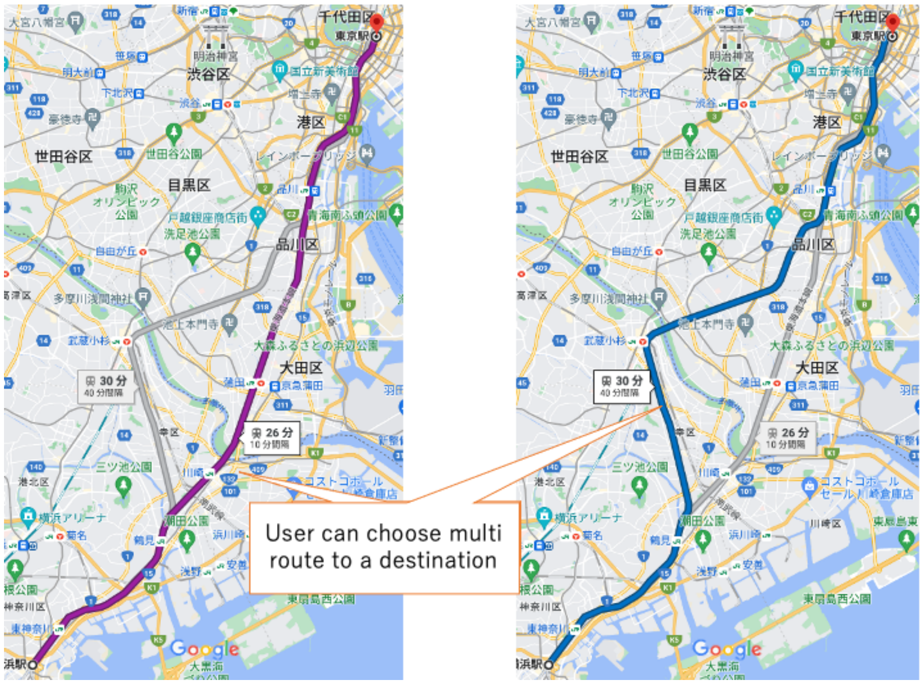


Figure 2.2: Usecase of people flow control

People can choose the riding public transportation to a destination because urban areas have developed public transport.

sicians, watches sports games, and more, people gather there. Their place is equipped with infection control measures, such as separating seats, not talking to each other, and not yelling. Especially in the case where the place is urban, the people flow so big. Therefore, many people must gather at a specific station, train and bus. It generates a high risk that causes infection of COVID-19.

Also, in this case, spatio-temporal information processing is effective. Urban areas have developed public transport. People can choose the riding vehicle from some choices (Figure 2.2). Speculative data delivery supports it. When trains are constantly publishing arrival and stop times, people can know in advance, via speculative data distribution, which train they can catch when they arrive at the station. Then platforms are required for suitable data delivery per people's area. Therefore, spatio-temporal information processing is effective and important.

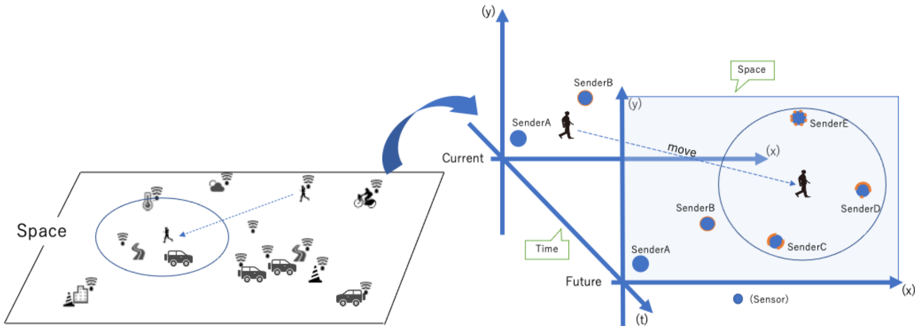


Figure 2.3: Convert to three-dimensional space

All data on real space can express in spatio-temporal. This image shows the result that data in real space convert to three-dimensional space, spatio-temporal.

2.2 Definition of Spatio-Temporal Information

The multiple data collaboration must need the common parts. For example, sensors to measure temperature generates the same meaning value. Even if the values are different for each sensor, they all indicate temperature. Hence, they can collaborate to calculate the average, max, and more. In other cases, sensors with different meanings, such as humidity and temperature, can be combined. They do not have the same meaning value, but their value includes the same category (environment). In order to combine various and multiple data, it is necessary to find common parts.

This study focuses on a thing that all data on real space can express in spatio-temporal. It means they have time and spatial information. Therefore, it converts the figure of real space to three dimensions using the time axis and spatial axes. Figure 2.3 the result that data in real space convert to three dimensional space of spatio-temporal. It reasons spatial information can express two dimensions by location information, and time information can express one dimension. This study defines the three-dimensional space, spatio-temporal information. Next, Figure 2.4 describes the time and space information on the three-dimension model because space information has two axes, x (latitude) and y (longitude). Time has an axis, t (time). Moreover, it describes some examples of applications using Spatio-temporal information.

Last, to define spatio-temporal information must consider the necessary elements to construct it. It is separated spatio-temporal information into time and spatial information. Time and spatial information include some meaning in themselves. The following section first defines what they mean as elements. Moreover, it indicates how to use each defined element.

2.2.1 Applications Using Spatio-Temporal Information

First, this study tries to (A): control data delivery using time. A client is challenged to handle data that he wants to because the number of data and receive timing is chaotic when he wants to subscribe to many topics. Then, This study tries to manage data streaming on pub/sub-model by time. It develops a middleware system to extend an IoT platform. It determines how to handle time elements to develop the system. The system provides some contributions. Users will begin to control data receive timing loosely by time. Moreover, the multiple data are published synchronized when users receive them. It indicates the possibility that multiple different data can collaborate.

Second, it will realize (B): dynamic topic optimization using space information. Basic pub/sub-model implemented topic-based data publishing. Subscribers can receive data via the target topic, and they must subscribe to the topic prior to receiving it. Nevertheless, when a subscriber moves on space, the topic they want to get, but subscribing to all targeted topics, always wastes network cost. This study tries to implement a method to change the target topics' dynamic by subscriber's location information. It provides a method to express spatial information on the comprehensive system.

Third, the goal of (C) and (D) data clustering and aggregation are to expand (A) and (B) systems to realize aggregation on the platform. The key to solving this problem is to build on (A) and (B) as one platform. (A) and (B) is each basic model of time and place to implement spatio-temporal information processing. Hence, (C) and (D) problems treats as an expanded system of (A) and (B). Besides, there is some research on data aggregation [83, 84, 85, 86]. An existed study developed data aggregation for one topic using time information [83] Other studies developed data aggregation for multiple sensors near each other in space [84, 85, 86].

Therefore, by indicating the implementation of data aggregation like these, this study demonstrates a spatio-temporal abstraction that indicates data aggregation in time, space, and both.

Last, it will try to (F) implement Spatio-Temporal information processing. This study tries to realize speculative data delivery using time and space. This method uses

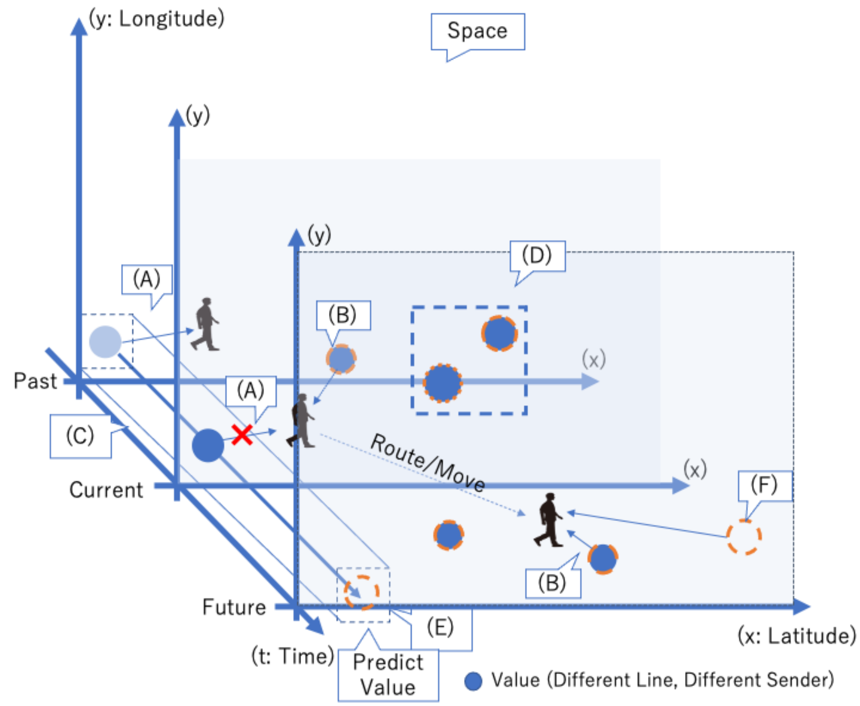


Figure 2.4: Applications using spatio-temporal information

Data on spatio-temporal can cluster, analyze, and collaboration using time and spatial informations. This image describes examples of applications.

defined elements and other information available in mobility systems. It indicates that defined formats and developed systems in this study can extend easily by combining other applications. In this case, it focuses on the routing information of mobility. Mobility systems suggest routes that users want to go to the destination. The route information includes time (duration) and spatial information (waypoints and destination).

Therefore, it indicates an application of spatio-temporal information by a developed system in this study that uses route information to control speculative data publishing.

This study does not try to (E) because this problem is different from other problems. Various previous studies have been conducted on this forecasting problem, and their problem is depended on each piece of information.

This study focuses on the performance of the IoT platform. Therefore, This study does not try to (E).

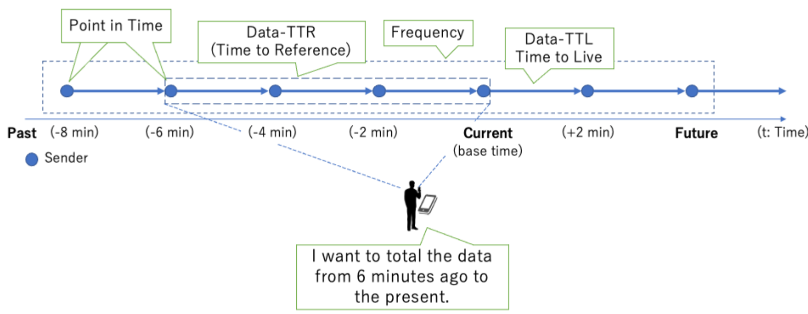


Figure 2.5: Definitions of time elements

Definitions of time elements express one axis and each circle. The axis means time, and each circle expresses the sending data from a sender. Time can express Point in Time, Frequency, Data-TTL, and Data-TTR.

2.3 Definitions of Time Elements

Time is the standard information that all data have. Time has some means, and its usage depends on its meaning. This study defines three kinds of time by meaning.

The three definitions:

- Point in Time
- Frequency
- Data Time to Live (Data-TTL)
- Data Time to Reference (Data-TTR)

This section describes them.

2.3.1 Point-in-Time

First, Point-in-Time means when the data has happened. Each circle of Figure 2.5 shows it. A Point-in-Time indicates the freshness of the data. When the value is old, the data is also old. Users can judge to use of the data by looking at the point in time. In addition, Point-in-Time indicates the log. When an application stores many data from a sensor, the application can analyze those data. Then, the application uses Point-in-Time to analyze data because it indicates the relevance ratio for each data. For instance, when Point-in-Time means continuous value, the data value is also continuous. When two data have the same Point-in-Time, differing only by days, they mean changing the value simultaneously. These analyzed results detect the trend of the data.

2.3.2 Frequency

Frequency means an interval of data sending. Almost all sensor devices send data regularly, and they have time information. Then, the application calculates the interval of the target sensor it understands the frequency of data sending. When the frequency is high, the sensor device sends much data; when the frequency is low, the sensor device sends little data. The frequency is different for each sensor. Therefore, subscribers must consider the value when creating or using the application because they must turn to the collaborating values if they want to collaborate with different sensor values. Nevertheless, it is not easy when they want to use many sensors. Therefore, this study aims at the practical use of frequency.

2.3.3 Data Time to Live (Data-TTL)

Almost information always changes its value. Environment sensors are an example of changing the value at a very high frequency. Weather information is updated every hour on the famous weather report applications, but the temperature value changes frequency higher than the weather.

On the other hand, people's events cause updated frequency lower than the previous example. For example, a shop's special sales information is updated daily. Nevertheless, the special sales information does not change easier than temperature information. Then, the special sales information's time to live is more protracted than temperature information's. As mentioned above, information has each time to live. The value is called Data-Time to Live (Data-TTL) in this study This value is used to judge whether subscribers use the received data.

2.3.4 Data Time to Reference (Data-TTR)

Last, this study defined Data Time To Reference (Data-TTR). Data-TTL expresses the limited time to guarantee the freshness of the information. However, data processing applications want to use old data to calculate and detect various information. For example, the average, maximum, minimum, and mode of sensor data for one hour are calculated by old stored data. Therefore, this study defined Data-TTR for users. Data-TTR indicates the time range that a subscriber wants to aggregate. Data-TTR has defined the same value as the receive frequency in this study. For instance, The Data-TTR is ten minutes when the receive frequency is 10 *.

2.4 Definitions of Space Elements

This study treats location in the physical world as space information because objects' location can be expressed by latitude and longitude values. In addition, location information is also the standard information that almost all data on the physical world have. If the device does not have it, it can collect easily by installing a GPS sensor.

The location has some means, and its usage depends on its meaning. This study defines three kinds of time by meaning.

The three definitions:

- Position
- Distribution
- Data Area to Live (Data-ATL)

This section describes them.

2.4.1 Position

First, Position indicates the location of data where it has happened. The environment and traffic information happens in the physical world, and their information has the location information, including themselves. Besides, stores, parks, and other facilities exist in the physical world and have location information. Therefore, almost all data has it. This information is important to detect the center of the data. About Distribution and Data-ATL will describe later the center is essential to determine them.

In addition, this parameter uses to indicate subscribers' location. Therefore, position is the most basic definition of space information.

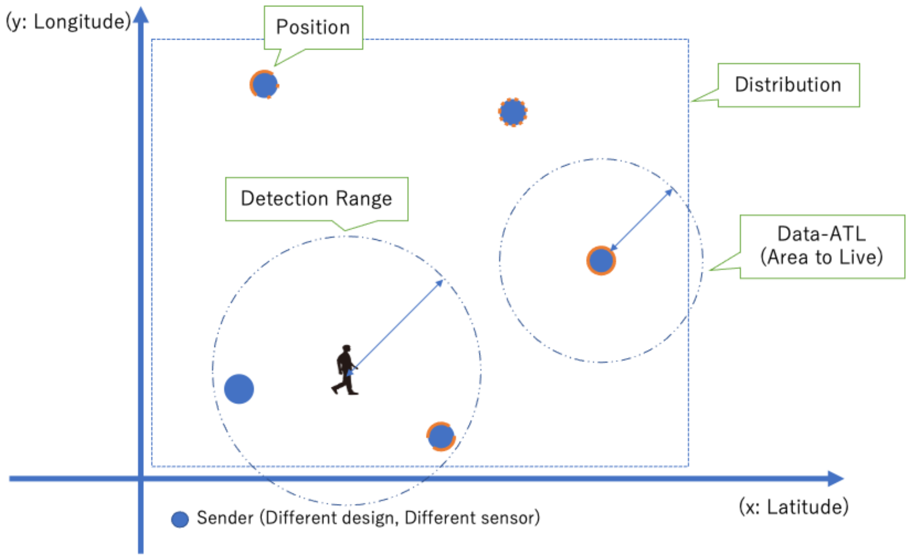


Figure 2.6: Definitions of spatial elements

Definitions of spatial elements express each circle and two-axis, latitude and longitude.Each circle expresses different senders. Spatial can express Position, Distribution, Data-ATL.

2.4.2 Distribution

Distribution means the scattered sensors in the physical world. It can indicate the relationship between sensors by their distance. There are many sensors worldwide, and their number will increase in the future. Then, for applications, we will want to cluster them by some conditions. Location information is also included. For example, when the two temperature sensors have been installed near each other, they detect from almost the same area. However, their value is not necessarily the same. In this case, a subscriber can not only select the sensor but also calculate the average, max, and min values from their sensors.

Therefore, distribution is important when subscribers want to aggregate multiple data.

2.4.3 Data Area to Live (Data-ATL)

Data Area to Live (Data-ATL) indicates the effective range of the data. Data-ATL basically spreads from the point of the sensor as the center. This area is different from each sensor's spec, kind of it, and the publisher's definition. For example, if a subscriber wants to receive weather information near him, then the value of the long-distance sensor does not have meaning for him. Similarly, a subscriber wants to search for special sales information, but the system sends him to it from a long distance, then the subscriber cannot go to the store, or it is difficult to go there.

From these examples, Data-ATL is important to judge subscribers and applications whether the data uses or not. In this study, the value is defined by each publisher.

2.4.4 Detection Range

Detection Range means the range that users want to search data on location. Almost all data exists in the physical world, so they have location information. In addition, it depends on the location when users search for data, events, and more. Nevertheless, the location has expressed a point if it only uses position. It is too narrow to search for data. Hence, this study defines the detection range users want to search. It can express the area to detect data according to users' POI.

2.5 Summary

This chapter writes about the definition of spatio-temporal information in this study. All data on real space can express in spatio-temporal. It means they have time and spatial information. Their information does not depend on the type of data. It means all data always have their information. Therefore, this study focuses on spatio-temporal. First, it converts the figure of real space to three dimensions using the time axis and spatial axes. This study defines the three-dimensional space, spatio-temporal information. Moreover, it indicates examples of applications using the information.

Next, spatio-temporal information is defined as collaborating time information and space information. Time information uses to control data send timing and to cluster continuous data from one sensor. Besides, Space information is used to judge the target topic subscribers want to get and cluster different sensors handling the same information.

Moreover, this chapter describes definitions of time elements, Point-in-Time, Frequency, and Data Time To Live (Data-TTL). Point-in-Time indicates the timestamp of happened or published data. Other time elements are based on this value. Frequency indicates the interval time of published data.

In addition, the study defined an additional time element to realize this approach. It is Data Time To Reference (Data-TTR). The amount of data depends on the subscriber when the system provides data aggregation because the target is different among subscribers. Data-TTR indicates the time length of the target data.

This study described the definitions of time elements and the implemented system using them from the above. In addition, it indicated the possibility of them.

This chapter describes definitions of space elements, Position, Distribution, and Data Area To Live (Data-ATL). Position indicates latitude and longitude as location information of the data. There are all the data on the physical world has the information. Other space elements are based on this value.

Distribution indicates the scattered sensors in the physical world. Therefore, this means the relationship among sensors by their distance. The system can cluster the sensors using this definition. For example, when the subscriber wants to measure the temperature value correctly, the system clusters his location and sensor distribution near him to calculate the temperature's average or max or mode.

Data-ATL indicates the effective range of the data. This value is essential to determine whether or not to send data to the subscriber. When the sensor's effective range is so short, target subscribers are also limited. If the system does not consider this value, it publishes wastefully.

Chapter 3

Development of Spatio-Temporal Information Processing Middleware to Extend IoT Platforms

This chapter describes the actual developed spatio-temporal information processing systems. The systems are used to extend IoT platforms. This section determines the delivery messaging model on IoT platforms. The model is pub/sub-model. The developed systems provide various functions to control data delivery.

First, it explains a smart data streamer (SDS). The system provides the function using time information. It prevents data overflow by managing users' requirements. Moreover, it realizes synchronized data delivery and data aggregation.

Second, it explains a dynamic topic optimization system (DTOS). The system provides the function using spatial information. It realizes dynamic changes to target topics. IoT platforms can reflect users' position in data delivery by DTOS.

Last, it indicates speculative data publishing using spatio-temporal information and spatio-temporal information abstraction. SDS and DTOS have almost a common data format. Hence, their data format can combine easily. It indicates spatio-temporal data abstraction.

3.1 Determine Delivery Model

First, we need to discuss data delivery models. Devices connect applications or clients in two ways, direct connection by a cable or remote connection by networks. If a device uses a direct connection, almost all of the case application and device constructs a peer-to-peer network, so the device's data is only sent to the connecting application. This connecting method is effective if the application requires very high reliability and high-speed data transfer. In addition, This method proposes high data security when it transfers to the connected device because it may intervene in other applications to this connection. Therefore, this connecting method uses susceptible machines' operations, or the machine cannot use computer networks. On the other hand, this connection is unsuitable for data delivery to many devices simultaneously. If using this method to deliver data to many devices, the core device needs to have many sockets to connect each device, so this method does not really.

In the case of these, remote connecting methods are very suitable. Remote connecting methods use in computer networks, so this method does not need the physical sockets to connect each device; the device only needs the function to connect the networks. At Remote connecting methods, devices can exchange data from multi devices at the same time. Currently, many client applications use this method because a server must connect many applications to various existing locations simultaneously. This study focuses on data delivery in various situations, so consider sending data to many clients and various applications in various locations, such as portable devices such as smartphones and autonomous cars. Therefore, there use the remote connecting methods in this study.

There are various protocols on it, from clients, applications, or servers to others. The protocol of these is used in practical situations by selected developers. This section describes two messaging models they mainly use and select the messaging model in this study.

3.1.1 Pull-Type Messaging Model

The pull-type messaging model uses on general web services. Clients have the authority of control to receive data in this model. Figure 3.1 shows a basic concept. The client sends a request when a client wants data from the server. The server sends data when it receives the request.

Figure 3.1: Image of pull-type messaging model

When a client wants data from a server, the client needs to send a request. In this messaging model, clients have the authority of control.

This model's feature is its high reliability. This model almost uses TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) to exchange data between servers and clients.

First, TCP has several functions to guarantee high reliable data exchange. For example, three handshakes provide an exact connection between a client and a server by checking a connecting peer. In addition, it has the functions of resending packets and revising errors. Therefore, this protocol can propose highly reliable data delivery, but a load of these functions is not lightweight, so this protocol is unsuitable for real-time data exchange, such as voice calls, video streaming, and more. This protocol is used on HTTP (hyper Text Transfer Protocol) and FTP (File Transfer Protocol) because these protocols require that what is sent data is not broken absolutely.

UDP can solve this problem. UDP is more lightweight than TCP, so that it can do high-speed data transfer. This protocol does not have several functions, which TCP has, for instance, three handshakes. Therefore, the overhead of UDP is more negligible than TCP, so it is very lightweight, but it cannot propose high reliability. This protocol uses the situation where it requires real-time data transfer or to multi-cast as voice calls and movie streaming. It causes that it sometimes lacks raw data on those applications. The protocol of them is used based on pull-type messaging mainly.

Database systems such as MySQL and PostgreSQL and the web service using HTTP use this messaging model. It is high reliability because this model uses TCP. However, the trigger depends on the client's activity, so if the server application generates or updates data at a high frequency, the client has the possibility that it cannot get all data from the server. As shown in Figure 3.2, the request and send process is done per client. Therefore, this messaging model uses services that are unnecessary to

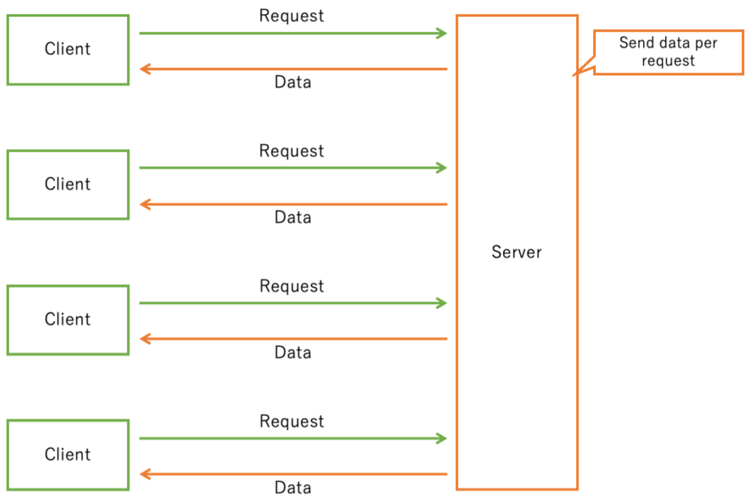


Figure 3.2: Example of push-type messaging model

Data sending timing depends on the client, and the process is done per client. Each client needs to send a request if they want the same data.

update information high frequency, such as SNS and Location News.

3.1.2 Push-Type Messaging Model

As shown in Figure 3.3, This messaging model does not need to inquire from users. A connection based on this model is sent to the client from the server immediately when the data provider sends data to the server. When an application uses this model, the application connects to the server continuously, so this messaging model consumes computer network costs more significantly than the pull-type messaging model. However, it can send data to a client immediately, so this model can prevent sending data to the client. So, this model is useful when a system or data provider frequently generates new information. Websocket protocol is implemented by using TCP. This protocol can guarantee high reliability by TCP and can do push-type data transfer. Nevertheless, this protocol can only use on HTTP, so if the system does not use HTTP, it cannot use it. Besides, there implemented several protocols based on the push-type messaging model to use various situations. This study describes four main protocols for using the push-type messaging model. These models call Publish/Subscribe messaging model

Figure 3.3: Image of push-type messaging model

Clients do not send a request to receive data. The server sends data when happens a trigger about data sending.

(pub/sub-model). An extended push-type messaging model implements the model. Figure 3.4 shows the overview of pub/sub-model.

3.1.3 Publish/Subscribe Messaging Model

First, XMPP (Extensible Messaging and Presence Protocol) [87] is implemented for instant messaging services. This protocol is constructed message based on the XML (Extensible Markup Language) format, so its feature is the flexibility and extensively of messages. In addition, this protocol can exchange a large size of a message; for example, it does not only exchange simple text messages but pictures by converting BASE-64 format. On the other hand, the overhead is also largely because the XML format's overhead is significant.

Second, AMQP (Advanced Messaging Queuing Protocol) [88] can exchange messages by its overhead is more petite than XMPP. AMQP is implemented based on queuing protocol and a push-type messaging model. This protocol provides various delivery message methods to guarantee high reliability. For example, this protocol provides a messaging directivity. AMQP can also realize sending messages to some clients at the same time. However, in a financial case, it wants to change sending address depending on the content of the message. In this case, AMQP can manage the sending address dynamically. In addition, this protocol has been constructed based on the queuing protocol and provides QoS (Quality of Service), so it can guarantee to send orders exactly. Therefore, this protocol has very high reliability.

Finally, MQTT (Messaging Queuing Telemetry Transport) [89, 90] is the most lightweight protocol of the three. This protocol mainly uses IoT scenes. The best feature of this protocol is the simple format, so it realizes the lightweight protocol. It

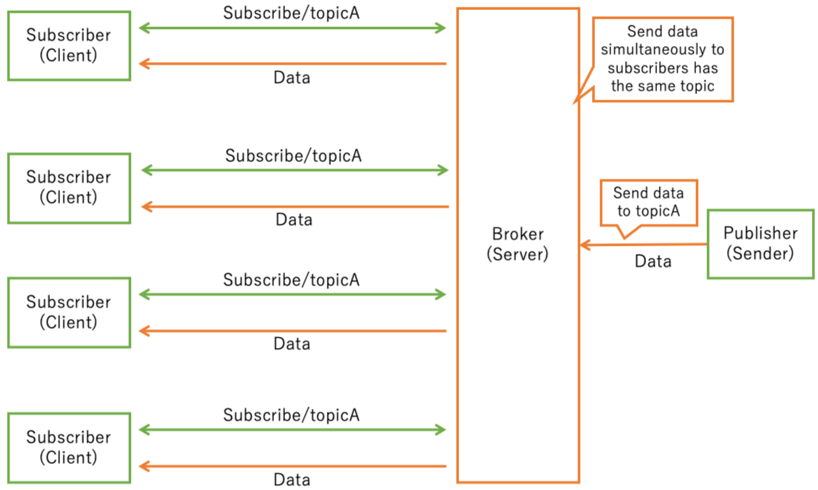


Figure 3.4: Overview of publish/subscribe messaging model

Clients (Subscriber) continuously connect to a server (Broker) via topic. When the sender sends data to the topic via the broker, the broker simultaneously sends data to subscribers that have the target topic.

can realize high-speed transfer massaging to some clients at the same time by using its feature. In addition, the protocol is also for energy conversation. In the aspect of reliability, it also has implemented QoS, so it can guarantee to send messages to a client rightly. However, it is possible to miss sending data, including lack of contents and missing order among some messages, because it does not have other functions to guarantee reliability.

From these features, its protocol is familiar to edge devices. The edge device means it has a micro-size body, a low processing CPU, and a small battery to use in various situations. [91]

All of these protocol has a typical network structure and connecting methods. There are three roles in a push-type messaging model, publisher; subscriber; and broker. Publisher means data provider and its generator. Publishers provide data to other clients via the server. Then, a broker makes do the role of the server to send data from publishers to subscribers. Last, subscribers receive data from publishers via the broker, so a subscriber means data receiver, and a broker means the server among publishers and subscribers. In the push-type model, a broker has an important role. Subscribers and publishers can realize independence from each other by using it, so whatever system halts, other applications are not affected by the halted application. In addition, the broker provides sending data to some subscribers at the same time. Then, with topic-based filtering as a general filtering model in the push-type messaging model, subscribers and publishers select the topic as the address. Subscribers have to register to the topic on the broker they want to subscribe to before receiving the information from it. On the other hand, when publishers send data to the broker (publish), they select the topic on the broker.

These features are familiar to IoT scenes because almost all applications and devices are independent of those scenes. Therefore, in this study, we aim to extend the platform implemented based on the pub/sub-model. The selection of optimal protocols in this study will write after the section.

3.2 SDS: Development of Data Delivery Middleware Using Time Information

This study focuses on IoT platforms based on the pub/sub-model and implements the subscriber-driven data delivery method using time control. The system is smart data streamer (SDS)[6]. This study aimed to prevent data overflow to subscribers and synchronize data from different senders. Moreover, the system can simultaneously deliver data to some subscribers who have the same target topics. SDS expands on the function of IoT platforms, and it is not only realizing subscriber-driven data delivery, but it has the function of util for subscribers, storing and showing information on topics. Therefore, SDS improves the usefulness of IoT platforms.

3.2.1 Requirements

SDS has two main requirements. First, subscribers can set data delivery timing from the IoT platform using the soft condition. Related works control by using content-based filtering, but the filter will become complex when the subscriber wants to receive several data with different send frequencies. This system aims to archive the sending control by using only time information. Therefore, subscribers do not need to set a complex filter or order many condition keywords. Second, subscribers can receive several data synchronized. Figure 3.5 shows the sensor data transmission in terms of frequency.

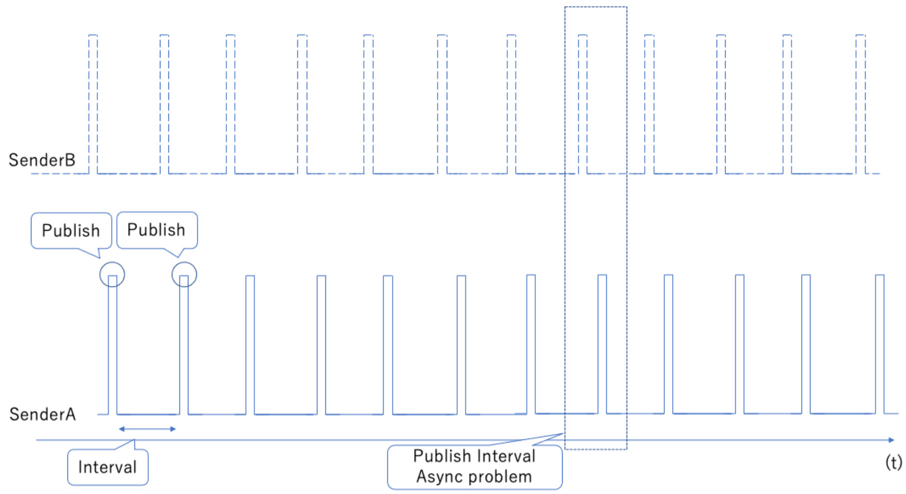


Figure 3.5: Publish interval async problem

Data sending depends on per sender. A user needs to consider the method to synchronize data when he wants to use data from some senders simultaneously.

At Figure 3.5, senderA and senderB send different inteval. In this case, the subscribers must implement the buffer system to combine because their data are sent from the IoT Platform to a subscriber independently. Therefore, this study defined subscriber-topic as an abstract topic. Figure 3.6 shows the image. The topic roles synchronize the different target topics and send all target topics data to the subscriber. The function provides complex data processing by collaborating different topics' data.

3.2.2 System Configuration

As Figure 3.7 shows the system configuration of SDS, the data sending function only connects to platforms. It means the system's independence because it does not consider the policy of the connecting IoT platform so that it will extend the system quickly. As described in the section 3.1, there are platforms of several types, so there are also several formats of publishing data. The protocol and the platform are decided about what they want to do and using environments. The case of SDS depends on each platform or protocol causes applications can not freely select the combination of the system and the platform, and SDS uses limited situations. Therefore, information about

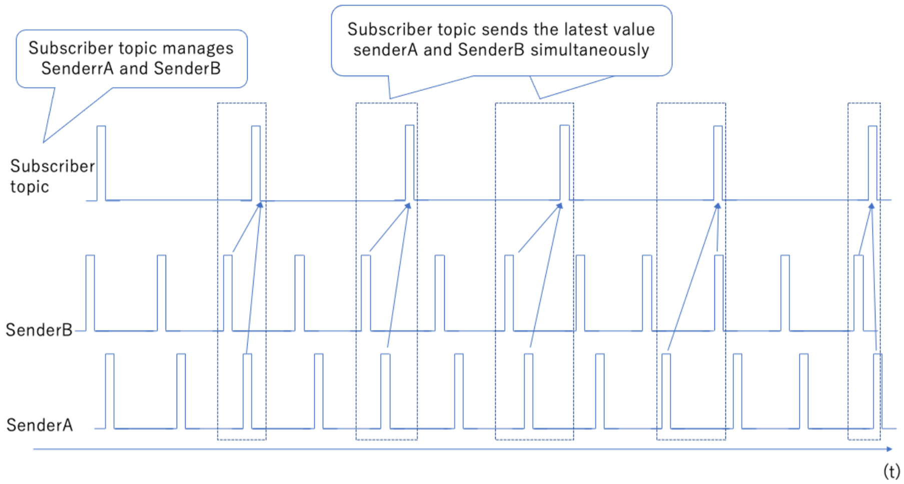


Figure 3.6: Image of working subscriber-topic

A subscriber-topic manages several topics, and subscribers subscribe to the subscriber-topic only. The subscriber receives data synchronized via subscriber-topic because it sends data from several topics as one topic.

topics and subscribers has on SDS, and only the data sending function connects to the platform. As a result, publishers do not consider the implementation to suit the using protocol. Subscribers can receive data in the same format because SDS decided without the policy of each protocol to understand the structure of the data quickly.

In addition, SDS is implemented as an independent system from platforms because this structure provides the future processing function's scalability.

Therefore, this study describes functions except for data transmission to platforms and creating topics on the platform. SDS has the following functions.

- Create and manage topics.
- Control data transmission by the time.
- Manage subscriber-topics.
- Index topics.

About subscriber-topic is discussed later. The procedure for data transmission is as follows. A publisher requests the topic creation to SDS.

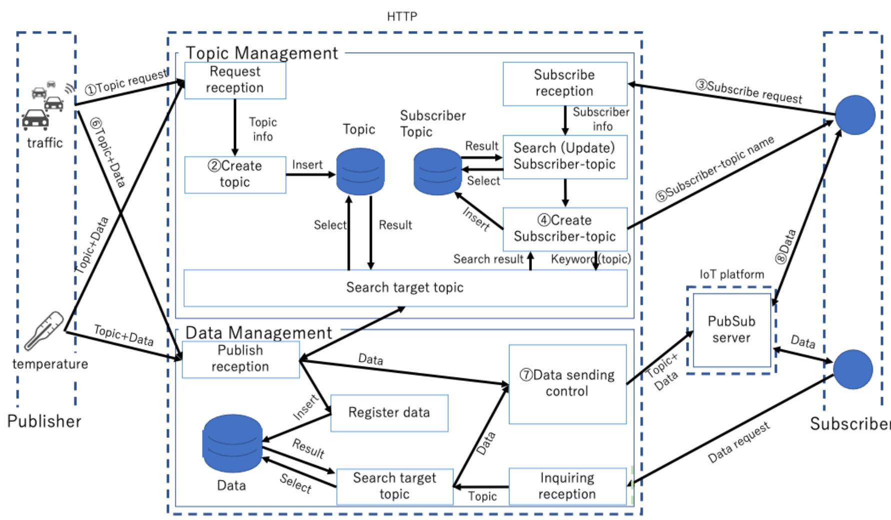


Figure 3.7: SDS's system configuration

SDS performs among publishers, subscribers, and brokers. The system uses HTTP to connect publishers and subscribers. The protocol to connect a broker depends on the broker's protocol.

1. SDS creates the topic following the request.
2. SDS stores the data when it can confirm the target topic after the publisher sends the data to the topic.
3. A subscriber requests the subscribing topic.
4. SDS checks if there is a subscriber-topic with the same name as the subscriber. When it exists, SDS sends the name as the response. SDS creates the subscriber-topic and sends the name if it does not exist.
5. SDS controls to send data by the time. SDS checks the target topic every time, and if the target subscriber-topic's current time and periodic time, SDS sends data to it.

Finally, SDS uses HTTP to connect publishers and subscribers. Users connect the system only when publishers send data or requests about the topic creation, and subscribers also request or inquire about the topic. (Figure 3.7) Then, the most important is the reliability of requests and the data sending, and publishers and subscribers do

```
{
  "Publisher": "test_user",
  "TopicName": ”fujisawa_weather",
  "Type": "Environment_Sensor",
  "Location": {
    "Latitude": 134.19824,
    "Longitude": 56.7869214
    },
  "Elements": {
    "thermometer": { "Unit": "C" },
    "liquid": { "Unit": "rh", "MinValue": 0, "MaxValue": 100 }
    },
  "Descrption": "sample create request”
}
```


Figure 3.8: SDS's publish JSON sample

Necessary elements are expressed as key and value. It aims to improve the visibility of users.

not need to connect SDS continuously. In addition, the request to SDS manages some information, including topic name, location, and category. SDS uses JSON Format because the format is high visibility. Figure 3.8 shows the for instance of the format. The complex information can be expressed simply by the nest, and users can easily confirm the information. About each function is described in each subsection.

Create and Manage Topics

SDS also assumes publishers and subscribers who do not know each other. Subscribers need to be able to check the topic information created by the publisher. This study defined the format as Table 3.1. If publishers create the topic following the format, the data format is unified, so subscribers do not have to consider the difference in the format per publisher.

The name of a registered topic cannot duplicate other topics as a rule to guarantee uniqueness. When a publisher tries to register a new topic on the server, the publisher cannot register it if the server's existing topic has the same name. In addition, the publisher who creates the topic only can send data to the topic because if multiple publishers can send data to the same topic, the topic's data is not dependable. SDS

Table 3.1: Properties of Topic Creation on SDS

Property	explanation
Publisher	Sender name
TopicName	Topic name
Type	Type of topic
Location (optional)	Location information (Latitude, Longitude)
Elements	Set contents (optional: unit, value range) A topic can has some elements
Description (optional)	Description about the topic

confirms each name of a publisher and a topic creator whenever it sends data to a topic, and data is published when it matches each name only. Finally, the existed sensor in the real world has geolocation information. The sensor which is fixed the location does not need to send location information each time. SDS can store the information, so a publisher can send data without the location information after sending it once when registering the topic. As a result, sending data is expressed, reducing publishing costs.

A Function What Send Controlling by Using Time

SDS publishes data following the period subscribers set when registering a subscription on SDS. Subscribers can receive data without depending on publishing timing using SDS. The system defined the minimum unit of the period as minutes and the upper limits of a set period are within a day because the connection cost is higher than the pull model if connecting terms without receiving data are too long. Therefore, this system does not consider the period over a day. This function is realized by monitoring subscriber-topics. SDS monitors all registered subscriber-topics every minute. At every minute, SDS checks the condition of all subscriber-topic by time, and it extracts and publishes data on topics that match the set period and current time. (In this study, the format of a set period refers to crontab, but this system only focuses on continuous data delivery. For instance, when a subscriber sets [5 7], SDS sends data every 5 minutes from 7 to 8. (This system uses 24-hour notation.) The part of the hour can order multiple numbers, so if a subscriber sets [5 7,10], it sends data every 5 minutes, from 7 to 8 and from 10 to 11.) This system realized synchronizing data delivery from different topics by this control and subscriber-topics.

Next, SDS publishes just data with the latest timestamp because this system focuses on synchronizing multiple data. The number of data per topic is different, but it needs to use the latest data and other data to synchronize them.

Create and Manage Subscriber-Topic

At previous section describes controlling data delivery by time. However, the topics that each publisher creates are independent, so the control function does not satisfy because it does not easily manage some topics. Then, this study defined a new topic to manage some topics easily. That is subscriber-topic.

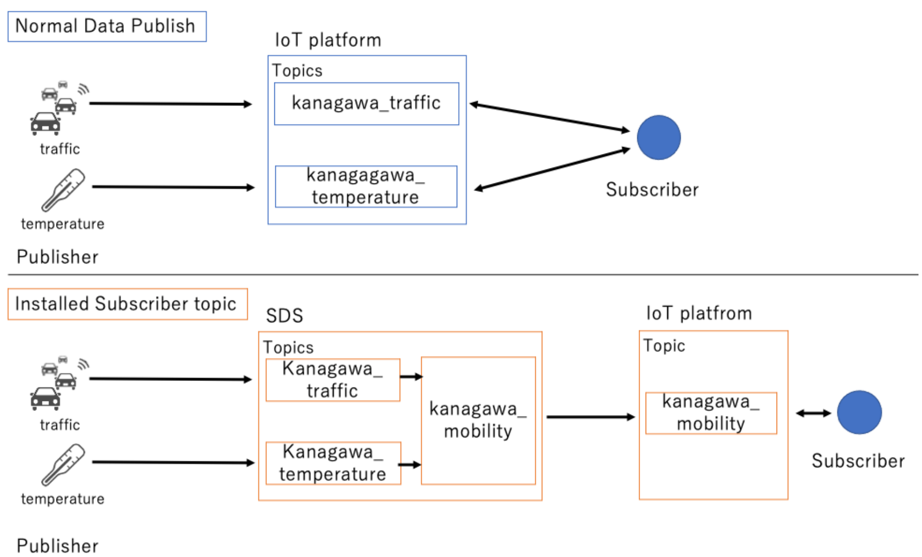


Figure 3.9: Image of how to use a subscriber-topic

A subscriber-topic manages some topics which a subscriber wants to receive data.
Hence, the subscriber does not consider publish interval async problem.

In this study, SDS publishes data from some topics simultaneously, so SDS does not must manage the subscription per topic. The reason is that important is how it publishes data simultaneously. Figure 3.9 shows an image of managing a subscriber-topic. A subscriber-topic has some topics for which a subscriber wants to receive data. subscriber-topic's format shows Table 3.2.

The content of **SubscriberName** of Table 3.2 is important in this system. A subscriber-topic manages some topics as an abstraction, and the name of the subscriber-

Table 3.2: Properties of subscriber-topic creation on SDS

Property	Explanation
SubscriberName	Subscriber name
Purpose	The using purpose of a subscriber-topic
TopicList	Target topic name (List format)
Location (Optional)	Subscriber location (Latitude, Longitude)
ReceiveFrequency	Express receive freqency [minutes hours]

topic does not allow duplication in this system. Subscribers cannot create the same name subscriber-topic. The name is decided by the subscriber's name and the purpose. Therefore, the subscriber-topic name is unique; the subscriber's name guarantees this uniqueness because the subscriber's name does not duplicate.

However, if subscribers want to share a subscriber-topic, a kind name is suitable for the subscriber's name. For example, subscribers want to receive data about traffic information there. Then, the location name is suitable as a subscriber name because the name is common information among some subscribers. Moreover, The purpose of people there is to receive traffic information there. In this case, the subscriber's name can be expressed as LocationName_TrafficInformation. (_ means connecting a word and a word.) It means that it receives traffic information on a specific place. Subscribers can receive traffic information without subscribing to SDS using this subscriber-topic. It realizes to reduce connection costs and the number of subscribers on SDS.

At the topic index, subscribers with the same purpose can easily search the topic by location information when they want to subscribe. Nevertheless, this information is unnecessary. Hence, publishers can set this information optionally. A Subscriber-topic can also remove on the server.

Last, this subsection describes the risk what using subscriber-topic. First, it is a delay in which the subscribers receive the same data. If some subscribers subscribe to the same topics, the topics publish data many times because the subscriber-topic is independent, so data is published per subscriber-topic. However, this system controls by minutes to ignore a bit different times, so this study judges this is not a problem.

Second, it is fatting the cost of connection. If a topic includes some subscriber-topic, the topic's data will be published as subscriber-topic, so the connection costs are fatter than standard publishing. However, SDS will be used to extend an IoT platform, so they exist in the same network space. In addition, the cost does not increase linearly because each subscriber's period does not always match. Therefore, this study considers the increasing cost is light.

Search Topic Function

SDS manages two kinds of topics, subscriber-topic and created topics by publishers. They have different information from Table 3.2 and Table 3.1. SDS has each endpoint to receive each data, and the system has two endpoints to search for interesting topics. SDS can search the list of registered topics and search for detailed information on each topic.

3.3 Data Aggregation Using SDS

SDS provides synchronized data publishing, but this method only uses the latest data. Then, applications cannot use old data during the receive frequency. It losses the use opportunity of old data. This section suggests and implements data aggregation using SDS to use the old data.

3.3.1 Format to Subscribe

The format was extended to the SDS's format. The change point is the TopicList. (Figure 3.10) It manages the topic name when SDS uses it. SDS searches for the name from the database. Then, it detects the latest value from the database.

However, it will need the other information to aggerate because there are various methods and some data to aggregate. TopicList has been extended to express the procedure to aggerate. The extended format is shown in Figure 3.10. The main change points are three.

First, each name does not indicate the specific topic on the database. The name indicates a temporary topic name that a subscriber sets it. SDS uses the name when it publishes the aggregated result. The subscriber can treat the result using it.

Second, each topic includes Procedure. The subscriber describes the procedure by which he wants to aggregate data. Then, the procedure is expressed by combining variables and operators. For example, Add means the addition operator, and Min means

```
"TopicList" : {
"fujisawa_environment_information" :
  {"Procedure":"Hot(fujisawa_environment_sensor)"},
"fujisawa_average_tempreture": {"Procedure":"AVE(T)",
  "VariableList": {
   "T": {
    "TopicName": "fujisawa_environment_sensor",
    "Elements": "temperature"
    }
   }
  }
 },
```

Figure 3.10: Extended TopicList is to realize data aggregation

It can express procedures and variable elements to calculate multiple data.

the minus operator. All operators determined the three characters, and the aggeration operators are in uppercase only. (Defined Add, Min, Mul, Div, SUM, AVE, MAX, MIN, MID, Mod, Hot.) Hot operator is special. It is used when a subscriber wants to latest data only. The New operator get the specific value in a topic, but the Hot operator can get all value in the topic.

Third, each topic includes VariableList. The subscriber describes variables to aggregate data there. It constructs a variable name and topic information to aggregate and calculates.

3.3.2 Data Aggregation

A previous section explained that a procedure is described by combining variables and operators. SDS aggregates registered procedures by their information. It evaluates from the right side of a procedure. In addition, the parentheses are the highest priority in the procedure. However, this evaluation does not consider the priority among the four arithmetic operators because this procedure can include not only it but aggregation operators.

In addition, a subscriber wants to aggregate data when it needs the time range of data. Then, the system uses the Data-TTR per target. Published data have been stored in a value list of each subscriber-topic in the database. The data is stored within Data-TTR, and it deletes after publishing to subscribers by the receive frequency.

3.4 DTOS: Development of Data Delivery Middleware Using Spatial Information

Dynamic topic optimization system (DTOS) will realize the change topic dynamically by users' location. SDS of the described previous section is implemented about time processing only. This section aims at the system extension for location (spatial) information processing. This extension method coexists with SDS. Therefore, this system will manage both time and spatial information.

It is necessary to realize spatio-temporal information processing because the system can collaborate flexibly and efficiently when it realizes it. This section describes the system configuration and organizes the common part of SDS and DTOS.

3.4.1 Requirements

DTOS also provides the two main requirements. First, the system can reflect the subscriber's location information to the IoT platforms. Already, the database systems such as MySQL and Oracle database have the function to compute and detect the value from the stored own it by using their query. Therefore, there is already study using the function. The system's goal does not miss the new sensor value and event to the subscriber who does not move from there.

However, the database system cannot change the latest location information when the subscriber moves from there. Therefore, DTOS will reflect the subscriber's information in the database.

Next, we defined the Dynamic Topic Optimization function as shown in Figure 3.11. When considering the use case of moving subscribers, they change the target topic because they want information about where they have moved.

Therefore, this study defines subscriber-topic for DTOS. This subscriber-topic has common elements with the SDS, but the element indicates a different mean. The system realizes the dynamic topic optimization by it.

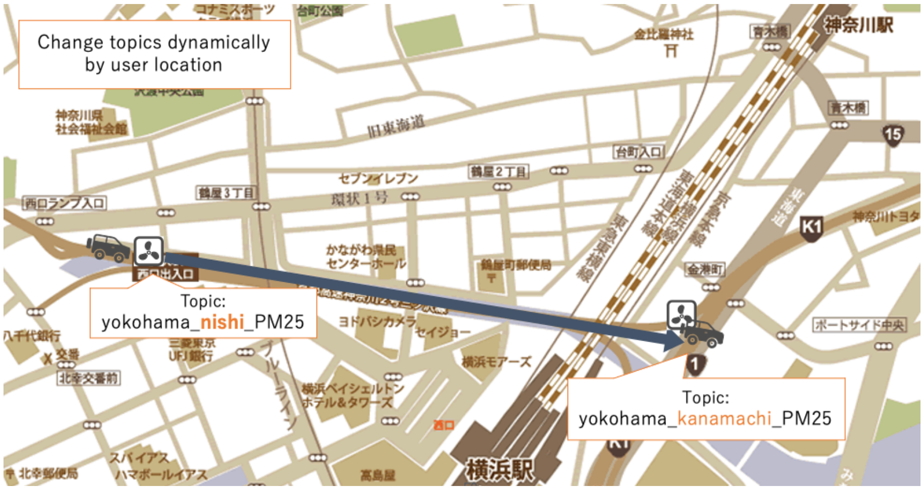


Figure 3.11: Overview of DTOS

Dynamic Topic Optimization (DTOS) provides the function to change the target topic dynamically by user's location.

3.4.2 System Configuration

Figure 3.12 shows the system configuration of the DTOS. The system stores the subscriber's information and performs the publishing process optimized. DTOS is implemented based on SDS, so the system has some standard functions with SDS. This section describes the DTOS's function, including the different points to the SDS.

First, it describes publishing the data from a publisher to subscribers via SDS and the IoT platform in the following.

1. A publisher sends the request that creates a topic to the DTOS.
2. DTOS creates the topic by itself, following the received request.
3. A subscriber sends the request that registers to subscribe to the DTOS.
4. DTOS creates the subscriber-topic, follows the request, and sends it to the subscriber.
5. A subscriber starts to subscribe to the IoT platform by ordering the subscriber-topic name.

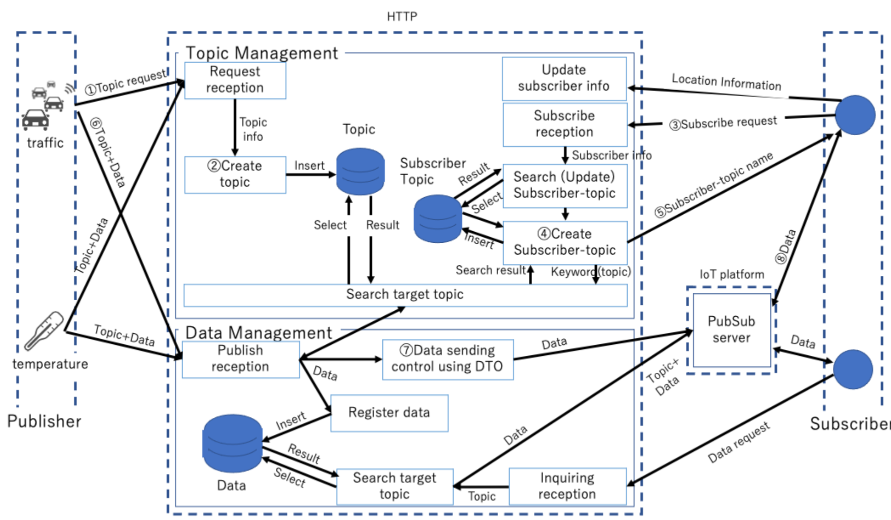


Figure 3.12: DTOS's system configuration

DTOS also performs among publishers, subscribers, and brokers. The system uses HTTP to connect publishers and subscribers. The difference with SDS is this system focuses on spatial information.

6. When the publisher sends the data to the DTOS, the DTOS confirms the topic name and the publisher name on the server.

7. DTOS sends data to the target subscriber-topic.

8. IoT platforms send data from the DTOS to subscribers.

For example, when the publisher created the topic including weather (he followed the naming convention as 3.2 section) and the subscriber subscribes to the topic including the same word, the DTOS confirmed matched the topic name and data's location and subscriber's detection range. The DTOS sends data to the subscriber when their information match.

Like this, the subscriber can receive what he wants without considering the dynamic topic changing by moving him when the data is sent via the DTOS.

Following the itemize shows the function to realize the process.

- Create and manage topics.
- Register to subscribe

- Update the subscriber's information.
- Publish data to IoT platforms
- Dynamic Topic Optimization

Last, the system uses an HTTP connection to connect to publishers and subscribers because the system uses the location information, so it requires a high-reliability protocol. In addition, the database uses MySQL.

The Definition and Naming Convention of Publisher's Topic

The Creating function of publishers' topics has the same process as SDS. DTOS defines the new naming convention to choose the target topic rightly.

- Topic name is expressed by collaborating words
- A last word of a topic uses the type of it.

Publishers define the topic name following the rule. The type indicated naming convention no.2 roles: matching words, publisher topics, and subscriber-topics. For instance, when a topic treats weather information in Fujisawa in Kanagawa, the topic name is expressed as kanagawa_fujisawa_weather. (In this paper, DTOS used _ (under bar) to concatenate characters) DTOS and subscribers can understand the topic's meaning by confirming the last word of the topic name.

However, the topic name alone does not tell us the effective range or center of the values indicated by the sensor. Then, DTOS adds the location information in the property of the SDS defined. Publishers send the information only when they create the topic, and it is stored in the database of DTOS. Table 3.3 shows properties and about Data-TTL described in the previous section, and detail explaining describes section 2.3.3.

Last, DTOS does not allow topic name duplication because the basic creating topic rule follows the SDS's rule. The relationship between the publisher and its topic is clear. In addition, when the topic has the same meaning on existed topic, the publisher adds his name to the topic.

Register to Subscribe

Table 3.4 shows the information to register to subscribe. Subscribers register not only target topic information but also their location information. A subscriber can update its

Table 3.3: Properties of Topic Creation on DTOS

Property	Explanation
Publisher	Publisher name
TopicName	Topic name
Type	Type of topic
Location	Target topic name
EffectiveRange	Subscriber location (Latitude, Longitude)
DataTTL (Optional)	Expiration time per data (unit: seconds)
Description (Optional)	Description about the topic

Table 3.4: Properties of subscriber-topic creation on DTOS

Property	Explanation
SubscriberName	Subscriber name
Purpose	The using purpose of a subscriber-topic
TopicList	The type name that a user wants(List format)
Location	Location information (Latitude, Longitude)
DetectionRange	The range is that a user wants to detect.
Direction (Optional)	Expressed by combining N(North), S(South), W(West), and E(East)

information to send it to DTOS; this endpoint is important because its location changes as the subscriber move.

Next, it describes purpose and topic on Table 3.4. The target topic changes according to the location when the information is in the physical world. Nevertheless, it is difficult for the subscriber to search and grasp all topics on the DTOS prior to subscribing, and it is also difficult to change the target whenever he updates his location. Therefore, the method is not realistic.

DTOS provides the function to change the topic automatically when the subscriber's location is updated. The function calls Dynamic Topic Optimization (DTOS). The topic name created by the publisher must follow the naming convention. Then, the last word of it indicates the type of information. When type treats the same information among multiple topics, it can express the common words among them. Therefore, the subscriber inputs the topic value, which is not a proprietary noun but an abstract word such as type. The topic has a list as shown in Table 3.4 because the number of topics the subscriber wants is not one. DTOS searches the target topic and target subscribers by using the last word of topics and location information each other, and it publishes data following the search result.

Then, DTOS made one regardless of the number of topics subscribers wanted to subscribe to, and subscribers can receive all data via the one. This study defines this only one topic as subscriber-topic. purpose value uses to determine the subscriber-topic name. Subscriber want to subscribe to some subscriber-topics. DTOS creates and manages some subscriber-topics for every subscriber in this case. Therefore, a subscriber-topic name is determined by collaborating subscriber name and purpose value.

SDS also defines subscriber-topic, but the role is different. SDS's subscriber-topic purposes of synchronizing data receive timing for some target topic. DTOS's purpose of manage some abstract target topic words.

However, the rule of subscriber-topic is no different. Therefore, the name does not allow duplicates, and created subscriber deletes his subscriber-topic only.

Last, low frequency updated data as the special sale event can miss sending data to the subscriber because of the frequency. Therefore, as shown in Figure 3.12, DTOS always stores the latest data on the database. The system sends it to the subscriber-topic when a subscriber registers to subscribe new subscriber-topic. Then, subscribers can use Data-TTL (section 2.3.3) to determine whether to use that data.

Dynamic Topic Optimization

DTOS realizes to change topics dynamically and automatically by using the subscriber location information and the subscriber-topic. Then, the subscriber fixes the subscribing topic while receiving data from the DTOS. The topic calls subscriber-topic. The subscriber can receive all data without changing the topic when he receives data via his subscriber-topic. All raw data are sent via the DTOS and the IoT platform, and all subscribers connect to the IoT platform, so they can receive data simultaneously when they subscribe to the same subscriber-topic.

DTOS controls publishing by using some information, the topic name, the location

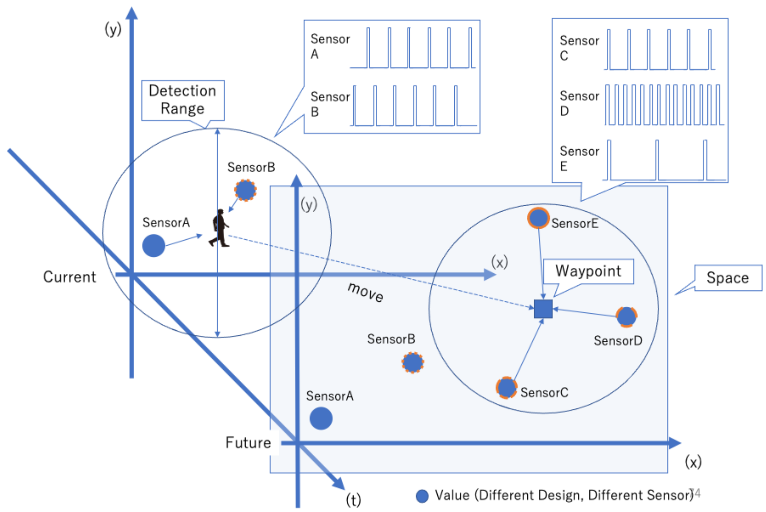


Figure 3.13: Overview of speculative data publishing

Speculative data publishing is realized by combining SDS and DTOS. This method can send several data synchronized and speculatively. Existing general map and navigation applications estimate the user's future position (waypoint), and this system uses the information.

of the published topic, topic value in the subscriber-topic, and the subscriber's location. DTOS checks this information organizes target subscriber-topics, and publishes data. Subscribers can receive appropriate data automatically without their operation by this method.

3.5 Realizing Speculative Data Publishing

A previous section described the collaboration of time and space information, and it is called spatio-temporal information processing in this study. This section describes implements of the methods that publish data speculatively. Figure 3.13 shows the overview of speculative data publishing.

3.5.1 Requirement

At this time, the method assumes the use of navigation applications. There are various Map applications and more as many navigation applications worldwide currently.

Users can index the destination, and the application suggests a route to it for the user.

When it realizes speculative data publishing, it needs to predict the user's location in the future. However, the information changes continuously. If it predicts the location every sec, the amount of location data is large, and the network cost increases to send them to the server.

From the above, this method uses the information suggested by the navigation application to reduce communication costs from the user to the server by realizing speculative data delivery. Therefore, it assumes it can previously receive waypoint information from the navigation application.

3.5.2 Format to Create and Publish

When a publisher creates a topic, he must set some parameters as the topic information. The basic format follows SDS and DTOS, as shown in Table 3.1 and 3.3. The rule of topic creation also follows them.

Data-TTL is an important parameter because the system uses the value when it sends data to subscribers. The system does not send data immediately when it receives data from the publisher. When to send data depends on the subscriber-topic's parameter (Receive Frequency). The system leaves the decision to the subscriber to choose whether or not to use it; then, the Data-TTL value is important. The system sends data, including published time and the Data-TTL. Subscribers can judge choose whether or not to use it using those values.

3.5.3 Format to Subscribe

This section does not indicate the system configuration because the system is implemented by collaborating SDS and DTOS. Therefore, the almost main parts are familiar to them. A main different point is the format of location information. The location information of DTOS and SDS currently indicates the user's location only.

However, at this time, the system needs to know the user's location in the future. Then, the system defines the extended location format. Figure 3.14 shows it. The new format can analyze waypoint information.

In addition, the method uses the periodic delivery method from SDS to realize speculative data delivery. Therefore, the format is constructed by all previous formats. The complete format is shown in Figure 3.15.

```
{
  "Location": {
   "Latitude": 56.7869214,
   "Longitude": 134.19824,
   "MovingInformationList" : [
    {
      "Waypoint": {
        "Latlng": "56.7869214,134.19824",
        "Distance": 20,
        "Duration": 5
      }
    }, …
  ]
 },
```

Figure 3.14: Extended Location is to realize speculative data publishing

It can express waypoints. Their information are suggested by general navigation application.

```
{ "SubscriberName": "TestUser",
   "Purpose": "Test",
   "TopicList": ["sensor", "temperature"],
   "ControlMode": "Periodic_Dynamic",
   "Location": {
      "Latitude": 56.7869214,
      "Longitude": 134.19824,
      "MovingInformationList" : [
         { "Waypoint": {
            "Latlng": "56.7869214,134.19824",
            "Distance": 20,
            "Duration": 5 }
            },
         ]
      },
   "ReceiveFrequency": "1 *",
   "DetectionRange": 10.0
 }'
```

Figure 3.15: Example of complete format to subscribe when using speculative data publishing

Subscribers must set all information as shown in the above image. The information includes using all elements of SDS and DTOS both; Hence, The format indicates an example of spatio-temporal information.

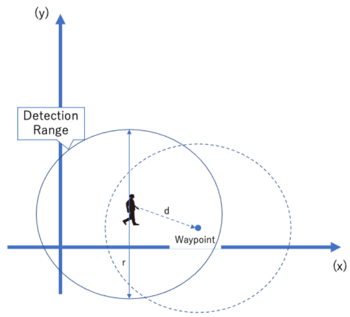


Figure 3.16: A waypoint is within detection range

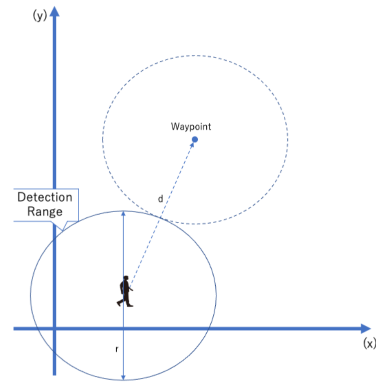


Figure 3.17: Two detection range are touching

3.5.4 Detection Method: Automatic Topic Selection

Speculative data delivery is implemented using the format in the previous section. This study defined speculative data delivery as that the subscriber can previously receive data from the current point and several waypoints. In addition, it delivers data periodically and optimally by calculating elapsed time and duration time of each waypoint.

The type of clustering can be separated into three by distance among waypoints. The image of each type is shown in Figure 3.16, 3.17, and 3.18. First, when it is the case of Figure 3.16, a waypoint is within the detection range of the current point (r), so their detection range is almost duplication. It means the distance between the two points (d) is shorter than the (r). It may detect the same target topic in each detection range. Then, the system removes the duplicated topic after detecting the target topic at each point. It is the same case when the waypoint is not in the detection range and the distance between two points is shorter than the two times detection range. This case can indicate ($r > d$).

The second is the case of Figure 3.17. A waypoint is not in the detection range of the current point, and the detection range of two points is just touching each other. In this case, each detection range does not duplicate perfectly. Therefore, the system detects all target topics at each point. This case can indicate ($r = d$).

The last is the case of Figure 3.18. A waypoint is not within the detection range of the current point, and the distance between two points is longer than the detection

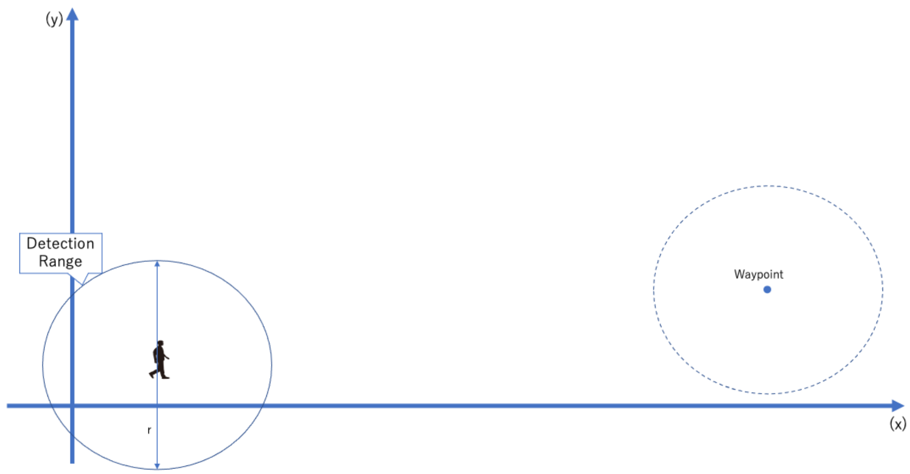


Figure 3.18: Two points are far

range. In this case, the system may miss detecting the target topic when it detects only each point because it would create areas in which no detection is made. The method generates a temporary point to detect all areas, as shown in Figure 3.19. The point and the detection range define the temporary point. First, the system calculates the azimuth and distance of the current point and the waypoint. Next, it calculates the temporary point by the azimuth and the detection range. It changes the current point to the calculated temporary point and repeats the process until the waypoint is exceeded. As a result, the system can detect all areas between the two points. This case can indicate ($r < d$).

3.5.5 Speculative Data Publish Method

The method depends on the SDS's function. Therefore, the basic format and data sending processing is the same.

However, the different point is that the subscriber's detection range changes. The image of relative spatio-temporal information of this case is shown in Figure 3.13.

The first detection result includes the no target topics when the subscriber moves because the target detection area changes by changing the subscriber's location. When the subscriber moves to the waypoint, he sends a request to update his location every time; in this case, the network cost becomes large, and it is not taking full advantage of

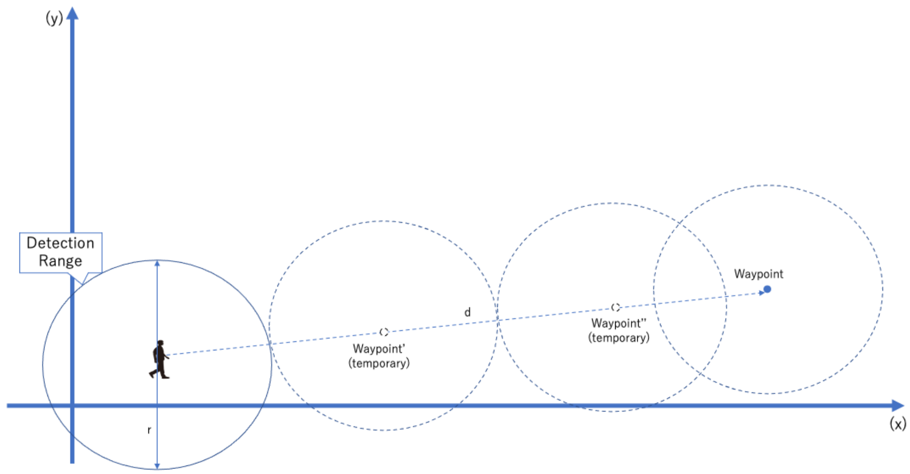


Figure 3.19: Automatic topic selection

When two points are far, the system may miss detecting the target topic when it detects only each point. Automatic topic selection is used to prevent it. The method generates a temporary point to detect entire areas, and data is detected at all points.

the waypoint information.

Therefore, the system uses the duration included in the waypoint. Duration indicates the expected required time to reach the waypoint. The system records the timestamp whenever the subscriber registers on it. It calculates the elapsed time by the timestamp and current time. After that, it compares the result and the duration. The system repeats the comparison until the sum of durations exceeds the elapsed time if the target has some waypoints. Then, the system removes the waypoint. Its duration is shorter than the elapsed time. It can optimize the target topic list for the subscriber's location without updating their information.

In addition, as shown in Figure 3.13, the system has the function of synchronizing the publish timing among some target topics. Therefore, the subscriber does not need to consider the cache or buffer to synchronize the unspecified topics and the subscriber-topic roles.

Table 3.5: Properties of subscriber-topic creation to do speculative data publishing Properties are constructed by combining SDS and DTOS, and location information must add waypoint information (Latlng, Distance and Duration).

Property	Explanation
SubscriberName	Subscriber name
Purpose	The using purpose of a subscriber topic
TopicList	The type name that a user wants(List format)
Location	Location information (Latitude, Longitude) and Waypoint information (Latlng, Distance, Duration)
ReceiveFrequency	Express receive freqency [minutes hours]
DetectionRange	The range is that a user wants to detect.
Direction (Optional)	Expressed by combining N(North), S(South), W(West), and E(East)

3.6 Spatio-Temporal Information Abstraction

This study, presented in the previous chapters, defined each format to perform the systems, SDS, and DTOS. This work defines the standard format for spatio-temporal information processing using their format. Table 3.2 and table 3.4 show each format. Unique keys are only one, ReceiveFrequency in SDS, and DetectionRange in DTOS, and they have almost the same key. Therefore, it is easy to merge their format and Table 3.5 shows the result. The table shows the format to realize speculative data publishing. In addition, contents in TopicList and Location information can extend. For example, the contents of the TopicList change from SDS to DTOS. Then, the subscriber can periodically receive some data near him. Table 3.5 shows the necessary properties to do speculative data publishing. This format does not include new properties. The change point adds additional information to the location property. Therefore, complex control also can implement quickly by following the format suggested in this study.

On the other hand, the contents of the TopicList change from DTOS to SDS. Then, the subscriber can immediately receive data from specific topics. Subscribers can receive data from some topics flexibly by the defined format.

Last, this design can implement some types of data delivery Currently. The following is a description.

Periodic: This delivery method is implemented SDS. It uses time elements, and subscribers order the specific topics in a subscriber-topic. The system checks targets and publishes data to them periodically.

Dynamic: This delivery method is implemented DTOS. It uses spatial elements, and subscribers order the abstract keywords in a subscriber-topic. The system checks targets using location information and keywords when the system receives data from publishers.

Aggregation(Time, Spatial, Hybrid): This delivery method has been realized by extending SDS. It uses time elements and Data-TTR. The system calculates and aggregates stored data periodically. Subscribers receive the calculated result, so they do not process the received data.

Like this, the system can realize aggregation by spatial elements because necessary elements and the subscriber-topic already have been defined on DTOS. Therefore, the system can also provide spatial and hybrid (time and spatial) aggregation.

Speculative: This delivery method has been realized by SDS and DTOS. It uses both time and spatial elements, and subscribers order the abstract keywords in a subscriber-topic. In addition, the system uses the subscriber information, current position, destination, and waypoints. The system checks targets periodically, and it calculates the estimation of subscribers' current position. It publishes data by the result without updating subscribers' information by themselves.

3.7 Discussion

It indicated the developed systems in this study. SDS and DTOS are basic systems for spatio-temporal abstraction. SDS described the usage of time elements. The system provides the two new primary a function and a definition. It is periodic data publishing and target topics abstraction.

Time has been used to record when data happen. SDS uses the time to control data delivery for unspecified users. It defined the necessary format to implement it and confirmed the method's effectiveness.

In addition, this study found the publish interval async problem in the process. It defined the abstraction topic, subscriber-topic, to solve the problem. Subscriber-topic has been able to multiple topics management. These generated a new function to calculate various data.

Next, DTOS described the usage of spatial elements. This system provides a bridge between clients and brokers and the database. It defined the necessary format to implement it as well as SDS. It was considered the independent system from the SDS when SDS developed because they have common parts, but the providing function is different. As a result, DTOS could provide a function focused on using spatial information. The system indicates the usage of spatial information and the common format of SDS.

At this time, it suggested data aggregation by extending SDS. The approach is the simple use of defined time elements. In addition, it uses all definitions to implement the approach. It provides various data clustering by time elements for subscribers. Subscribers will develop applications without time problems because SDS solve them.

However, in this case, it determined a value of Data-TTR simply. It can not consider the case that a subscriber wants to aggregate data older than receive frequency. In this case, SDS needs to manage a large amount of data effectively.

In addition, this approach can combine the location information also. It will describe the system using space elements in the next chapter. Therefore, it will describe the possibility what combining them in the last chapter.

3.8 Summary

The difference between the value of each sensor causes the independent receiving problem. Subscribers need to consider the problem when using data combing over two sensors. This study defined subscriber-topic to resolve this problem. A subscriber-topic manages some topics as one object. In addition, this study implemented periodic publish control. Subscribers can receive data on some topics from them.

However, the data has freshness. The time diverges between generated and published to subscribers. SDS provides the data's Data-TTL when it publishes. Subscribers can determine whether use it or not by the value. Last, the study suggested data aggregation. It is implemented by expanding the SDS.

Besides, this study implemented a dynamic topic optimization system (DTOS). DTOS provides abstract topic specifications for subscribers. This approach uses subscribers' positions and keywords that they are interested. The system searches the stored topic on the server by them. Then, it uses sensors' Data-ATL to determine whether send or not send to the subscriber.

This chapter described the definition of space elements to extend IoT platforms from the above and indicated the system's effectiveness in using defined elements to control publishing data.

In addition, it indicates two major applications of extended time and spatial information. First, it is data aggregation using time information. This study developed it by extended SDS. SDS can control data delivery by time, but only sent data is the latest. Nevertheless, the data can generate new values by combinate with other data. SDS realized data aggregation what a little changing data format. The TopicList element can have procedures and variable lists. The system implemented new functions to calculate the following procedures, but users do not must consider the complex things because the base data format does not change.

This study developed speculative data publishing by combining SDS and DTOS. The method uses all elements defined in this study. The system's changed point is a little. The Location element can have information on waypoints. General map applications generate information when users use a function to search routes to a destination. Therefore, users must not consider complex things when they want to use speculative data publishing.

These two major applications indicated the feasibility of spatio-temporal information abstraction. This study realized various spatio-temporal information processing using a common data format. It can express spatio-temporal, and users and developers can do the various data processing combination by changing used elements in the data format.

Chapter 4

Evaluations of Spatio-Temporal Data Delivery

This chapter describes the evaluation of spatio-temporal data delivery. The developed systems in this study perform among clients and IoT platforms. Hence, the systems' performance effect data delivery. This study evaluates the performance of the primary systems of spatio-temporal information processing. Therefore, it evaluates the performance of SDS and DTOS. Moreover, it describes the current limitation of using the systems.

4.1 Experimental Conditions

Table 4.1 shows the experimental conditions and Table 4.2 shows the virtual environment. It uses the virtual environment to measure the system load and processing time. Table 4.3 shows the parameters of AMQP and MQTT. The evaluation does not consider the network effect because subscribers, publishers, and the IoT platform have been installed on the computer.

4.2 Evalueations of Time Element Processing

This study evaluated SDS by comparing the network cost of the general IoT platforms. It measured the data traffic of three protocols based on pub/sub-model, XMPP, AMQP, and MQTT. Next, it measured the system load and the processing time when the number of connections and publish frequency increased.

Table 4.1: Experimental environment (Host machine)

OS	macOS Catalina (v10.15.7)
CPU	2.4 GHz 8 Core Intel Core i9
Memory	64 GB 2667 MHz DDR4
Langage	python3 (v3.9.2)
IoT platform	RabbitMQ(v3.8.8) [92]

Table 4.2: Experimental environment (Virtual machine)

OS	Ubuntu 18.04 (64bit)
CPU	1 Core
Memory	8192 MB
IoT platform	RabbitMQ

Table 4.3: Paramters of AMQP and MQTT

Protocol	Parameter	Value
AMQP	Heartbeat(seconds)	60
MQTT	QoS	0
MQTT	Keepalive(seconds)	60

This study focused on the data traffic connecting a subscriber to an IoT platform and the system. The reason is that the system is used to extend the IoT platform's control function, and the system connects to one IoT platform. The evaluation did not measure the data traffic during the system and the IoT platform because it assumes they have been installed in the same location. (e.g., management by the municipality of the city) Then, the network cost during them does not affect subscribers. In addition, a publisher uses HTTP when he sends data to the system. The data traffic's size depends on the content's size, and it is the same meaning as the general HTTP connection.

However, the data traffic among subscribers and the system depends on the data size, the number of subscriber-topics, connection to the system, and the IoT platform. This evaluation focused on the network cost of subscribers.

SDS and the IoT platform have installed the same machine because this case becomes the experimental environment simply. It uses SOXFire [93] as an IoT platform based on XMPP and RabbitMQ[92] as an IoT platform based on MQTT and AMQP.

4.2.1 Data Traffic to Connect

First, it measured the data traffic until the subscriber can become to receive data. It measured the packets while clients sent requests to each platform ten times. Then, a monitor tool for the network captures the data traffic and calculates the sum of each trial. All protocol needs the most data traffic until the connection to the IoT platform, and a subscriber is complete. They send a ping to confirm whether the subscriber lives

Table 4.4: Data traffic to connect

Topics	AMQP	MQTT	SOXFire	SDS
1	2170	346	16347	1831
2	2551	522	16347	1831
3	2932	698	16347	1831

Table 4.5: Amount of data traffic when sending data
This value have been calculated by adding lengths of body and header. (unit is byte)

Topics	AMQP	MQTT	SOXFire	SDS(AMQP)	SDS(MQTT)
1	256	195	585	261	195
2	512	390	1170	361	295
3	768	585	1755	461	395

or not after creating a connection. Therefore, it measured the data traffic when the subscriber started to connect to the platform. Table 4.4 shows the result. When using SDS, it recorded the data traffic when the subscriber connected by HTTP because he connects to it only when he registers to subscribe. In addition, the size of data traffic depends on the protocol of using the IoT platform because the subscriber receives data from it. This study used RabbitMQ as an IoT platform based on MQTT or AMQP, so the subscriber connected to RabbitMQ when it measured its data traffic. In addition, it found that if the publisher does not send data within the ordered time, the IoT platform sends data to check the connection with the subscriber. (64 bytes in MQTT, 156byte in AMQP)

4.2.2 Comparing The Data Traffic by The Number of Topics

It used the data with the same length and format shown in Figure 4.1 to experiment and measure the data traffic. Then, it measured the raw data part, not including the metadata part, because metadata is sent only when the publisher creates a topic. Besides, SOXFire uses XML format.

To measure used Wireshark, which recorded the traffic until the subscriber received data that the publisher sent. It calculated the data traffic by removing and using data to connect to the subscriber and the IoT platform. It did do the measurement ten times to confirm the data traffic correctly. Table 4.5 shows the result.

MQTT and AMQP do not generate waste data traffic because they can subscribe

JSON（Using normal topic）：

{"topic": "1min_publish", "elements": {"value": "10”}, "publish_timestamp": "2021-01-21 13:58:42"}

JSON（Using subscriber-topic）：

[{"topic": "1min_publish", "elements": {"value": "10"}, "publish_timestamp": "2021-01-18 23:50:33"}, {"topic": "2min_publish", "elements": {"value": "20"}}]

Figure 4.1: Evaluation data sample

multi topics in one connection. Nevertheless, the header is added to each data, so the more topics were added, the more data traffic was. On the other hand, the data traffic can reduce by merging some data into one object. The reason is that the data increase body part only in this case.

Therefore, if the system sends data from more than two topics, the suggested method can reduce 20 % - 30 % of data traffic. In addition, the effect become increases with the number of topics.

4.2.3 Data Traffic Considering The Synchronized Received

The previous sections indicated the result of the cost by connection and data size. This section evaluated the data traffic considering the synchronization received from several topics on AMQP and MQTT. XMPP removed the evaluation because its data traffic is too extensive than other protocols. This evaluation has been calculated by multiplicating the size shown in Table 4.6 and the number of the data. It assumed the data size was the same.

Table 4.6 shows the result. The number of data was calculated by collaborating on topics with different publishing frequencies. It calculated the value by multiplicating their number and the data size shown in Table 4.6. Their common denominator defines the number of data. Therefore, the data traffic becomes large.

On the other hand, the number of data of SDS is not affected by the combination of

Table 4.6: Number of data generated per period

Number of data generated before one synchronization and amount of communication required for delivery when data with different transmission frequencies are acquired synchronously.

Topics	Frequency	Data	AMQP	MQTT	SDS + AMQP	SDS + MQTT
2	1,2	3	1024	564	361	295
3	1,2,3	11	2816	2068	461	395
3	1,2,5	17	4352	3196		
4	1,2,5,10	18	4608	3384	561	495
4	1,2,3,10	58	14848	10904		

Table 4.7: Rate of decreasing using subscriber-topic

Percentage reduction in amount of data transmission per protocol when data is sent without synchronization and when reception timing is synchronized.

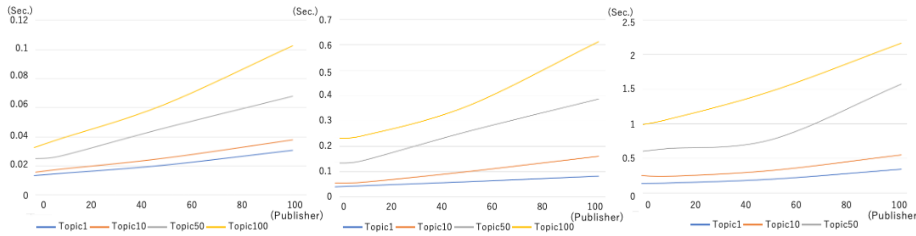
Topics	Data	Rate of Decrease(%) AMQP	Rate of Decrease(%) MQTT
2	4	64.7	62.1
3	11	83.6	81.5
3	17	89.4	87.6
4	18	87.8	85.4
4	58	96.2	95.6

publishing frequency. SDS synchronizes the data of some topics to one object, and it sends the object as data to the subscriber. Therefore, this approach can greatly reduce data traffic. Table 4.7 shows the result. From Table 4.7, the approach reduced 70 % of data traffic, including the data traffic of HTTP to subscribe, compared to AMQP and MQTT.

4.2.4 Delay by Using SDS

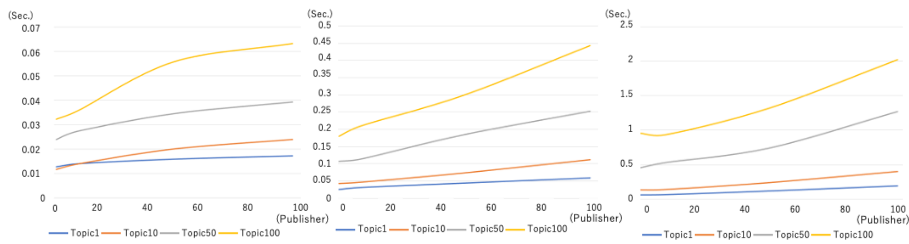
SDS searches all subscriber-topics registered on it to control sending data. It confirms the matching of the subscriber's period, creating a list of sending data from the subscriber-topic. Therefore, this process needs time when SDS sends data to subscribers more than general IoT platforms.

This experiment measured the delay time by the number of subscribers, topics including a subscriber-topic, and the number of publishers (SDS's load). The experiment did do on AMQP and MQTT. SDS and subscribers are created on the virtual machine



(a) The number of subscribers: 1 (b) The number of subscribers: 10 (c) The number of subscribers: 50

Figure 4.2: Time required from sending data through SDS to receiving data when using AMQP



(a) The number of subscribers: 1 (b) The number of subscribers: 10 (c) The number of subscribers: 50

Figure 4.3: Time required from sending data through SDS to receiving data when using MQTT

shown in Table 4.2, and it does not consider the effect of networks and other processes. It measured ten times, and it calculated the average.

Figure 4.2 shows the result of using AMQP, and Figure 4.3 shows using MQTT. Compared to them, MQTT is better than AMQP in all results, but the difference was minimal. Besides, the delay time is over 1 second when 50 publishers have 50 topics over. However, a subscriber has topics over 50, which is rare because this means a subscriber-topic treats sensors over 50 simultaneously.

On the other hand, the delay time is less than 0.5 seconds when the number of topics is 10, whether the load is high or low. This delay time is short enough, and the method performs every minute. The result could confirm that SDS must not consider the delay time.

Table 4.8: Changing system load by increasing subscribers and publishers

		CPU(%)			Memory(%)		
Subscribers	Publishers	Average	Mode	Max	Average	Mode	Max
1	1	2	1.84	3	0.5	0.5	0.5
	10	14	13.01	18	0.5	0.5	0.5
	50	39	38.16	42	0.6	0.6	0.6
	100	40	39.3	43	0.6	0.6	0.6
10	1	2	1.79	3	0.5	0.5	0.5
	10	13	12.75	16	0.5	0.5	0.5
	50	39	37.79	41	0.6	0.6	0.6
	100	39	38.38	42	0.6	0.6	0.6

Table 4.9: System load when transmitting at 5-second intervals

	CPU(%)		
Publishers	Average	Mode	Max
100	14	13.01	28

4.2.5 Loads of The Number of Connections and Publish Frequency

This approach generates the system load by publishing data via SDS. This evaluation measured the change in system load with the number of publishers and publishing frequency. The experiment uses the virtual machine shown in Table 4.2, and publishers send data from the host machine. It recorded the performance of the CPU and the memory.

First, it created the environment where publishers publish data on publish frequency of 0.5 seconds as a high load situation. Each publisher published 100 times while dumping CPU and memory usage every second. It repeated this process ten times to calculate the average, the mode, and the max. Table 4.8 shows the result. The load also increases when the number of publishers increases, while the number of subscribers does not affect the load. In addition, it did not confirm the increase in memory usage because SDS only stores the latest data.

Next, it created the environment where publishers publish data on publish frequency of 5 sec as a low load situation. Then, each publisher randomly has a wait time of 1-3 seconds to disperse the publishing. Table 4.9 shows the result. Memory usage does not change the high load situation. CPU usage is less than half of the high load situation. Therefore, it can confirm that it was a minor of the system load.

4.3 Evaluations of Spatial Elements Processing

This section describes the evaluation that evaluated the effectiveness by comparing dynamic topic optimization data traffic and the general method. Besides, the processing cost increases when it uses this approach. Then, this evaluation measured the CPU and memory usage to evaluate the system load and confirmed the delay by the approach. The experimental condition assumed the same as SDS. Therefore, this section does not indicate the experimental condition.

4.3.1 Data Traffic by Changing Topics

Subscribers must subscribe to the target topic before receiving the data using the IoT platform based on the pub/sub-model. When a subscriber moves, the target topics also change. Therefore, subscribers must change topics according to their position. Nevertheless, subscribers must subscribe in advance, but the network cost and wasted data increase when they subscribe to all topics they want. Then, if only topics around the current location can be subscribed to as the subscriber moves, unnecessary data communication can be prevented.

First, it measured the data traffic to change topics using MQTT and AMQP. Figure 4.4 shows the result. A subscriber changed topics manually, using MQTT and AMQP. AMQP happened the rebuilding connection because it must cut off the connection to unsubscribe. MQTT can change topics with the same connection because it can unsubscribe per topic. In addition, MQTT can transfer multi requests to one request. Therefore, the result has calculated the greatest case.

The data traffic of MQTT is lesser than AMQP from figure 4.4. The minimum value is 266 bytes when the subscriber changes one topic to another using MQTT. Then, the requests and response data traffic to subscribe is 172 bytes, and the requests and response data traffic to unsubscribe is 94 bytes. It is 456 bytes when the subscriber changes two topics to the other two.

The data traffic does not change by the number of topics; it depends on the subscriber's name and position length. The data traffic does not change by the number of topics, and it depends on the length of the subscriber's name and position. It needs 296 bytes when the subscriber sends a request with 100 bytes, including the subscriber's name and position.

Therefore, this approach is effective when the changing target topics are two more.

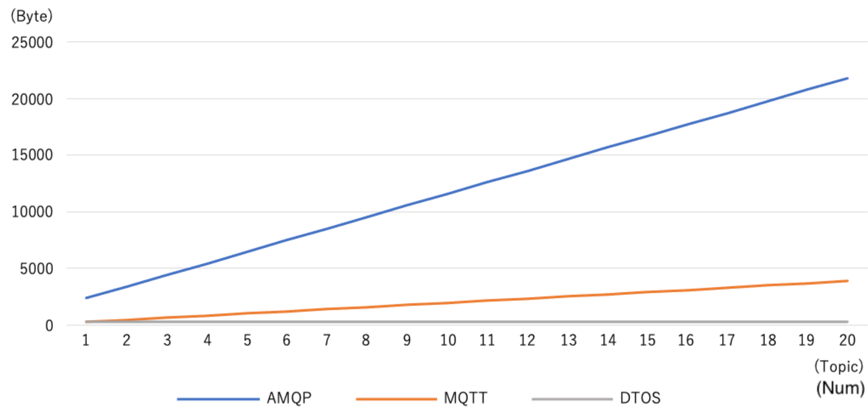


Figure 4.4: Changing amount of data traffic by increasing subscribers

In addition, the bytes were reduced by 70 % in MQTT and 90 % in AMQP when the subscriber wanted to change five topics to the other five topics.

4.3.2 Loads of The Number of Subscriber-Topics and Publish Frequency

Next, it evaluated the system load. This approach happens to process costs. The process is that it sends data to target subscribers. Then, the system load depends on the number of subscriber-topics and published data.

At this time, the experiments created two situations. There were the high load situation and the low load situation. They are the same situations at SDS's evaluation. It measured the change in system load by increasing the number of subscriber-topics in each situation. The measured values were calculated by dumping CPU and memory usage during 100 publishing per publisher.

First, memory usage did not increase. The reason is that the system does not store data because it sends data immediately. Therefore, it confirmed that this approach does not have a large load on memory usage.

Next, it confirmed CPU usage at high and low loads. Figure 4.5 shows the result. It found the increased system load by the number of subscriber-topics, which does not depend on the publishing frequency. In addition, in the case of high publish frequency,

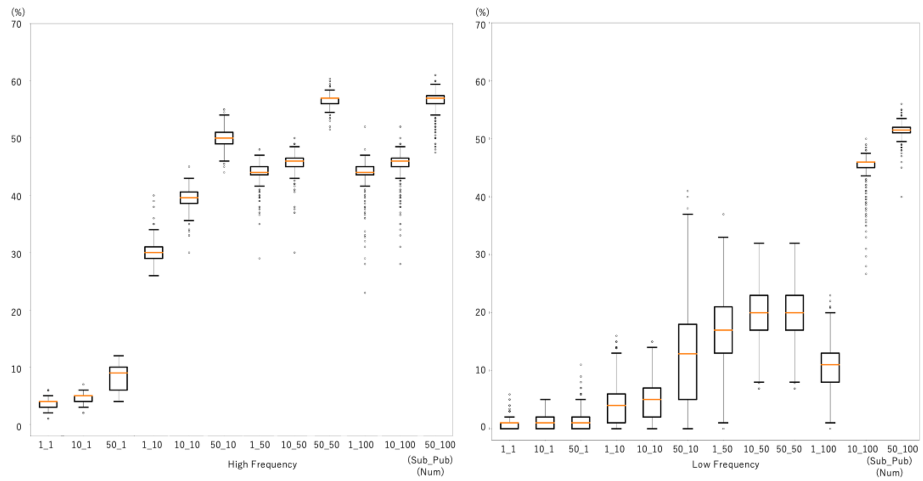


Figure 4.5: System load of DTOS in two situations

Variation of CPU occupancy by number of publisher and subscriber-topics in high and low frequency situations.

the system load increased by the number of publishers, while in the case of low publishing frequency, the system load increased gently.

4.3.3 Delay of The Number of Subscriber-Topics and Publish Frequency

DTOS publishes data immediately when a publisher sends it. Then, the system publishes data to them after it searches for target subscriber-topics. The process becomes complex by the number of subscriber-topics. It delays when the process needs a long time. In addition, the system load depends on the number of publishers, which indicates the possibility that the high system load causes the delay. Therefore, this evaluation measured the delay by the system load. It measured the same experiments in the previous section and recorded the time from when a request arrived at DTOS until it was published. The result shows in Figure 4.6.

It confirmed the processing time increase by the number of publishers and subscriber-topics. Besides, there was a large difference in processing time for the same number of publishers and subscriber-topics in environments with different publish frequencies.

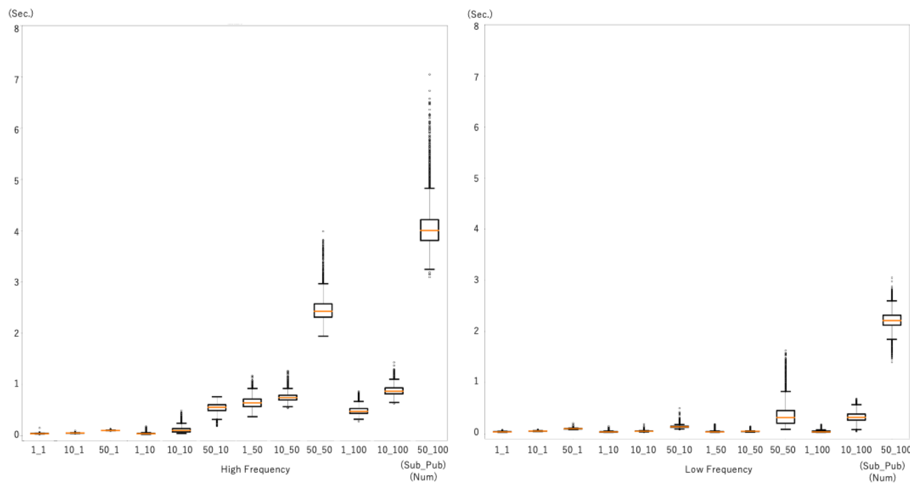


Figure 4.6: Processing time of DTOS in two situations

Variation of the processing time by number of publisher and subscriber-topics in high and low frequency situations.

The average processing time is 3.6 seconds, and the max is 7seconds, while the average processing time is 2 seconds, and the max is 4 seconds in low publish frequency. Therefore, About the processing time depends on the publishing frequency.

4.3.4 Loads and Delay by The Number of Elements in A Subscriber-Topic

A subscriber-topic can include some keywords of topics in DTOS. The process becomes complex by the number of elements of a subscriber-topic. This evaluation measured the system load and the processing time when a subscriber-topic increased the number of elements. (The elements are in the topic list.) The elements do not make so many because the subscriber does not order specific topic names at DTOS. For example, fujisawa_weather and yokohama_weather are summarized as weather. When a subscriber wants environmental information in a wide area, the kind name of the environment is about six, the weather, temperature, wind speed, humidity, rainy percentage, and illuminance.

At this time, the upper limit number of elements of a subscriber-topic is set at 20.

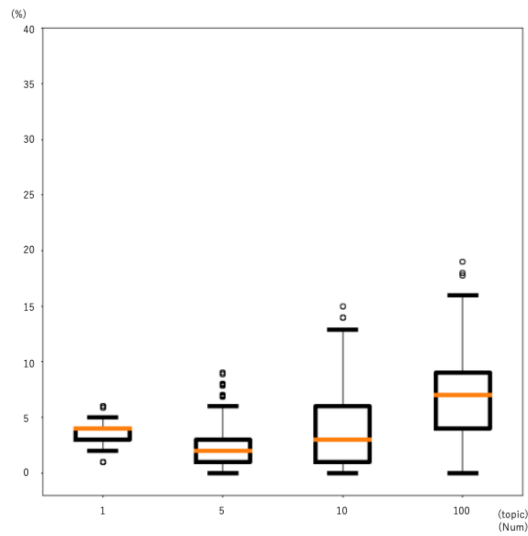


Figure 4.7: Changing amount o data trafic by the number of elements in a topic

It measured the system load when one, five, ten, and twenty elements. The number of subscriber-topics was fixed to one only to measure changing the system load by the elements, and the experimental condition used low publish frequency. Figure 4.7 shows the result of CPU and memory usage.

First, from Figure 4.7 , CPU usage increased when the number of elements increased. The average is less than 10 % when it sets 20 elements. It confirmed 20 % at max because of many cases by duplicating the published timing. Next, the processing time also increased when the number of elements increased. Figure 4.8 shows the result. It is less than 0.05 sec when the element is one, which is less than 0.15 sec when it is 20.

Therefore, it found that DTOS does not have a high load of CPU usage and processing time.

4.4 Discussion

This approach can reduce network costs by realizing synchronized receiving. In addition, it considers another approach that does not always connect to the client and the server because the synchronized receiving reduces the data receiving frequency. When an application uses the IoT platform of pub/sub-model, it needs to subscribe/unsubscribe

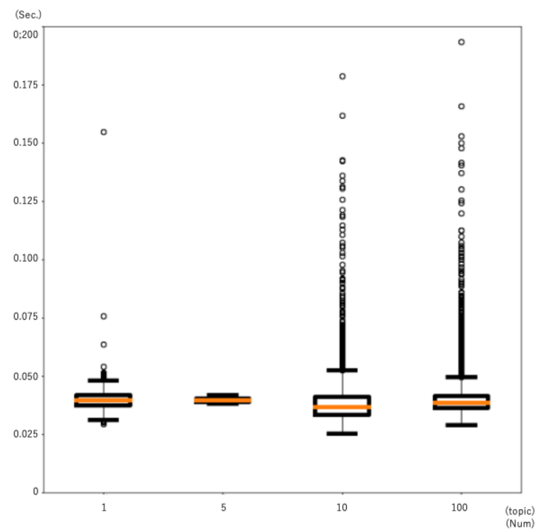


Figure 4.8: Changing the processing time by the number of elements in a topic

to inquire about its data. Nevertheless, SDS is implemented by HTTP, which can provide the functionality to inquire about data without a subscription.

It implemented the function and measured the data traffic when it received data from four synchronized topics. The amount of data traffic was 1,495 bytes.

From the result, the function is effective when a subscriber wants to receive data from one topic, or the receiving frequency is low. However, SDS cannot publish when it uses HTTP to send data to the subscriber. Therefore, it was able to show that publishing methods can be more effective if they are used in different ways depending on the number of target applications and their uses.

Next, it discusses spatial information processing. The evaluation found that CPU usage depends on the number of subscriber-topics and publishers. At this time, the evaluation made publishers send at 0.5 and 5-sec intervals. It is hard to imagine that all publishers send the intervals because they are high frequency in the real world. Therefore, the CPU usage will be reduced better than the evaluation. Nevertheless, the CPU usage increases drastically when so many publishers send data intensively. It should be designed to be load-balanced, as it places a heavy burden on the machine that performs DTOS.

Next, it found that the processing time also depends on them. It will need to con-

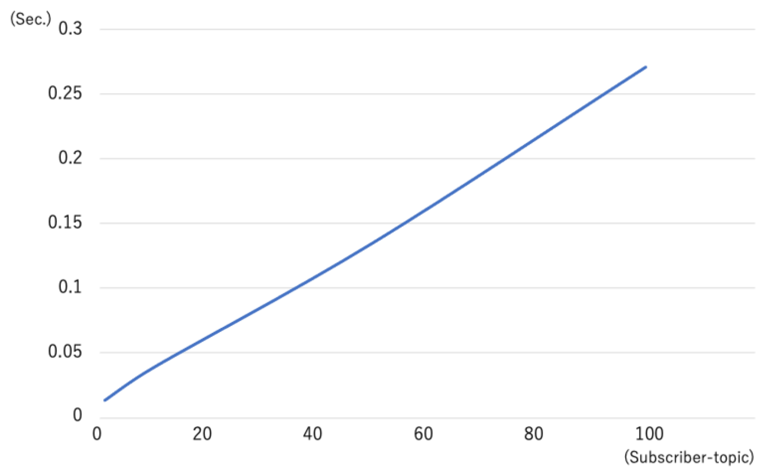


Figure 4.9: Increased processing time of SDS by number of subscriber-topics

This figure shows the processing time increases linearly by the number of subscriber-topics.

sider the number of subscriber-topics in the real world because it will increase drastically. It is important how to reduce it on the server. The topic-sharing method described at section 5.2.2 is effective in achieving it.

4.4.1 Increased processing time due to extended system for IoT platforms

The previous section described use case examples of this study. The scale required of the platform is different by the situation. Spatio-temporal information processing has been realized by two extended systems, SDS and DTOS. This section discusses per system. First, it indicates SDS. At SDS, the total processing time is important to publish all target subscribers. This reason is that SDS does not depend on publishers. SDS performs only set schedules. In this study, SDS checks the database every minute. Therefore, this system performs only every minute regardless of publication by publishers.

However, when the number of subscriber-topics increases, the total time to process all publication increases too. It indicates that the later the subscriber transmits, the more likely there will be a reception delay. Protocol choose AMQP because its load is

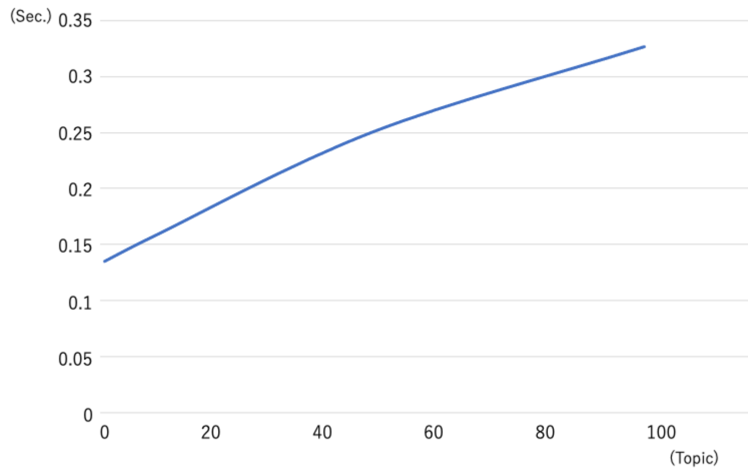


Figure 4.10: Increased processing time of SDS by number of topics

This figure shows the processing time increases linearly by the number of topics.

more significant than MQTT, so all MQTT's value is smaller than AMQP. It estimated the processing time per the number of subscriber-topics, the number of publishers, and the number of topics. The result has indicated an almost linear increase by per subscriber-topics, per publisher, and per topic. Figure 4.9 and figure 4.10 shows the example of increasing per subscriber-topic, and per topic. It has indicated an increase of 2.64 milliseconds per subscriber-topic, 0.16 milliseconds per publisher, and a rise of 0.21 milliseconds per topic. When SDS performs every minute, the limitation does not exist because each process time is short for SDS. The reason is that it indicated that the check cycle of SDS can become shorter than now. Therefore, SDS can control more precisely. However, this delay time is not zero, so the system must implement data delivery control by priority.

Second, it discusses DTOS. DTOS publishes data per receiving data from publishers. Therefore, the processing time and loads depend on receiving data per publisher. For SDS, only the total time is important; for DTOS, the processing time per publisher is also important. In addition, it publishes messages immediately when it receives data from publishers. Delay affects the system's spec directly. This study estimated process time using the evaluation's result. Moreover, it considers its limitation. It estimated the processing time per the number of subscriber-topics and publishers. When it has

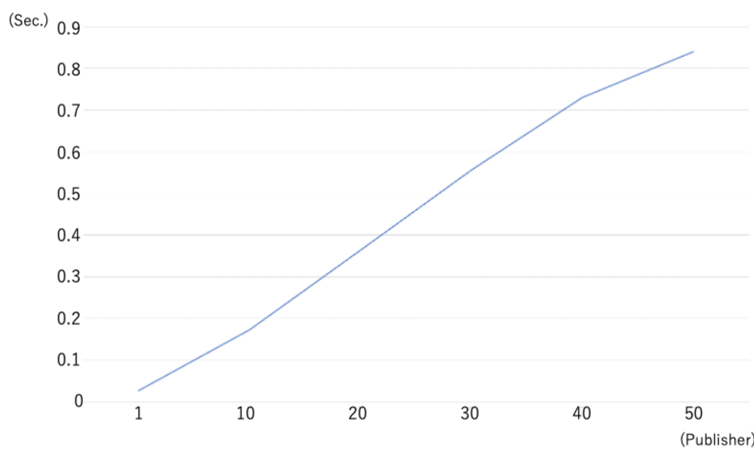


Figure 4.11: Increased processing time of DTOS by number of publishers

This figure shows the processing time increases linearly by the number of publishers.

evaluated DTOS, the virtual machine has prepared 3 core for DTOS, DB, and an IoT platform.

Figure 4.11 shows the increased average processing time for every ten publishers and a subscriber-topic. Figure 4.12 shows the increased average processing time for every ten subscriber-topics and a publisher.

They indicate that the processing time increases almost linearly. This study estimated the amount of increase in processing time.

The result indicated that when there were one subscriber-topic and a high load situation, an increase of 16.2 milliseconds per publisher. Moreover, the result indicated that when there were one publisher and a high load situation, an increase of 32.5 milliseconds per subscriber-topic.

In addition, figure 4.13 shows the result of increased load. The number of subscriber-topics has not affected the load. The value indicates almost 20 %. The number of publishers affects the load a little. However, it increased by almost 0.377 % for every 10 publishers. On the other hand, a load of an IoT platform increase easier than DTOS. Therefore, it has been found that it is better to run this system and IoT platforms on separate machines.

The result is almost the same when it is a low load situation. Therefore, it can be

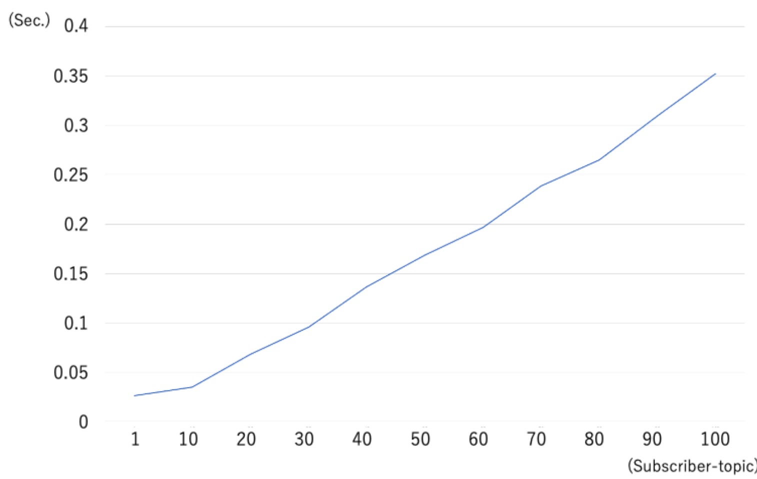


Figure 4.12: Increased processing time of DTOS by number of subscriber-topics

This figure shows the processing time increases linearly by the number of subscriber-topics.

claimed that these values are the limitation of current DTOS.

Finally, based on the above, specific limitations are shown. The processing time happens on 5.74 sec on average when there are 100 subscriber-topics and 50 publishers in this study. Besides, cars can drive within 60 km/h on ordinary roads. The car can move 62 meters every second at the max speed. Therefore, the subscriber misses the data if there is a target topic with an effective range within 350 meters and an updated frequency within 5.74 sec.

Nevertheless, this problem can solve by scaling out of the systems. The evaluations indicate the linear increasing processing time on both SDS and DTOS. It means that when subscriber-topics, publishers, and topics increase, the processing time increase too. The method is two mainly to prevent this problem. First, it is what improves the system's spec. In this study, the systems' spec is a minimum model when evaluated. Therefore, the entire processing time can be reduced by improving the systems' spec. It depends on CPU spec.

Second, it is what sets multiple units. Multiple units can be installed to distribute to clients. It reduces the number of topics, subscriber-topics, and publishers per unit. When SDS uses to publish data for visitors to events, the SDS is performed as a server.

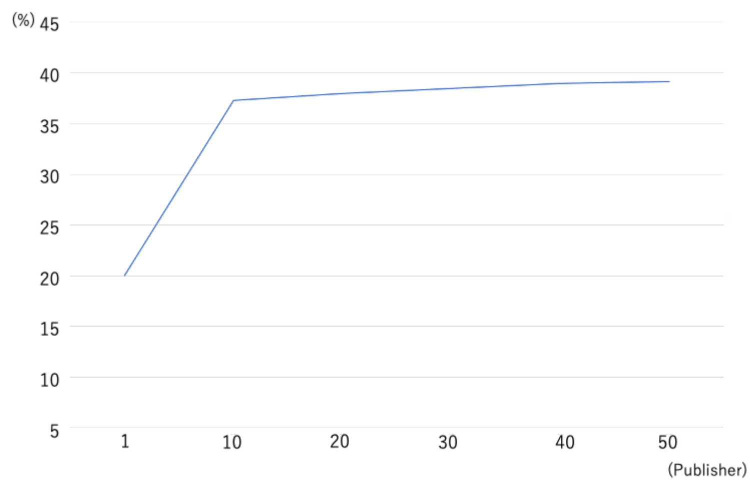


Figure 4.13: Increased load of DTOS by number of publishers

The server can handle 20,000 visitors (as subscribers-topics), and several publishers and topics. The number of topics and publishers becomes smaller than the number of subscriber-topics. Therefore, the system can ignore their effects. The result indicates that a unit of SDS can handle all subscribers in middle-scale events and some areas of the city (towns and villages).

If there are more than 20,000 visitors to an event, the organizer must install multiple SDS servers. The most simple case, the largest hall in Japan has a capacity of 100,000. That case must need five units of SDS. Like this, the indicator is effective for other situations. The number of publishers is always smaller than subscribers and subscriber-topics.

On the other hand, a DTOS can handle 2,000 subscriber-topics. This value was estimated as the number of processes that could be completed within one minute. It is because the DTOS is used by collaborating SDS to realize speculative data publishing. Moreover, in that case, the system does not publish when it receives data. It prevents the increasing processing load of the system. Hence, the system can handle subscribers of more than 2,000. From the results, a unit of the speculative data publishing system can handle subscriber-topics at least 2,000. This number can improve by increasing cores and units. In addition, this number indicates the limit when the number of clients can be controlled at once. Carpool services and help services in condominiums will have fewer clients to connect to the system to because users are dispersed by time and location. Therefore, this study considers the systems that can use for smart cities.

4.5 Summary

This chapter described evaluations of spatio-temporal information processing. The evaluation measured the performance of two systems because spatio-temporal information is developed by combining time and spatial information processing. The kind of process collaboration is some. Therefore, this study evaluated basic processes. They are periodic and synchronized data broadcast using time and dynamic topic optimization data broadcast using spatial information. Their results indicated that the performance was enough to install this system for cities. Besides, it indicated the system's limitations, but the system can avoid them because it can do the speculative data broadcast as described in section 3. Therefore, this study's methods were found effective for cities.

Chapter 5

What Will Spatio-Temporal Information Processing for City-Scale Sensor Data Delivery Lead in The Future?

This chapter closes the review and the future direction of this study. This guideline is useful when people consider the future figure of cities. First, the section described the review of this study. In addition, it describes the position of study in next-generation cities. The next-generation cities are called Connected Smart Cities.

The next section indicates the figure for Connected Smart Cities. After that, it discusses the technologies and must consider the requirements for realizing them. Last, it describes the conclusion.

5.1 Review and Position of This Study to Support Smart cities

This study has proceeded to make the city Smart. For that purpose, the first chapter systematically summarized various sensing technologies. Moreover, this study did practice Smart Sensing and the discussion.

After that, it described the importance of the data being clustered by anything when there is a more increasing amount of data. This study expressed data in three dimensions, time and spatial, to clustered data. The data that can be expressed in this way is called spatio-temporal information. It defined the elements that construct spatio-temporal information.

Next, this study indicated the developed systems using those definitions and necessary rules. Various spatio-temporal information processing is realized using the definitions. This study indicates some actual processing. First, SDS provides data delivery control using time. It prevents data overflow by managing users' requirements, and it realizes synchronized data delivery. Moreover, the extended SDS realized data aggregation using time. Almost data formats and functions did not change from SDS. Extended SDS can manage procedures and variable lists to calculate multiple data.

Second, DTOS provides data delivery control using spatial information. The system provides the function using spatial information. It realizes dynamic changes to target topics.

Last, this study realized speculative data delivery by combining SDS and DTOS. Speculative data delivery uses the elements defined in SDS and DTOS and moving information. The moving information is suggested by general map applications. The information includes characteristic points(waypoints) and duration. The system adds the information into spatial information. Moreover, the system controls data delivery periodically by time elements. Therefore, it realized speculative data delivery by spatio-temporal information processing.

In addition, it evaluated the systems' specs of SDS and DTOS. The results indicated the processing speed and the loads are no problem when the systems use for the use case examples described in section 2.

Each IoT platform uses a different protocol and structure, but the data format does not depend on the protocol's features. Therefore, the format suggested in this study can use all IoT platforms. It leads to the integration of data formats among several platforms. Multiple platforms can integrate seamlessly, an example of integration between SOXFire and Blockchain Marketplace. About the example describes below section. This study has enormous value for smart cities because it causes the activation of data use.

This study indicated the potential of spatio-temporal information abstraction. It described section 1.6.2, IoT platforms for cities treat sensitive people's data because it includes the privacy information. Therefore, platforms prefer to handle data in a city. However, when the city handles it simply, the engineers cannot express the relationship between the city and the data. It causes chaos in cyberspace because there probably

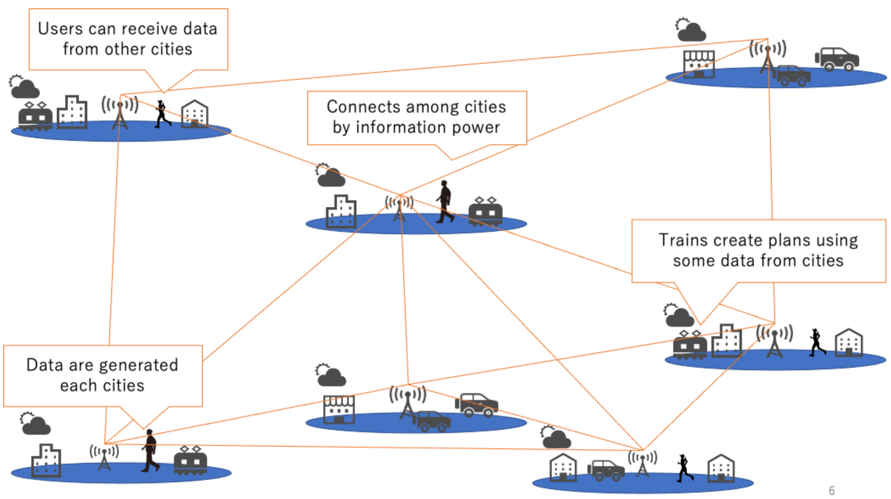


Figure 5.1: Overview of connected-smart-cities

Connected-smart-cities is constructed in collaboration with some smart cities. Cities are connected by information, and they exchange data with each other. Therefore, in society, users and applications can use data from the city lived them and other cities.

has irrelevant data in the city. To prevent this situation is critical to expressing the relationship with them.

This study realized the expression of spatio-temporal information processing. It has a large impact on the suitable placement of IoT Platforms. All IoT platforms can use definitions determined in this study. It will guide the integrated data expression and suggest the necessary information for the city's applications. When IoT platforms use the same data format, the visibility is high. Therefore, data senders can rightly judge where they send data. It will realize the well-maintained cyberspace. Then, engineers and researchers can consider the figure for next-generation cities. How, when, what, and why does a city collaborate with other cities. This study indicates the first step to considering next-generation cities. It indicates connected-smart-cities as one figure of next-generation cities. Connected-smart-cities means cities can connect in cyberspace. Figure 5.1 show the overview.

It simulates various city collaborations. Cities must collaborate to help each other when they are in trouble. For example, when disaster strikes the city, citizens' lives are at risk because the city's functions do not work. Then, other cities must help, providing food and clothes, rescue activities, and more. Connected-smart-cities support it. They can do monitoring and exchange the city's information and status easily by them expressed in cyberspace. Therefore, connected-smart-cities are one of the next-generation cities that should become.

5.2 Future Direction of Spatio-Temporal Information Processing and Sensing Data Delivery

In the future, increasing data all over the world. The data are used in various situations, including connected-smart-cities described in the previous section. This section discusses the future direction and the necessary technologies that focus on city information.

This study suggested the base for expressing city data in cyberspace by defining spatio-temporal information. It has made it possible for suitable IoT platform placement in the city. The next step is connecting among cities. It discusses tasks and the necessary technologies to solve them.

5.2.1 Spatio-Temporal Information Processing with An Understanding of Hierarchies

The place where people live is expressed by combining area names. People call it an address. The largest area of an address indicates the country. The country includes many areas, such as prefectures in Japan, states in the U.S., Etc. Prefectures and states also include many small areas. The accumulation of multiple areas represents an address. For example, the Keio University Shonan Fujisawa Campus address is Endo 5322, Fujisawa city, Kanagawa prefecture, in Japan. The address can express the combining multiple layers in cyberspace. Each layer has a different name and size. 5322 is a point, Endo includes the point, Fujisawa city includes the area included Endo, Kanagawa prefecture includes the area included Fujisawa, and Japan includes the area included Kanagawa. Figure 5.2 shows it. Therefore, an address indicates a hierarchical structure. It is the same all over the world.

The developed systems in this study indicated how to treat spatial information. It treats a position to control data broadcast without understanding the hierarchical structure. This system can control data broadcast by following keywords set by the subscriber. Figure 5.3 shows the overview of the system that understands hierarchical

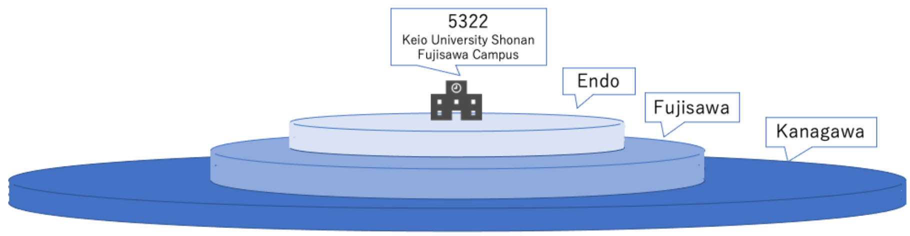


Figure 5.2: Example of Spatial hierarchical structure

Keio University Shonan Fujisawa Campus is built at 5322 Endo, Fujisawa, Kanagawa. A spatial representation of that address would be this hierarchical structure.

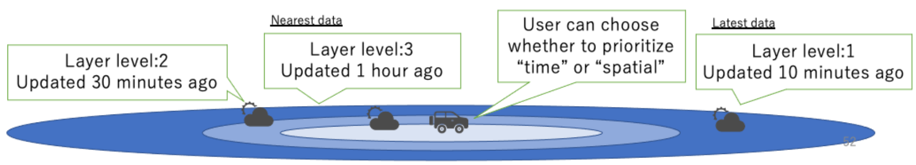


Figure 5.3: System overview of understanding hierarchies

The method uses layers to represent the region. Each layer has a level to express the degree of abstraction. It means layer size. A user can choose whether to prioritize using the information.

structure. If this system understands the hierarchical structure of the city, it can suggest other data instead of when not finding want data for the application, as shown in Figure 5.3.

Moreover, subscribers can set priority to receiving data. For example, each layer has a level to express the degree of abstraction. The higher its level, the more it points to a specific and narrow area. The subscriber chooses whether to prioritize the level or the last update time. The system sends data that it chooses using the information. It allows for a more flexible data broadcast that fits subscriber requirements.

5.2.2 Spatio-Temporal Information Processing in Real-Time

In this paper, the study focused on some use cases without real-time processing. From the result, the systems can use the use case examples in section 2.

On the other hand, the systems have some constraints when used for real-time pro-

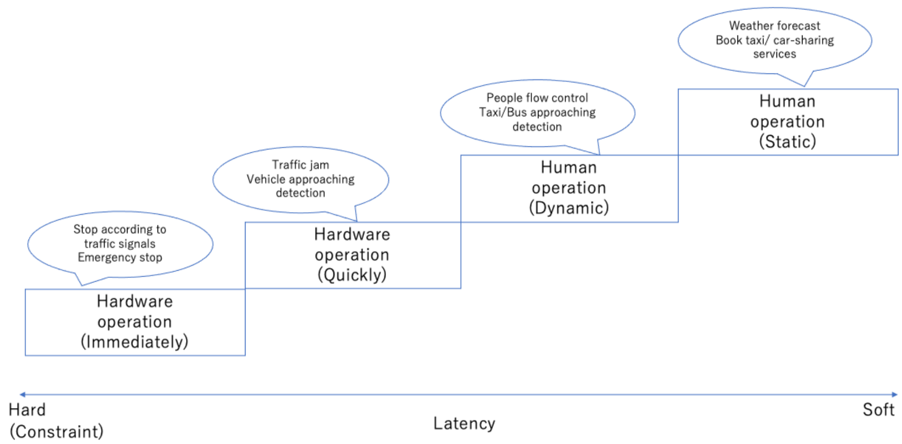


Figure 5.4: Requirement of latency each use case

The latency requirement depends on using situations. For example, very tight latency is required to immediately control hardware, such as an autonomous car stopping according to a traffic signal. On the other hand, when users book taxi services, the latency requirement is easy.

cessing.

It indicates examples that the systems are required hard constraints. Figure 5.4 shows the relations among use cases and the constraint of real-time processing speed. Real-time processing speed includes processing speed and network delay. Therefore, it means latency.

The horizontal axis of the figure.5.4 is latency. The left side is a hard constraint, and the right side is a soft constraint.

First, the system is required for most hard constraints when systems are used to control hardware immediately. For example, the tightest latency is required for dynamic hardware control, such as an autonomous car, because it needs to change the car's status according to traffic signals and under emergency. The constraint affects the safety of users. Second, hard latency constraints are required when systems are quickly used for hardware control. In this case, systems are used to know situations of a little future. For example, drivers and cameras are difficult to find other cars at the T-junction in advance. When the system can send the location to each other, they can prevent traffic accidents. Then, the latency of data sending is easier than traffic signals. The latency must be between a few seconds and tens of seconds.

When these processes are delayed, the hard real-time processings have risks of causing damage to users.

Third, soft constraints are required when systems are used for dynamic human control. For example, people flow control is included. When it holds a large-scale event, the number of visitors is so large in the urban city. Control of human flow is very important after the event because people want to move quickly. Hence, it causes the crowded stations of buses, trains, and taxis. However, in urban cities, public transportation has developed enough for users to choose various methods. Therefore, systems suggest other transportation routes for people by suitable traffic status. Data publishing must be speculative at this time because users cannot choose and move to the station immediately. Data must send the data minutes in advance. Last, the softest constraint is required when systems are used for static human control. These are such as booking systems. When a user travels from the day after tomorrow, he can know the weather forecast for the day after tomorrow at any time until the day after tomorrow. At this time, the latency is not almost affecting the user. Therefore, this case is the most accessible constraint. This study has focused on these soft real-time processes. The result in Section 4 confirmed that the systems developed in this study satisfy the requirements for soft real-time processes.

In addition, there is a limit to the number of users one platform can handle. Therefore, platform managers must consider the placement where they install them. It is also reflected in network latency. In this study, the middleware for IoT platforms manages to publish control and data aggregation. Nevertheless, the process becomes complex to realize them. It occurs with the increase in processing time. Hence, the latency is impaired throughout the service. When it realizes the use case of the left region (hardware operation) of figure 5.4 process sharing becomes important to improve the latency to realize real-time processing in the real world. Therefore, it is important to consider what should be processed by the client and what should be processed by the IoT platform. Moreover, the systems developed in this study do not have functions to control hard real-time processing. For example, the system cannot set priority for the delivery of data for clients. To realize hard real-time processing must need optimized control and high-speed transmission. It needs a new function to realize it.

When the system can solve all these problems, it can use to handle real-time processing.

This paper suggests one idea to solve the data publishing problem. It is the data sharing among M2M.

Other methods include:

- Placing IoT platforms where
- Setting priority on what publish critically
- Reducing the process on IoT platforms

For example, Scale and network issues can be resolved by where the IoT platform is deployed. When IoT platforms install for a short interval, per IoT platform can concentrate on publishing data in a narrow area. It leads to Local 5G network install, and it can realize the super high-speed data transmission.

Next, the latency requirement is different per application. Currently, SDS and DTOS publish data following the data on the stored database. They do not consider the priority per client. By providing them with the function to prioritize data delivery, data delivery that more appropriately satisfies latency requirements can be realized.

Share The Topic Information

If vehicles and pedestrians are individuals with different subscriber-topics, the advantages of pub/sub-model cannot be utilized. This study considered the method of sharing the topic information among nearby subscribers in the physical world. Sharing subscriber-topics among continuous subscribers (continuous means near their location) is valid because they have a relationship in this case. The ordering topic information does not depend on the user on DTOS because the user does not order the specific topic.

On the other hand, the system depends on location information, the kind of mobility, and the purpose. For example, when the system focuses on Several different groups of pedestrians sightseeing using the tourism application, their location is not different, and their purpose is the same. The system finds that these pedestrians want to receive the same data. In this case, As shown in Figure 5.5, DTOS can suggest the existed topic to the other subscriber by their purpose and their location. DTOS uses the location of each other, detection range, and direction to search sharing subscribers. When DTOS could find the sharing topic, the subscriber could receive data without creating a new subscriber-topic. Then, the important point is the actual data usage time.

For example, a traffic signal's effective range and update frequency are narrow and high. Figure 5.5 shows that the back subscriber is not in the effective range, and the

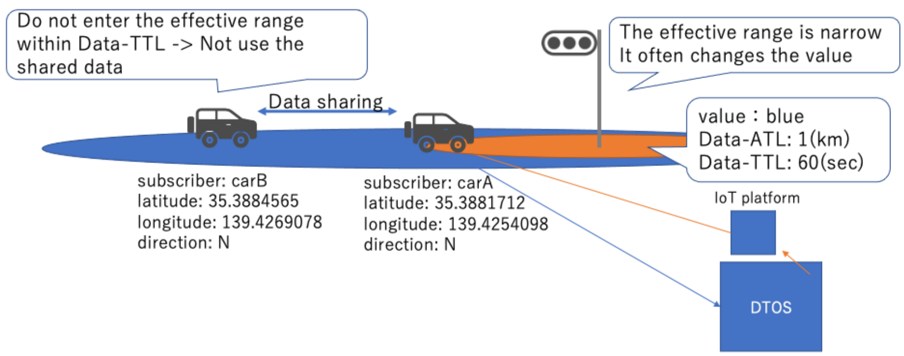


Figure 5.5: Overview of data sharing among vehicles

Vehicles need information on speed and direction as moving information.DTOS can judge the relationship among vehicles using each position and their moving information. Moreover, Data-TTL supports the decision of whether or not to use shared users.

value may have changed when it entered the effective range. Therefore, this study focuses on the time to live the data. This study calls it Data Time to Live (Data-TTL), and the effective range of the data is called Data Area to Live (data-ATL). The definition of Data-TTL has been described in the previous section. The publisher can set Data-TTL in seconds and Data-ATL in kilometers. These values are important to share subscriber-topics among subscribers.

5.2.3 City Platforms Integration

Each IoT platform per city is independent. The scale, platform system, base protocol, and the data handled are all different. Moreover, the data may contain sensitive information such as privacy. Then, system integration and federation are crucial.

First, it describes the practice of system integration in this study. This study developed the bridge system to integrate between SOXFire in Japan and Marketplace in Greece. SOXFire is implemented based on the pub/sub-model, using the XMPP protocol. It is very effective to handle complex sensor devices. Currently, many devices have several sensors per device. General pub/sub platforms do not store the information. When using general pub/sub platforms, publishers send data, including the meta-information, every time because of does not store the topic's meta-information. If the publisher does not include the information, it is not easy to specify the topic for subscribers who do not have sensor information already.

SOXFire solves the problem. The system has a powerful feature that stores meta-information per topic. Meta-information contains the handled topic name and data information free (name, unit, value range, Etc.). It stores the information so that users can know the detail of topics, and publishers send meta-information when they create topics only. This system can store information following the format defined in this study.

On the other hand, Marketplace in Greece has been implemented based on blockchain technologies. The system aims the realize open data commerce. Marketplace allows anyone to publish data and they sell and buy it. They can set the monetary value of data. It needs to offer a dependable data trading place. To offer a dependable data trading place must guarantee data security. Marketplace guarantees it using blockchain technologies.

SOXFire and Marketplace are different purposes. If these platforms integrate, the data published range spreads and adds new value to the data. This study developed SOXCollaborator to integrate them. At this work, the data flows from SOXFire to Marketplace. This reason is that SOXFire does not have a function to guarantee data security. Nevertheless, SOXFire can publish data on a large scale, so SOXCollaborator receives data from the SOXFire and uploading to Marketplace.

Figure 5.6 shows the system architecture.

The important point is the data format definition when developing SOXCollaborator. The sent data from the SOXFire includes the following information. The publisher name, topic name (sensor name or device name), the value of that, and publish timestamp. However, the case of users publishing data to the blockchain marketplace does not satisfy only this information. We thought of the following additional data. The login information, monetary value, and callback URL are used in the blockchain marketplace. The callback URL is the most important of those. A Publisher does not connect to the Marketplace directly when sending the data via the SOXFire. A Publisher cannot note when it is done action oneself data on the Marketplace. Therefore, clients need to prepare the API to receive the notification from the Marketplace. The blockchain marketplace can notify the client by using a callback URL when actions client's data. For instance, the time is when it sold the data. Table.5.1 shows all the necessary data to collaborate in the blockchain marketplace.

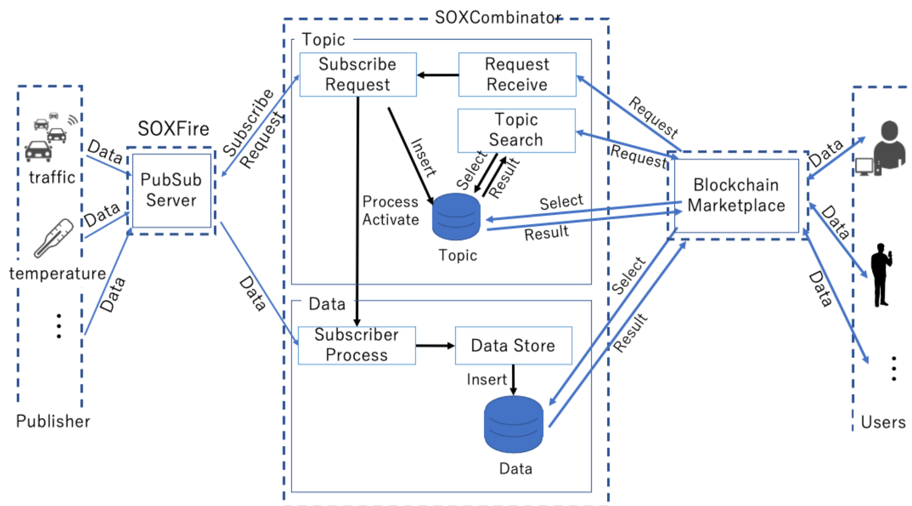


Figure 5.6: System configuration of SOXCollaborator

SOXCollaborator performs among the SOXFire server and Blockchain Marketplace. It do a role as bridge among them. Hence, the system has some functions, including subscribing to the SOXFire server and storing data on a database.

5.2.4 City Platforms Federation

The population of cities changes domestically because of the declining birthrate and aging population, population outflow, or overcrowded. Besides, large-scale disasters such as typhoons and tsunamis and pandemics such as COVID-19 attack cities and people worldwide. These disasters have happened over a wide area. Neighboring cities must help each other when involving disasters to support and help citizens. Solving the task focuses on making the city smart using information technologies.

However, as mentioned above, using platforms are different per city. Each platform has often implemented a different protocol, and they have different rules. Therefore, the platforms' interoperability is low. When realized Connected-smart-cities, platforms must install per city. They must connect to help and cooperate with each other. The previous section explained an integration method among different city platforms. It provides functions to subscribe and store and search instead of the users. The system does not consider the data publication range.

For example, a video of a surveillance camera in a city includes citizens' sensitive

Table 5.1: Necessary parameters for collaborating the blockchain marketplace

Eelements	Description
Publisher	Publisher name
Topic	Sensor or device or information name
Category	Information's category
Value	Value or contents
Timestamp	Publish timestamp
Username (login info)	Username of the external services
Password (login info)	Password of the external services
CallbackURL	Endpoint to receive notification

information. Those videos should usually only be made available to a limited number of people; for instance, the city's camera is monitored by the city's staff, and in the case of a building camera is monitored by the guards of the building. Nevertheless, other city workers also monitored it during an emergency, such as a disaster. Therefore, the data publishing range depends on the circumstances. This section's study provides federation in addition to an integrated function. The study aims to provide integration and federation flexibility among IoT platforms.

In this study, a method has been developed by Node-RED and SOXFire, Synerex[94], and FIWARE[95]. Node-RED[96] is open-source software. It can express a process as a node, as shown in Figure 5.7. Nodes are subjected to various processes. It can express the application data flow by connecting each node. Everyone can use it easily because Node-RED performs on GUI. In addition, Node-RED supports various protocols. Therefore, Node-RED judges the current city situation, and it supports data transformation for connecting other platforms.

IoT platform engineers receive two main benefits from using Node-RED.

- Not necessary to override the platform's programming: Developers' development cost is high when it requires the programming override to integrate or federate among different platforms. Node-RED avoids the problem. It already supports various protocols, and engineers can create custom nodes easily. It satisfies the interoperability to connect among IoT platforms that have different protocols.

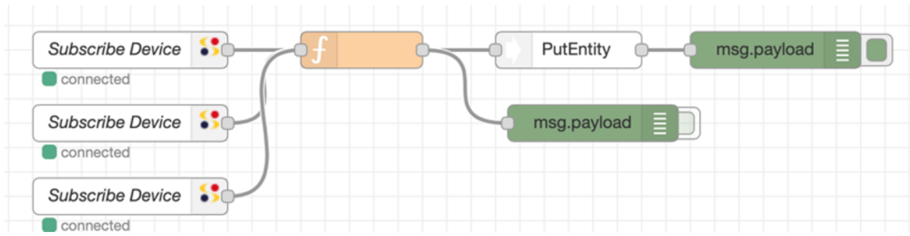


Figure 5.7: Image of Node-RED

Node-RED can express a process as a node, and developers can write the process each node, and nodes connect by a wire. Hence, the developer can express data flow on Node-RED.

- Provides the visualization of data flow: Node-RED expresses the data flow by combining nodes. Per node has one process, and wires connect nodes. A series of processes are expressed in them. When it models the city situations on Node-RED, other people can understand the data flow per city situation to watch and understand it. It causes a high mutual understanding among the system users.

Next, FIWARE is an IoT platform, and it uses mainly in Europe [97, 98, 99, 100]. FIWARE is a RESTFul-type IoT platform. In Europe, they develop extension systems to upgrade FIWARE. The platform clearly defined rules for writing data format and protocol. This study uses the authentification function of FIWARE. In addition, FIWARE is not developed by us. We have implemented SOXFire, and Synerex has been implemented by Nagoya University, so the authors already understand these platforms, but FIWARE is not. Therefore, FIWARE can be used to confirm the versatility of this study.

In this study, three platforms have different roles in the demonstration. Figure 5.8 shows their relationship. Management platform has functions to discriminate city situations.

It monitors the current status in cities and triggers to start processing other servers. A's city platform performs on FIWARE. The demonstration assumes this server usually limits the city's information publish. When a disaster happens, the server unlocks the limit, and other city workers can access the server. Last, the B's city platform on Synerex. This server can do visualization of city information. In addition, the demonstration assumes that the server and A's city platform are in a supportive relationship.

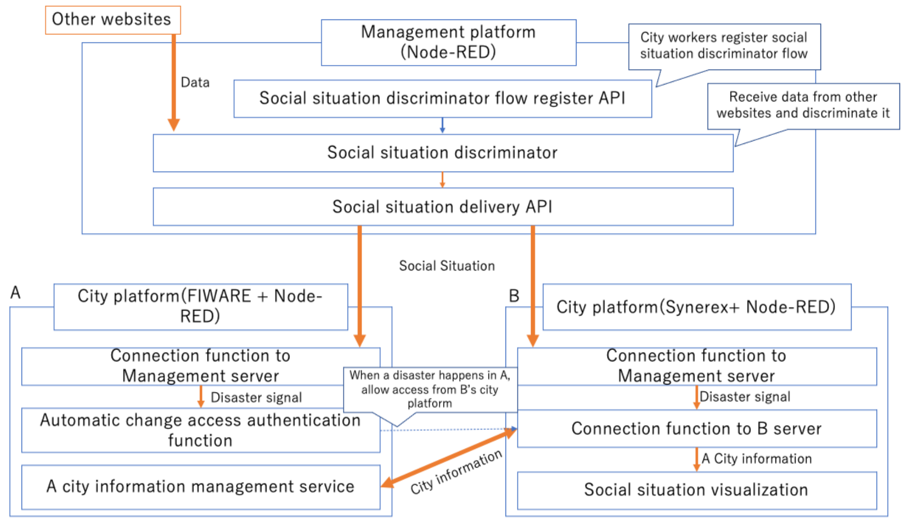


Figure 5.8: City platforms placement of a demonstration

Management platform usually does monitoring social situations from other websites. When a disaster happens, the platform sends the information to other platforms. Each platform behaves as appropriate for the situation.

When a disaster happens, the server accesses A's city platform, receives the information, and does visualization.

Then, automatic change in the authentification of the city platform is realized by Node-RED and FIWARE. Node-RED, FIWARE, and Synerex realize platform integration in the demonstration.

IWARE authentication manage application uses KEYROCK [101]. A demonstrator already creates Synrex users in KEYROCK. The users cannot usually access the target platform. In addition, it created the custom nodes to access FIWARE and to change the authentication of users in KEYROCK. The nodes connect to Management platform nodes. When a disaster happens in the city has A's city platform and Management platform catch the information from the website of the Japan Meteorological Agency. Then, Management server sends information to A's city platform and B's city platform via Node-RED. A server changes the authentication of Synrex users in KEYROCK following Node-RED nodes. Synerex users can access B's city platform and receive the information from the result. After that, B's city platform receives data from A's city platform via Node-RED. Then, B's city platform (Synerex) and A's city platform

(FIWARE) are defined in different data formats, but Node-RED's node converts the data of FIWARE to the data of Synerex. Finally, B's city platform does visualization after it receives the data correctly.

The series of processes can do automatically without human resources. In addition, the IoT platform program also does not change to realize it. It indicates the possibility of easily realizing the integration and federation among multiple platforms by the result of this study.

5.2.5 Data Security for Connected-Smart-Cities

Supporting connected-smart-cities is important for data security. Currently, the amount of data is increasing all over the world. It indicates the increasing number of data senders. By increasing them, the amount of data will be saturated worldwide. Then, anyone can upload data from anywhere and anytime. That is great, but it includes the risk. Anyone can become a sender, and malicious people also become one. In addition, fake news is a problem. Therefore, data security is important to sustain a healthy information world.

First, in information security, basic concepts are confidentiality, integrity, and availability [102].

Confidentiality: Require that the user is correctly configured for access authentication. Information often includes sensitive ones. It is personal information, secret information of an organization, and more. That information is never open to the public; only selected users should access it to protect privacy. Therefore, organizations must configure access authentication correctly. Nevertheless, correctly means without excess or deficiency. When it is a deficiency, the necessary user cannot access it. It causes them to stop what their work on. If the case of happened a disaster occurs, it is fatal for cities and living people. Therefore, organizations must configure access without excess or deficiency.

Integrity: Require that the information is perfect. Perfect means the latest and without falsification of the information. For example, when information flows over the network in plain-text, a malicious user can steal it. Moreover, the malicious user overwrites the information and sends it to the correct client. The client cannot know the fake information. It is a risk involving the crime. Therefore, the structure is important to guarantee that the data does not change by another user and can confirm that data has not been overwritten. Besides, the information is sometimes incorrect when the data does not update for a long time. A malicious user does not overwrite it, but the information is not correct. For instance, when map information does not update, the user misses the route to the destination. The critical information must sustain the latest information. Therefore, Perfect requires the latest without falsification of the information.

Availability: Require systems to continue to operate without stoppages due to system failures, Etc. For example, when a malicious user attacks the system and the system does not take the measure of it, the system gets down. It causes uncertainty in getting information for users. Availability indicates the occupancy rate of the system. System engineers must set its objective and take measures to prevent unexpected the system down.

It is important to guarantee information security, but it is not enough for high security. When using IoT platforms, the relationship between topics and senders is not a necessary one-to-one. It depends on the abstraction of a topic. For example, the robots ran synchronously; they shared their information via a common topic. The case of the sharing users that they know each other is no problem, but when they do not know each other, they do not know whether the sending data is correct or not. Therefore, the IoT platform is required dependable data exchange.

First, it is the guarantee of the sender. It proposes sending control to prevent impersonation. The systems developed in this study have the inquiring function. It checks the sender name and the name of the topic creator. If the names are different, the system rejects sending data. It prevents sending data from the other sender. However, the system checks the simple because the system compares only the name. The other sender can send data via the topic if he uses the name of the topic creator. It is not perfect. The system must have a login function. Login authentification can reject sending data by other senders. The function is implemented in general IoT platforms, including FIWARE, RabbitMQ, SOXFire, Synerex, Etc. Therefore, the system developed in this study can provide using the function implemented in IoT platforms.

Moreover, the data will have monetary value, as shown in an example of Marketplace. To trade data safely needs that prove its correctness. Correctness means the data does not consist the impersonation and falsification.

The impersonation by a malicious user can prevent a login function. However, when the data has been tampered with, users must see through it. For instance, Transport Layer Security (TLS) has been used to protect the data while flowing on the network with various protocols. TLS provides data encryption. Hence, malicious users cannot steal and overwrite the encrypted data while flowing on the network. Nevertheless, the latest data is stored on the server. The stored data must also encrypt because malicious users can overwrite it due to plain text. Therefore, the data stored on the server must be encrypted, and encrypted data must send to clients. It is a necessary key to encrypt and decrypt between a sender and a receiver. Exchanging the key between them is difficult because, using the pub/sub-model platform, publishers and subscribers connect via a broker. Therefore, it must consider the efficiency and safety method of exchanging keys.

5.3 Conclusion

By developing sensing technologies, there is increasing data worldwide. The basic concepts for supporting it are Internet-of-Things, Smart Sensing, and Participatory Sensing. The other technologies are embedded data on Internet, and more. These technologies increase in-city data significantly. The reason is that the sensing objects depend on cities. For example, at Participatory Sensing, the data is collected by people. In the case of the city being a target, citizens discover the problem in their city, and they post it to the city via systems.

In addition, the author did the practice of Smart Sensing. The practice focused on using public cars to collect data in a city. Taxis, buses, and garbage trucks drive a citywide. This study chose garbage trucks in them. Buses drive fixed routes, and they drive main roadways in a city. Hence, they may not drive on minor roadways. Taxis' driving routes depend on users, and the routes tend. Buses and taxis' are not easy to drive a citywide without tend. On the other hand, garbage trucks must drive citywide comprehensively because garbage is generated in people's lives. Therefore, this study lets garbage trucks collect in-city status.

This study developed a method to detect road makings damage. Road markings are painted wide-range, and they have damage quickly. City workers need to waste time and workforce costs to inspect it. This study realized the automatic inspection using image processing and CNN. The original model was named C's model. The result indicated that C's model has a high accuracy for detecting the status of road markings better than other machine learning models, LinearSVM and Random Forest. Moreover, the author tuned the model structure to perform on micron-computers, such as Raspberry pi. It realized the high accuracy and lightweight model.

From these backgrounds, everyone knows the number of in-city sensing data will increase. In the future, platforms for supporting it must be necessary. The platforms are required a suitable placement for installation in cities and distributed computing to process many data. This study organized and indicated the expression of in-city data. It described the three-dimension expression combining time and spatial (latitude and longitude). The three-dimension expression is named spatio-temporal information in this study. After describing related works and their tasks, this study defined each element of spatio-temporal information for Pub/Sub model IoT platforms.

First, it described the requirements of IoT platforms in this study. IoT platforms in Smart Cities are used for publishing data for several clients. In this study, the system requirements focus on independence. When IoT platforms are used in smart cities, they publish several clients and do not depend on clients' status. IoT platforms' role in smart cities is to separate data providers and receivers. Therefore, this study aims to extend IoT platforms to realize data sending control using spatio-temporal information processing. Besides, this study does not require latency because it implemented complex processing to realize data sending control.

Based on the above, we summarized this study's three target use cases, carpool, disaster prevention, and human-flow control. These use cases do not require hard constraint of latency.

In addition, the examples have two main requirements. It is integrity and commonality. Integrity means integrating or collaborating various data from several publishers. When the platform processes spatio-temporal information, it can flexibly control data delivery. It publishes some data at the same time from different senders. It will lead to integration and collaboration among some data.

Some users or applications in a situation use the standard template for receiving data. Commonality means the use of a common template. It shows that some clients aim for the information for the same purpose. Therefore, a data delivery platform can send the same data using the standard template. When the excepted case, templates depend on per client. It means the use case is difficult to share with some clients because it depends on the interest or privacy of each client. Above that, it indicated use cases.

Next, this study defined that time elements are expressed as Point-in-Time, Frequency, Data Time to Live (Data-TTL), and Data Time to Reference (Data-TTR). Point-in-Time is an essential element of time elements. Other defined elements are applications of it. This study developed a middleware system (SDS) using time to control data delivery. SDS extends platforms using pub/sub-model. The system provides two main contributions. First, it realizes the control of data delivery over time. It focuses on the frequency of data publishing. Pub/sub-model platforms send data based on push type. A server sends clients data immediately when it is received. Users cannot choose received or denied. It causes an over-receive for users. SDS configures a user receive frequency. It controls data delivery following the set parameter to suit a user's requirement.

Moreover, the system realized the topic abstraction over time. It is named subscriber-topic. A subscriber-topic manages multiple topics in a topic. It solves the publish interval async problem. The problem is caused by push-model data delivery. The reason is that the data publishing timing depends on the sender; It is independent because senders do not consider each other. Therefore, users must implement functions to synchronize multiple data when they want to combine them. A subscriber-topic sends data following the user's receive frequency, sending data of managed topics. It realizes synchronized data publishing for a user. In addition, by using these functions, the SDS can do data aggregation. Then, SDS uses a Data-TTR as additional time information.

Next, this study defined the spatial elements, Position, Distribution, and Data Area to Live (Data-ATL). Position is an essential element of them. Spatial information expresses the position of data that happened or was observed. Distribution expresses the relationship of data by distance. This study also developed the system (DTOS) using defined spatial elements. DTOS provides the automatic and dynamic topic change method. General pub/sub-model platforms do not store the topic information. DTOS stores topic information in a database. The information includes the position, Data-ATL, and Data-TTL. Users can receive data from suitable topics using them. DTOS has achieved a representation of spatial information that the system can handle.

In addition, combining SDS and DTOS has generated a speculative data publish method. The method is realized by synchronizing data publishing and using users' and topics' positions. It has been implemented by common elements and processing developed systems in this study. It indicates an example of the practice of using spatio-temporal information. Moreover, this study suggests five applications, periodic, dynamic, aggregation (time, spatial), and speculative. These applications use the same data format. Hence, it indicates spatio-temporal information abstraction.

Next, the base systems, SDS and DTOS, evaluated their specs. The systems generate additional costs and delays because they must be placed between clients and an IoT platform. Both systems have an increased system load when increasing numbers of subscriber-topic and publishers. The processing time also becomes long when the system load is high. However, the case of SDS and the speculative data publishing method process every minute. They do not process any faster than that. Hence, the long process time may not be a problem. A high system load can solve by improving a host machine's spec. A city platform must have high spec enough to process the applications because it affects citywide. In this study, the machine resource is minimum, but the actual machine has better resources than this study. Therefore, the next step of this study will evaluate and search the optimizing machine resources for a city through fieldwork.

Last, it described the future directions of this study. The most contribution is the defined spatio-temporal information and its abstraction. In this study, the real space's information, time and spatial, can is imported into cyberspace. Nevertheless, currently, the information does not understand the real space's all concepts. It suggested forward one concept that should be understood. It is spatial hierarchies. Spatial hierarchies express all positions in real space. The concept connects positions. It can indicate the relationship between position-to-position, area-to-are, and city-to-city. When it installs the concept to spatio-temporal information, users and systems can freely choose the data that considers the real space's features more flexibly.

Moreover, IoT platforms will be placed per city by contributions of this study. When all cities place IoT platforms, they can connect to each other in cyberspace. This study named the future city Connected-smart-cities. In Connected-smart-cities, cities can connect and collaborate freely. It proposes that the generation of new city functions helps cities improve living people's lives. This study describes the necessary concepts to realize connected-smart-cities. It is integration, federation, and data security.

Secondly, this study indicated the practice of federation. This study integrated and federated two different platforms as a demonstration: FIWARE and Synerex. The federation platform has developed by Node-RED. This reason is to realize the visualization of city situations and their flow. Actually, it created custom nodes to express and connect each platform. When a disaster in a city, the operation flow is excepted by nodes on Node-RED. At the demonstration, the center server captured a happened disaster from the website of the Japan Meteorological Agency. The server sends a notification to the city server on Node-RED. The city server changes the access authentication when it receives the notification. After that, Synerex can access the city server, displaying the information received from the city. This scenario indicated city federation.

Last, it described data security. There will be an increasing amount of data citywide and worldwide. Much of the data will be widely available. As the example of Marketplace, data will have a new value, including monetary value. Therefore, data security will be more important than now. This study described basic information security and additional important IoT data security concepts and technologies. These are applications to realize open-data commerce. All elements described in the future direction of this study are necessary concepts to realize next-generation cities.

Finally, this study indicated the spatio-temporal information abstraction and developed middleware for in-city data applications. The defined elements, developed systems, and future directions help improve people's lives. It hopes this study's result will be used in the future.

Bibliography

[1] Takafumi Kawasaki, Takeshi Iwamoto, Michito Matsumoto, Takuro Yonezawa, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. A Method for Detecting Damage of Traffic Marks by Half Celestial Camera Attached to Cars. In *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MOBIQUITOUS'15, page 246–253, Brussels, BEL, 2015. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[2] Takafumi Kawasaki, Makoto Kawano, Takeshi Iwamoto, Michito Matsumoto, Takuro Yonezawa, Jin Nakazawa, and Hideyuki Tokuda. Damage Detector: The Damage Automatic Detection of Compartment Lines Using A Public Vehicle and A Camera. In *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, MOBIQUITOUS 2016, page 53–58, New York, NY, USA, 2016. Association for Computing Machinery.

[3] Takafumi Kawasaki, Makoto Kawano, Takeshi Iwamoto, Michito Matsumoto, Takuro Yonezawa, Jin Nakazawa, and Hideyuki Tokuda. C's: Sensing The Quality of Traffic Markings Using Camera-Attached Cars. *SICE Journal of Control, Measurement, and System Integration*, 10(5):393–401, 2017.

[4] 別宮 広朗, 河崎 隆文, and 中澤 仁. 物体検出とセマンティックセグメンテーションによる車道上に限定した未知物体検出手法の提案. In **研究報告ユビキタスコンピューティングシステム（*UBI*）**, volume 2021-UBI72, page 1–8, 2021-11.

[5] 坂内 理人, 別宮 広朗, 河崎 隆文, 檀上 誠, and 中澤 仁. 物体検出を用いた道路陥没箇所検出モデルにおける合成画像を使用した学習の有効性の検討. In **研究報告ユビキタスコンピューティングシステム**（*UBI*）, volume 2021-UBI72, page 1–8, 2021-11.

[6] 河崎 隆文, and 中澤 仁. SDS : Pub/Sub モデルに基づく時間を用いたサブスクライバドリブンなデータ配信システム (特集 ソフトウェア論文). **コンピュータソフトウェア** = *Computer software*, 39(1):3–17, Feburary 2022.

[7] Takafumi Kawasaki, Tadashi Okoshi, and Jin Nakazawa. *A Mobility-Aware Pub/Sub Architecture for Short-Lived Data in Smart Cities: Poster Abstract*, page 645–646. Association for Computing Machinery, New York, NY, USA, 2020.

[8] Orfefs Voutyras, Aamir H. Bokhari, Akira Tsuge, Georgios Palaiokrassas, Takafumi Kawasaki, Xavier Cases-Camats, Jin Nakazawa, Antonios Litke, Tadashi Okoshi, and Theodora Varvarigou. A Methodology for Generating Systems Architectural Glimpse Statements Using The 5W1H Maxim. *Computers*, 10(10), 2021.

[9] Georgios Palaiokrassas, Petros Skoufis, Orfefs Voutyras, Takafumi Kawasaki, Mathieu Gallissot, Radhouene Azzabi, Akira Tsuge, Antonios Litke, Tadashi Okoshi, Jin Nakazawa, and Theodora Varvarigou. Combining Blockchains, Smart Contracts, and Complex Sensors Management Platform for Hyper-connected SmartCities: An IoT Data Marketplace Use Case. *Computers*, 10(10), 2021.

[10] Takuto Yamauchi, Takanori Hirano, Jialong Li, Takafumi Kawasaki, Yin Chen, Akira Tsuge, Tadashi Okoshi, Jin Nakazawa, Nobukazu Yoshioka, Georgios Palaiokrassas, Antonios Litke, and Kenji Tei. A development method for safe node-red systems using discrete controller synthesis. In *2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing Communications (GreenCom) and IEEE Cyber, Physical Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 130–137, 2021.

[11] 米澤 拓郎, 河崎 隆文, 吉田 拓人, 伊藤 友隆, 上津原 一利, 古城 篤, 中澤 仁, and 河口 信夫. Cityfeder: 異種スマートシティ基盤を柔軟につなぐプログラ

マブル・フェデレーション機構. In **マルチメディア，分散協調とモバイルシンポジウム** *2022* **論文集**, volume 2020, pages 11–18, jun 2020.

[12] Twopager Amsterdam Smart City. `https://issuu.com/amsterdamsmartcity3/docs/asc-twopager-0416`.

[13] Smart Santander. `https://www.smartsantander.eu/`.

[14] Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, et al. SmartSantander: IoT Experimentation over A Smart City Testbed. *Computer Networks*, 61:217–238, 2014.

[15] Toyota Woven City. `https://www.woven-city.global/jpn`.

[16] Tom Jefferies and Jane Anderson. city verve internet of things demonstrator: City verve commissions: Manchester's plinth.

[17] Aksel Ersoy. The Impact of Smart City Technologies: Lessons from Three UK Cities. *Proceedings of the Institution of Civil Engineers-Urban Design and Planning*, 172(5):169–175, 2019.

[18] Degital futures, MANCHESTER The Univesity of Manchester. `https://www.digitalfutures.manchester.ac.uk/what_we_do/societal-challenges/`.

[19] Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, and Ernoe Kovacs. Building A Big Data Platform for Smart Cities: Experience and Lessons from Santander. In *2015 IEEE International Congress on Big Data*, pages 592–599. IEEE, 2015.

[20] Mitsui Repark. `https://www.repark.jp/`.

[21] Enoshima Yacht Harbor. `https://www.s-n-p.jp/enoshima-yacht-harbor/weather_info.html`.

[22] Hakone Navi. `https://www.hakonenavi.jp/volcanic_gas_info/volcanic_gas/`.

[23] Jin Nakazawa, Hideyuki Tokuda, and Takuro Yonezawa. Sensorizer: An Architecture for Regenerating Cyber Physical Data Streams from The Web. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, page 1599–1606, New York, NY, USA, 2015. Association for Computing Machinery.

[24] Mina Sakamura, Tomotaka Ito, Hideyuki Tokuda, Takuro Yonezawa, and Jin Nakazawa. Minaqn: Web-based Participatory Sensing Platform for Citizen-centric Urban Development. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 1607–1614, 2015.

[25] BF Spencer Jr, Manuel E Ruiz-Sandoval, and Narito Kurata. Smart Sensing Technology: Opportunities and Challenges. *Structural Control and Health Monitoring*, 11(4):349–368, 2004.

[26] Azedine Charef, Antoine Ghauch, Patrick Baussand, and Michel Martin-Bouyer. Water Quality Monitoring Using A Smart Sensing System. *Measurement*, 28(3):219–224, 2000.

[27] Nathalie Mitton, Symeon Papavassiliou, Antonio Puliafito, and Kishor S Trivedi. Combining Cloud and Sensors in A Smart City Environment. *EURASIP journal on Wireless Communications and Networking*, 2012(1):1–10, 2012.

[28] N Suma, Sandra Rhea Samson, S Saranya, G Shanmugapriya, and R Subhashri. IoT Based Smart Agriculture Monitoring System. *International Journal on Recent and Innovation Trends in computing and communication*, 5(2):177–181, 2017.

[29] M Haefke, SC Mukhopadhyay, and H Ewald. A Zigbee Based Smart Sensing Platform for Monitoring Environmental Parameters. In *2011 IEEE International Instrumentation and Measurement Technology Conference*, pages 1–8. IEEE, 2011.

[30] Mustafa Kocakulak and Ismail Butun. An Overview of Wireless Sensor Networks Towards Internet of Things. In *2017 IEEE 7th annual computing and communication workshop and conference (CCWC)*, pages 1–6. Ieee, 2017.

[31] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. Participatory Sensing. In *WSW'06 at SenSys'*, 2006.

[32] Wataru Sasaki, Yusaku Eigen, Arturo Medela, Antonis Litke, Vanessa Clemente Núnez, Tadashi Okoshi, and Jin Nakazawa. SmileCityReport: Emotion-aware Participatory Sensing for Smart Cities with Double-sided Photo Shooting. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2020.

[33] Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. From Participatory Sensing to Mobile Crowd Sensing. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 593–598. IEEE, 2014.

[34] Mina Sakamura, Takuro Yonezawa, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. LiPS: Linked Participatory Sensing for Optimizing Social Resource Allocation. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 1015–1024, 2014.

[35] Takuro Yonezawa, Mina Sakamura, Nobuo Kawaguchi, and Jin Nakazawa. Civic Crowdsensing Through Location-aware Virtual Monsters. In *International Conference on Human-Computer Interaction*, pages 464–476. Springer, 2020.

[36] Viengnam Douangphachanh and Hiroyuki Oneyama. A Study on The Use of Smartphones for Road Roughness Condition Estimation. *Journal of the Eastern Asia Society for Transportation Studies*, 10:1551–1564, 2013.

[37] Tetsuya Kuno and Hiroaki Sugiura. Detection of Road Conditions with CCD Cameras Mounted on A Vehicle. *Systems and computers in Japan*, 30(14):88–99, 1999.

[38] Hiroya Maeda, Yoshihide Sekimoto, and Toshikazu Seto. An Easy Infrastructure Management Method Using On-board Smartphone Images and Citizen Reports by Deep Neural Network. In *Proceedings of the Second International Conference on IoT in Urban Space*, pages 111–113, 2016.

[39] Adham Mohamed, Mohamed Mostafa M Fouad, Esraa Elhariri, Nashwa El-Bendary, Hossam M Zawbaa, Mohamed Tahoun, and Aboul Ella Hassanien. RoadMonitor: An Intelligent Road Surface Condition Monitoring System. In *Intelligent Systems' 2014*, pages 377–387. Springer, 2015.

[40] Junji Takahashi, Yusuke Kobana, Yoshito Tobe, and Gullaume Lopez. Classification of Steps on Road Surface Using Acceleration Signals. In *Proceedings of the 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services on 12th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 229–234, 2015.

[41] Kazuya Tomiyama, Akira Kawamura, S Fujita, and T Ishida. An Effective Surface Inspection Method of Urban Roads According to The Pavement Management Situation of Local Governments. *Journal of Japan Society of Civil Engineers, Ser. F3 (Civil Engineering Informatics)*, 69(2):I–54, 2013.

[42] Boyu Zhao, Tomonori Nagayama, Noritoshi Makihata, Masashi Toyoda, Muneaki Takahashi, and Masataka Ieiri. Iri Estimation by The Frequency Domain Analysis of Vehicle Dynamic Responses and Its Large-scale Application. In *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, pages 41–46, 2016.

[43] Yin Chen, Jin Nakazawa, Takuro Yonezawa, Takafumi Kawsaki, and Hideyuki Tokuda. Cruisers: A Public Automotive Sensing Platform for Smart Cities. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pages 767–768. IEEE, 2016.

[44] Yin Chen, Jin Nakazawa, Takuro Yonezawa, Takafumi Kawasaki, and Hideyuki Tokuda. An Empirical Study on Coverage-Ensured Automotive Sensing Using Door-to-Door Garbage Collecting Trucks. In *Proceedings of the 2nd International Workshop on Smart*, SmartCities '16, New York, NY, USA, 2016. Association for Computing Machinery.

[45] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing Multimedia Content Using Attention-based Encoder-decoder Networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015.

[46] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[47] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital Selection and Analogue Amplification Coexist in A Cortex-inspired Silicon Circuit. *nature*, 405(6789):947–951, 2000.

[48] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[49] Christopher JC Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[50] Johan AK Suykens and Joos Vandewalle. Least Squares Support Vector Machine Classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[51] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image Classification Using Random Forests and Ferns. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee, 2007.

[52] Emir Kremic and Abdulhamit Subasi. Performance of Random Forest and SVM in Face Recognition. *Int. Arab J. Inf. Technol.*, 13(2):287–293, 2016.

[53] Andy Liaw, Matthew Wiener, et al. Classification and Regression by RandomForest. *R news*, 2(3):18–22, 2002.

[54] Claudia Lindner, Paul A Bromiley, Mircea C Ionita, and Tim F Cootes. Robust and Accurate Shape Model Matching using random forest regression-voting. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1862–1874, 2014.

[55] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[56] Wenjian Xu, Chi-Yin Chow, Man Lung Yiu, Qing Li, and Chung Keung Poon. MobiFeed: A Location-aware News Feed System for Mobile Users. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 538–541, 2012.

[57] Jie Bao and Mohamed F. Mokbel. GeoRank: An Efficient Location-Aware News Feed Ranking System. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, page 184–193, New York, NY, USA, 2013. Association for Computing Machinery.

[58] Wenjian Xu and Chi-Yin Chow. A Location- and Diversity-Aware News Feed System for Mobile Users. *IEEE Transactions on Services Computing*, 9(6):846–861, 2016.

[59] Yahoo News.
`https://news.yahoo.co.jp/`.

[60] Ludger Fiege, Felix C Gartner, Oliver Kasten, and Andreas Zeidler. Supporting Mobility in Content-based Publish/Subscribe Middleware. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 103–122. Springer, 2003.

[61] 落合 秀也, 木村 慎吾, 松浦 知史, 砂原 秀樹, 福原 英之, and 江崎 浩. Content-Based Network におけるデータ処理コンポーネント最適配置問題. **研究報告マルチメディア通信と分散処理** *(DPS)*, 2009(12):1–8, 2009.

[62] 宇野 聡将, 井ノ口 仁也, and 青山 幹雄. MQTT を用いた車載エッジコンピューティングアーキテクチャの提案と評価. **第** *79* **回全国大会講演論文集**, 2017(1):229–230, 2017.

[63] Michael J Pazzani. A Framework for Collaborative, Content-based and Demographic Filtering. *Artificial intelligence review*, 13(5):393–408, 1999.

[64] Guoliang Li, Yang Wang, Ting Wang, and Jianhua Feng. Location-aware Publish/Subscribe. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 802–810, 2013.

[65] Lisi Chen, Gao Cong, and Xin Cao. An Efficient Query Indexing Mechanism for Filtering Geo-textual data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 749–760, 2013.

[66] Gianpaolo Cugola and Alessandro Margara. High-performance Location-aware Publish-subscribe on GPUs. In *ACM/IFIP/USENIX International Conference*

on Distributed Systems Platforms and Open Distributed Processing, pages 312–331. Springer, 2012.

[67] Efficient Construction of Safe Regions for Moving KNN Queries over Dynamic Datasets.

[68] Weihuang Huang, Guoliang Li, Kian-Lee Tan, and Jianhua Feng. Efficient Safe-region Construction for Moving Top-k Spatial Keyword Queries. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 932–941, 2012.

[69] Long Guo, Jie Shao, Htoo Htet Aung, and Kian-Lee Tan. Efficient Continuous Top-k Spatial Keyword Queries on Road Networks. *GeoInformatica*, 19(1):29–60, 2015.

[70] Michael Reibel. Geographic Information Systems and Spatial Data Processing in Demography: A Review. *Population Research and Policy Review*, 26(5):601–618, 2007.

[71] Sara L McLafferty. GIS and Health Care. *Annual review of public health*, 24(1):25–42, 2003.

[72] John Fortney, Kathryn Rost, Mingliang Zhang, and James Warren. The Impact of Geographic Accessibility on The Intensity and Quality of Depression Treatment. *Medical care*, pages 884–893, 1999.

[73] Robin Haynes, Graham Bentham, Andrew Lovett, and Susan Gale. Effects of Distances to Hospital and GP Surgery on Hospital Inpatient Episodes, Controlling for Needs and Provision. *Social science & medicine*, 49(3):425–433, 1999.

[74] Abolfazl Mollalo, Behzad Vahedi, and Kiara M Rivera. GIS-based Spatial Modeling of COVID-19 Incidence Rate in The Continental United States. *Science of the total environment*, 728:138884, 2020.

[75] Ivan Franch-Pardo, Brian M Napoletano, Fernando Rosete-Verges, and Lawal Billa. Spatial Analysis and GIS in The Study of COVID-19. A Review. *Science of the total environment*, 739:140033, 2020.

[76] Srikanta Sannigrahi, Francesco Pilla, Bidroha Basu, Arunima Sarkar Basu, and Anna Molter. Examining The Association Between Socio-demographic Composition and COVID-19 Fatalities in The European Region Using Spatial Regression Approach. *Sustainable cities and society*, 62:102418, 2020.

[77] Marc Schröder and Pedro Cabral. Eco-friendly 3D-Routing: A GIS Based 3D-Routing-Model to Estimate and Reduce CO2-emissions of Distribution Transports. *Computers, Environment and Urban Systems*, 73:40–55, 2019.

[78] Elaine Sherman, Anil Mathur, and Ruth Belk Smith. Store Environment and Consumer Purchase Behavior: Mediating Role of Consumer Emotions. *Psychology & Marketing*, 14(4):361–378, 1997.

[79] Chung-Hoon Park and Young-Gul Kim. Identifying Key Factors Affecting Consumer Purchase Behavior in An Online Shopping Context. *International journal of retail & distribution management*, 2003.

[80] Shabnam HA Zanjani, George R Milne, and Elizabeth G Miller. Procrastinators' Online Experience and Purchase Behavior. *Journal of the academy of marketing science*, 44(5):568–585, 2016.

[81] Rahul Bhagat, Srevatsan Muralidharan, Alex Lobzhanidze, and Shankar Vishwanath. Buy It Again: Modeling Repeat Purchase Recommendations. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 62–70, 2018.

[82] ノッカルあさひまち. `https://www.town.asahi.toyama.jp/gyosei/gaiyo/kotsu/1594702432592.html`.

[83] Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Distributed Event Aggregation for Content-based Publish/Subscribe Systems. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 95–106, 2014.

[84] Jose Rivera, Christoph Goebel, and Hans-Arno Jacobsen. Distributed Convex Optimization for Electric Vehicle Aggregators. *IEEE Transactions on Smart Grid*, 8(4):1852–1863, 2017.

[85] Navneet Kumar Pandey, Kaiwen Zhang, Stéphane Weiss, Hans-Arno Jacobsen, and Roman Vitenberg. Minimizing The Communication Cost of Aggregation in Publish/Subscribe Systems. In *2015 IEEE 35th International Conference on Distributed Computing Systems*, pages 462–473, 2015.

[86] Riccardo Giambona, Alessandro EC Redondi, and Matteo Cesana. MQTT+ Enhanced Syntax and Broker Functionalities for Data Filtering, Processing and Aggregation. In *Proceedings of the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pages 77–84, 2018.

[87] Peter Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. Technical report, 2011.

[88] Steve Vinoski. Advanced Message Queuing Protocol. *IEEE Internet Computing*, 10(6):87–89, 2006.

[89] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. MQTT-S—A Publish/Subscribe Protocol for Wireless Sensor Networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE, 2008.

[90] Roger A Light. Mosquitto: Server and Client Implementation of The MQTT protocol. *Journal of Open Source Software*, 2(13), 2017.

[91] Hamdan Hejazi, Husam Rajab, Tibor Cinkler, and László Lengyel. Survey of Platforms for Massive IoT. In *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*, pages 1–8. IEEE, 2018.

[92] RabbitMQ. `https://www.rabbitmq.com/`.

[93] Takuro Yonezawa, Tomotaka Ito, Jin Nakazawa, and Hideyuki Tokuda. SOX-Fire: A Universal Sensor Network System for Sharing Social Big Sensor Data in Smart Cities. In *Proceedings of the 2nd International Workshop on Smart*, pages 1–6, 2016.

[94] 河口 信夫, 米澤 拓郎, and 廣井 慧. Synerex: 超スマート社会を支える需給交換プラットフォームの設計コンセプトと機能. **研究報告モバイルコンピューティングとパーベイシブシステム** *(MBL)*, 2020(49):1–6, 2020.

[95] FIWARE. `https://www.fiware.org/`.

[96] Node-RED. `https://nodered.org/`.

[97] Flavio Cirillo, Gürkan Solmaz, Everton Luís Berz, Martin Bauer, Bin Cheng, and Ernoe Kovacs. A Standard-based Open Source IoT Platform: FIWARE. *IEEE Internet of Things Magazine*, 2(3):12–18, 2019.

[98] Victor Araujo, Karan Mitra, Saguna Saguna, and Christer Åhlund. Performance Evaluation of FIWARE: A Cloud-based IoT Platform for Smart Cities. *Journal of Parallel and Distributed Computing*, 132:250–261, 2019.

[99] Maria Fazio, Antonio Celesti, Fermín Galán Márquez, Alex Glikson, and Massimo Villari. Exploiting The FIWARE Cloud Platform to Develop A Remote Patient Monitoring System. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 264–270. IEEE, 2015.

[100] JA López-Riquelme, N Pavón-Pulido, H Navarro-Hellín, F Soto-Valles, and R Torres-Sánchez. A Software Architecture Based on FIWARE Cloud for Precision Agriculture. *Agricultural water management*, 183:123–135, 2017.

[101] FIWARE KEYROCK.
`https://fiware-idm.readthedocs.io/en/latest/`.

[102] Georg Disterer. ISO/IEC 27000, 27001 and 27002 for Information Security Management. *Journal of Information Security*, 2013.