Research Article

STEIM: A Spatiotemporal Event Interaction Model in V2X Systems Based on a Time Period and a Raster Map

Cheng Xu, Hengjie Luo, Hong Bao, and Pengfei Wang

1Beijing Key Laboratory of Information Service Engineering, College of Robotics, Beijing Union University, Beijing 100101, China
2Communication and Information Centre, State Administration of Work Safety, Beijing 100013, China

Correspondence should be addressed to Hong Bao; baohong@buu.edu.cn

Received 24 April 2019; Revised 2 November 2019; Accepted 4 December 2019; Published 10 February 2020

Academic Editor: Jose F. Monserrat

Copyright © 2020 Cheng Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Vehicles (IoV) is an important artificial intelligence research field for intelligent transportation applications. Complex event interactions are important methods for data flow processing in a Vehicle to Everything (V2X) environment. Unlike the classic Internet of Things (IoT) systems, data streams in V2X include both temporal information and spatial information. Thus, effectively expressing and addressing spatiotemporal data interactions in the IoV is an urgent problem. To solve this problem, we propose a spatiotemporal event interaction model (STEIM). STEIM uses a time period and a raster map for its temporal model and spatial model, respectively. In this paper, first, we provide a spatiotemporal operator and a complete STEIM grammar that effectively expresses the spatiotemporal information of the spatiotemporal event flow in the V2X environment. Second, we describe the design of the operational semantics of the STEIM from the formal semantics. In addition, we provide a spatiotemporal event-stream processing algorithm that is based on the Petri net model. The STEIM establishes a mechanism for V2X event-stream temporal and spatial processing. Finally, the effectiveness of the STEIM-based system is demonstrated experimentally.

1. Introduction

The Vehicle-to-Everything (V2X), which is the information exchange between vehicular systems and the outside world of information exchange, is a critical technology for future intelligent transportation systems. V2X enables vehicles, base stations, and infrastructure to obtain real-time traffic information, road information, and pedestrian information [1]. V2X improves safety, reduces congestion, improves traffic efficiency, and provides on-board entertainment information [2]. Currently, V2X applications in the Internet of Vehicles (IoV) remains in the exploratory stage as technical applications for traffic safety, information security problems, and all types of application stability, and interoperability problems have not been tested [3].

V2X provides a connection with the outside world; thus, it is a basic and key technology for intelligent automobiles, autonomous vehicles, and future intelligent transportation systems [4]. V2X primarily consists of four functions: Vehicle-to-Network (V2N), Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Pedestrian (V2P). V2X uses Dedicated Short Range Communications (DSRC) and Long-Term Evolution-Vehicle (LTE-V); both DSRC and LTE-V are well-known technical standards in the industry [5, 6].

In the Intelligent Connected Vehicle (ICV) environment, the cooperative intelligent interaction communication mechanism is the interaction mode between each intelligent agent. Data streams in the V2X include both temporal information and spatial information. In applications, vehicles periodically broadcast a Basic Safety Message (BSM) to other intelligent agents, including information such as position, longitude, latitude, posture state, heading angle, velocity, and acceleration to surrounding vehicles. The vehicle simultaneously receives similar information from surrounding intelligent agents. By processing and analysing these data streams, a vehicle can determine the current values of the surrounding agents. Real-time acquisition, interaction, and processing of information from surrounding vehicles and
the relative position of the current vehicle, combined with the actual working situation of vehicle traffic, are numerous complex events [7]. As a starting point, to ensure vehicle and public safety, a variety of different conditions may be derived from this information, such as early warning of unknown states, early warning of known states, emergency braking situations, and vehicle lane change warnings [8, 9].

Vehicles run in a complex environment in real time. Thus, complex event-processing technology is required to filter the massive data stream generated by event instance flows to obtain those that concern the user via an event-processing model [10, 11]. Events that concern users are sent to a client events library or disposed of based on a user’s predefined rules. The corresponding processing must also occur in real time [12, 13]. Complex event-processing technology has been successfully applied to IoV systems; however, the spatial data model in V2X driving scenarios has two problems: first, it is incompatible with the spatial model in a geographic information system. Without considering the direction of spatial data, a model that is suitable for both time and space is not available. Second, the direction of the spatial data is not considered. In a complex event processing model for the Internet of Things, the adoption of an appropriate temporal and spatial model can reduce the complexity of the model [14, 15].

1.1. Our Contributions. How to effectively express and address the spatiotemporal data flow in the IoV in V2X is a critical problem that must be solved to satisfy the challenges of the massive temporal and spatial information data stream produced in a vehicular network. In this paper, a spatio-temporal event interaction model (STEIM) is proposed. The STEIM model establishes the grammar and the operational semantics. The contributions of the current study are summarised as follows:

1. A spatiotemporal event interaction model is proposed for V2X. This model establishes the model’s syntax and operational semantics.
2. The raster map model in a geographic information system (GIS) is adopted as the spatial model in V2X. Based on this raster map spatial model, we provide a method for determining the spatial data direction relations in a vehicular network.
3. An event flow-processing algorithm model that is based on a Petri net is adopted. We expound on the effectiveness of the processing mechanism in the experimental results and analysis section.

1.2. Paper Organization. The remainder of this paper is organised as follows: In Section 2, we provide an overview of related studies and similar systems that inform this framework. We provide the data model and the temporal and spatial events for V2X in Section 3. Section 4 presents the STEIM grammar. Section 5 describes the design of STEIM and its operational semantics. Section 6 presents the STEIM algorithm based on a Petri net. The experimental results that concern the task performance and efficiency are provided in Section 7. Section 8 presents our concluding remarks.

2. Related Work

Many studies have discussed complex event handling in V2X. The approach proposed in Ashok and Vaidhehi [16] primarily relies on spatial abstractions of each object and mining frequent temporal patterns in a sequence of frames to form a regular temporal pattern. This process is repeated for each window of sequences, and support for a temporal sequence is obtained to discover the regular patterns of normal events. However, this approach depends on the outcomes of object detection and tracking methods that are applied to videos. Liu and El Kamel [17] used an algorithm that is designed to improve the information throughput by reorganising the surrounding vehicle platoons. The algorithm considers safety, fuel consumption, speed limits, heterogeneous vehicle features, and passenger comfort. However, the algorithm needs to determine when to activate the algorithm. Li and Yixiang [18] proposed a novel spatial-temporal complex event-processing language for vehicular networks that establishes syntax based on modified spatiotemporal relation operators. In addition, they designed an event-instance stream-processing algorithm. However, their approach did not consider the problem of protecting information security or reliable transmission of incident flow. The approaches in [19–22] achieve a reasonable level of privacy protection. In these studies, vehicles are issued pseudonym certificates, and certificate generation and provisioning are divided among multiple organisations. The schemes support anonymous authentication in V2X. Subsequent research includes automatic adaptation and optimisation of the configuration parameters for the selected road network but also involves heuristic optimisation of these parameters during a preprocessing procedure.

In terms of visual analysis of traffic data, Wang et al. [23] used visualization technology to design a spatiotemporal data visualization platform for traffic based on points and heat, which could accurately reflect road traffic conditions and provide scientific and reliable data support for urban road traffic planning and management. Sarraf and McGuire [24] conducted visualization research on accident data from sensors and designed an interactive view interface integrating multisource data. So et al. [25] proposed a traffic congestion index visual analysis system based on GPS trajectory. In conclusion, data visual analysis is increasingly used in intelligent transportation. Data visualization has achieved certain results in the application of traffic data such as trajectory data, traffic flow data, and taxi data. However, there are few research results on traffic accident data transmission visualization.

In terms of the identification method of accident occurrence, with the continuous increase of car ownership, the road traffic safety situation is particularly serious. Identifying traffic accidents requires traffic police departments to allocate police forces and deploy rescue equipment according to different road sections [26]. It is of great significance to improve the ability of traffic control. The identification of
multiple accident sections is based on the safety of sections to determine the dangerous sections of highways. At present, the identification methods of road sections with multiple traffic accidents mainly include the following categories: (1) applying traffic conflict analysis technology [27]. It judges the possible traffic conflict points by predicting the running track of vehicles and judges the more serious traffic conflicts as the road section with frequent accidents. The method relies less on the historical data of traffic accidents, but the workload of traffic conflict analysis method is larger. Therefore, it is only applicable to the identification of traffic accidents on small urban roads. (2) Identification method based on historical traffic accident data. It first segmented the studied road manually. Then the accident number method and the accident rate method are used to identify the road sections with multiple accidents. Or the accident data can be analysed by building models such as Poisson regression, negative binomial regression, and empirical Bayesian method, and the multiaccident sections can be judged according to the prediction of the accident development trend [28]. This kind of method requires manual segmentation of the road before identification, and the result of segmentation will directly affect the prediction of traffic accident-prone sections. On the other hand, time and weather conditions have a significant impact on traffic safety. The above identification methods of traffic accident-prone sections do not take into account time and weather conditions, which greatly reduces the accuracy of the identification results [29].

For the visualization of accident-prone road sections, Fan et al. [30] proposed context-aware big data analysis and spatiotemporal visualization methods for traffic data such as road networks, vehicles, drivers, and weather. Shafabakhsh et al. [31] combined geographic information technology and spatial statistical analysis to reveal the influence of spatial factors on fatal, injured, property loss, and other traffic accidents. Among them, nearest neighbor analysis and k-function were used to study the distribution of urban traffic accidents, and kernel density estimation was used to identify the dangerous areas in the urban network. Based on a large amount of human activity data and traffic accident data and combined with real-time data, Li et al. [32] established a universal prediction model for large-scale real-time traffic accident risk simulation.

Some event-processing models focus on the spatial relationships between two events. Jin et al. [33] presented a composite subscription model that describes moving events related to mobile objects and detects whether events are matched with subscriptions in a distributed manner. However, some objects do not share GIS spatial information to obtain a global map. Lee et al. [34] designed a road-network-aware spatial alarm processing system by developing a suite of optimisation techniques using motion-aware filters. The system can increase the hibernation time of mobile clients and reduce wakeups and alarm check frequencies while ensuring a high spatial alarm processing accuracy. Zheng et al. [35] used taxi trajectory data to study the spatiotemporal characteristics of urban residents’ travel patterns from popular areas and hotspots. They proposed a spatiotemporal trajectory clustering method to discover hotspots among popular destinations. However, Lee and Zheng did not consider the direction of spatial data.

Our project team began researching individual intelligent vehicles in 2012. We began our research on the technology, platform, and transfer of information concerning driving brain cognition for intelligent vehicles in 2014 [36–38]. Since 2016, we have been studying how V2X vehicles and subsystems should authenticate with each other [2, 39, 40–44]. The spatiotemporal event interaction model architecture is shown in Figure 1, which includes both individual intelligent vehicles.

Based on the actual surrounding vehicles relative to the location of the current vehicle, the vehicle relative location map in Figure 2 can be divided into 13 relative positions according to distance and orientation, namely, front, right front, left front, far ahead, far front left, far front right, left, right, left rear, right rear, far behind, far back left, and far back right. The vehicle relative location map in Figure 2 can be divided into 14 relative positions according to distance and orientation, namely, front, right front, left front, far ahead, far front left, far front right, left, right, left rear, right rear, back, far behind, far back left, and far back right.

Statistical analyses of the causes of traffic accidents are based on accident frequency and loss cost factors, such as comprehensive consideration. The working condition of an accident environment can be divided into vehicles that should stop, loss of vehicle control, vehicle turns, slowing down in the absence of light, and vehicle lane changes. Similarly, the system functions in V2X scenarios spawned a variety of applications for emergency cases, such as abnormal vehicle reminders, dangerous road condition reminders, speed limit warnings, warning for running a red light, dangerous traffic participants collision warnings, green wave guide, congestion reminders, and emergency vehicles ahead. These advanced warnings can help improve the safety and efficiency of public traffic and public transportation systems.

The specific scene application function of V2X systems involves distinguishing the relative positions of two objects, calculating the distance and time to collision, and providing a warning.

In this paper, we focus on generalising our previous work on formalising driving brain cognition for intelligent vehicles. In addition, we adopt a time period and raster map to form a temporal model and a spatial model, respectively, when building the spatiotemporal event interaction model. The result is the establishment of a mechanism that can be formulated in V2X.

3. Driving Brain Cognition

Formalization in V2X

In V2X, various sensors continuously generate a large number of multisource and multidimensional data. These data include time, location, direction, speed, acceleration, temperature, license plate numbers, and traffic signals. The majority of these data can be expressed using a common numerical model; for example, velocity can be expressed as a
The driving brain formalisation includes a time data model, a spatial data model, and a temporal event instance model.

### 3.1. The Time Data Model of the V2X

In V2X, sensors periodically identify the physical state and generate corresponding data. Therefore, in V2X, the specific time points of data occurrence cannot be obtained; however, they can be inferred from the time interval between two event occurrences. In this paper, the time interval is used to define the time data model for the vehicular network.

**Definition 1.** The time data model in V2X consists of a time interval, whose value is a pair of sequential time points:

\[
TIME = (start, end),
\]

where \(start\) represents the start of the time interval and \(end\) represents the end of the time interval. Time is expressed as 24 hours and is accurate to the second.

The temporal relationship among time interval data has been the subject of considerable research. Compared with the classic Allen definition, seven temporal sequence relationships are established based on time periods [45]. Based on the actual demands of V2X, we adopt the simplified Allen time series relationship model shown in Figure 3.

According to Figure 3, the temporal relationship of time data can be obtained by comparing the start time and the end time. Given the two time data sets \(TIME_1 = (start_1, end_1)\) and \(TIME_2 = (start_2, end_2)\), the time sequence relationship is determined as follows:

1. **TIME\(_1\) BEFORE TIME\(_2\)**
   
   Judgment condition: \(end_1 < start_2\)

2. **TIME\(_1\) EQUAL TIME\(_2\)**
   
   Judgment condition: \(start_1 = start_2\) and \(end_1 = end_2\)

3. **TIME\(_1\) OVERLAP TIME\(_2\)**
   
   Judgment condition: \(end_1 = start_2\)

4. **TIME\(_1\) DURING TIME\(_2\)**
   
   Judgment condition: \(start_1 < start_2\) and \(end_1 \geq start_2\) and \(end_1 < end_2\)

5. **TIME\(_1\) DURING TIME\(_2\)**
   
   Judgment condition: \(start_1 \geq start_2\) and \(end_1 \leq end_2\)

6. **TIME\(_1\) STARTS TIME\(_2\)**
   
   Judgment condition: \(start_1 = start_2\) and \(end_1 \leq end_2\)
3.2. The Spatial Data Model of the V2X. The spatial data model of V2X needs to be GIS-compatible. In a GIS, the map space can be represented by a raster spatial data model, in which space is considered to be a continuous whole. In addition, the space is divided into an array of uniform ruled grid cells. Each raster is a grid that has a unique number. The number of rows and columns depends on the resolution of the system and the characteristics of the spatial entity. The basic unit in the raster data model is the grid. The points, lines, broken lines, and areas in the raster spatial data model are represented in the black parts of Figures 4(a)–4(d), respectively.

Definition 2. The V2X spatial data model is a raster spatial data model, whose values are a set of contiguous raster sequence pairs denoted as

\[
\text{LOCATION} = \{(row_1, column_1), (row_2, column_2), \ldots, (row_n, column_n)\}.
\]

This spatial data model is collectively referred to as the “driving brain.” The driving brain model can be formalised in V2X. For example, monitoring the overtaking of a moving vehicle involves judging the directions of the two vehicles. The definition of the direction relationship of the driving brain formalisation spatial data model in V2X is presented as follows:

Definition 3. The formalised driving brain raster spatial data model has seven spatial direction relations: NORTH, EAST, NORTHEAST, NORTHWEST, EQUAL, OVERLAP, and IN, where the NORTH, EAST, NORTHEAST, and NORTHWEST relationships indicate that the directional relationship of the intersection of two spatial data sets is empty. This model can fully describe the disjoint position relations for eight types of spatial data.

The EQUAL, OVERLAP, and IN spatial position relations describe the relationship between two groups of spatial data. EQUAL reflects that the two spatial data are equal. The IN relation describes one spatial data set as a subset of the other spatial data set. The OVERLAP relationship describes non-EQUAL and non-IN relations and represents the direction of the intersection. The spatial orientation relationship in this definition is shown in Figure 5.

Based on Figure 5, the direction relationship of spatial data can be obtained by comparing the grid collection relations of the spatial data model.

There are two spatial data:

\[
\text{LOCATION}_a = \{(row_{a1}, column_{a1}), (row_{a2}, column_{a2}), \ldots, (row_{an}, column_{an})\},
\]

\[
\text{LOCATION}_b = \{(row_{b1}, column_{b1}), (row_{b2}, column_{b2}), \ldots, (row_{bn}, column_{bn})\}.
\]

Here, row\(_a\) and column\(_a\) represent the row value sequence and the column value sequence, respectively, of the spatial data:

\[
\begin{align*}
\text{row}_a &= \{row_{a1}, row_{a2}, \ldots, row_{an}\}, \\
\text{column}_a &= \{column_{a1}, column_{a2}, \ldots, column_{an}\}.
\end{align*}
\]

The min() and max() function obtain the minimum value and maximum value, respectively. Thus, min(row\(_a\)) and max(row\(_a\)) represent the minimum value and maximum value, respectively, of the row values in the spatial data LOCATION\(_a\). Similarly, min(column\(_a\)) and max(column\(_a\)) represent the minimum value and maximum value, respectively, of the column values in LOCATION\(_b\).

The judgement conditions of the direction relationship of the spatial data in V2X are as follows:

\[
\text{LOCATION}_a \text{ NORTH LOCATION}_b
\]

Judgement condition:

\[
\begin{align*}
\text{max}(\text{row}_a) &< \text{min}(\text{row}_b), \\
\text{min}(\text{column}_a) &> \text{min}(\text{column}_b), \\
\text{max}(\text{column}_a) &< \text{max}(\text{column}_b).
\end{align*}
\]

\[
\text{LOCATION}_a \text{ EAST LOCATION}_b
\]

Judgement condition:

\[
\begin{align*}
\text{min}(\text{column}_a) &> \text{max}(\text{column}_b), \\
\text{max}(\text{row}_a) &< \text{max}(\text{row}_b), \\
\text{min}(\text{row}_a) &< \text{max}(\text{row}_b).
\end{align*}
\]

\[
\text{LOCATION}_a \text{ NORTHEAST LOCATION}_b
\]

Judgement condition:
Figure 4: (a) Point, (b) line, (c) broken line, and (d) areas in the raster spatial data model.

Figure 5: Continued.
\[ \text{max}(\text{row}_a) < \text{min}(\text{row}_b), \ \text{min}(\text{column}_a) > \text{max}(\text{column}_b). \]

\[ (7) \]

**LOCATION\textsubscript{a} NORTHWEST LOCATION\textsubscript{b}**

Judgement condition:

\[ \text{max}(\text{row}_a) < \text{min}(\text{row}_b), \]

\[ \text{max}(\text{column}_a) < \text{min}(\text{column}_b). \]

\[ (8) \]

**LOCATION\textsubscript{a} EQUAL LOCATION\textsubscript{b}**

Judgement condition:

\[ LOCATION\textsubscript{a} = LOCATION\textsubscript{b} \]

**LOCATION\textsubscript{a} OVERLAP LOCATION\textsubscript{b}**

Judgement condition:

\[ LOCATION\textsubscript{a} \cap LOCATION\textsubscript{b} \neq \emptyset \]

3.3. **Spatial and Temporal Event Instance Model of V2X.**

In the driving brain architecture of V2X, all data are abstracted into event instances. As the data of V2X is spatiotemporal, the event instances of the system are also spatiotemporal. V2X focuses on the object condition attribute values, such as speed, time, location, direction, status, and value. These attributes are generated by various sensor types, which are embedded in the sensing equipment. Data collection occurs for particular objects, in which an object is a spatial-temporal event instance.

**Definition 4.** The V2X time event instance contains a sequence of various sensing data of objects in the vehicular network:
where ID is a unique identification value for the object in V2X. The value can be an IP address, a QuickMark code, or an RFID tag. \( \text{TIME} \) is a time attribute value that uses the time data model in Definition 1. LOCATION is a spatial attribute value that uses the spatial data model in Definition 2:

\[
\text{ATTRIBUTE} = \{ \text{Attribute}_1, \text{Attribute}_2, \ldots, \text{Attribute}_n \},
\]

where \( \text{Attribute}_1, \text{Attribute}_2, \ldots, \text{Attribute}_n \) are the values of other types of numeric attributes, including speed, acceleration, temperature, and humidity.

Definition 4 emphasizes that the event instance in V2X must include the spatial and temporal characteristics. Thus, the time attribute values and spatial attribute values of the event instance are critical. All types of sensors continuously generate data in V2X. Thus, these basic spatiotemporal event instance sequences constitute the instance flow of the spatiotemporal events.

**4. The STEIM Model**

In a vehicular network, various objects continuously generate temporal and spatial event flows. However, the people in the vehicle generally focus only on the event instances that satisfy certain conditions. Different vehicles tend to focus on different event instances, and different users tend to focus on different events. Thus, users need to indicate their interest in processing specific events. In the STEIM architecture, the model supports event-processing selections. These event-processing selections are transformed into the corresponding event instance flow handler by event-processing middleware to complete the processing of the event instance flow. In Figure 6, the spatiotemporal event-processing STEIM model for V2X involves the following syntax collection: the model is divided into three modules: vehicular users, on-board computer systems, and a cloud database. Vehicular users transmit their vehicle information to the on-board computer system via data-processing statements. The on-board computer system uses event flow-processing rules to parse the event-processing sentences and upload them to the cloud database. Vehicular users obtain vehicle information from the cloud database for other vehicular users.

**4.1. Constant Collection.** The value of an event handler expression is the Boolean set, whose elements are represented by \( B \):

\[
\text{BOOLEAN} = \{ \text{TRUE}, \text{FALSE} \}.
\]

A collection of event instances in V2X, whose elements are represented by \( VE \), is represented as

\[
\text{VehicularEvent} = \{ VE_1, VE_2, \ldots, VE_n \}.
\]

Each event instance can contain several attributes.

\[
VE = \{ ID, T, LOC, A \}.
\]

An identity collection of event instances in V2X has elements represented by \( VE_{id} \):

\[
\text{VehicularEventID} = \{ ID_1, ID_2, \ldots, ID_n \}.
\]

A collection of time attributes for event instances in V2X has elements represented by \( VE_t \):

\[
\text{VehicularEventTIME} = \{ (start_1, end_1), (start_2, end_2), \ldots, (start_n, end_n) \}.
\]

A collection of spatial attributes for the spatiotemporal events in V2X has elements represented by \( VE_{loc} \):

\[
\text{VehicularEventLOCATION} = \{ (row_1, column_1), (row_2, column_2), \ldots, (row_n, column_n) \}.
\]

The elements of other numeric attribute values of temporal and spatial events in V2X are represented by \( VE_a \):

\[
\text{VehicularEventATTRIBUTE} = \{ \text{ATTRIBUTE}_1, \text{ATTRIBUTE}_2, \ldots, \text{ATTRIBUTE}_n \}.
\]

**4.2. Variables’ Collection.** The variable that holds the instance of the space time event is denoted by \( x \), and its variable set is represented by \( X \). The variable \( x \) is composed of the object variable \( x_{id} \), the time variable \( x_t \), the location variable \( x_{loc} \), and the attribute variable \( x_a \), and is expressed as \( x = (x_{id}, x_t, x_{loc}, x_a) \).

Each subvariable is specifically defined as follows: the variable \( x_{id} \) holds the object identification value of the spatiotemporal event instance and is represented by \( X_{id} \); the variable \( x_t \) holds the time attribute value of the temporal and spatial event instance and is represented by \( X_t \), while the \( x_{loc} \) holds the spatial attribute value of the temporal and spatial event instance and is represented by \( X_{loc} \). The variable \( x_a \) holds the other numeric attribute values of a temporal and spatial event instance and is represented by \( X_a \).

**4.3. Expression Collection.** The set of event-processing statements in the STEIM model is referred to as spatiotemporal event interaction (SEI). The elements of SEI are represented by \( sei \). SEI consists of Boolean expressions. The object identification value Boolean expression that determines the occurrence of temporal and spatial events is \( IBsei \). Its elements are represented by \( IBsei \). The time Boolean expression set is \( TBsei \), and its elements are represented by \( TBsei \). The space Boolean expression is \( LBsei \), and its elements are represented by \( LBsei \). The general numeric
attribute Boolean expression set is ABsei, and its elements are represented by ABsei.

4.4. Grammar Interaction of STEIM. The STEIM model applies to a known constant set and a variable set. A variable is usually represented by a nonnull string. In addition, variables are not empty or followed by a string.

(1) The object Boolean expression IBsei is

\[ ibsei = xid = ID, \]

where “=” denotes an “equal” relation. The time Boolean expression is

\[ tbsei = x_t \text{BEFORE} | x_t \text{EQUAL} | x_t \text{OVERLAP} | x_t \text{DURING}. \]

In this formula, BEFORE, EQUAL, OVERLAP, and DURING represent the temporal relations of time data. For example, the expression VehicleTime BEFORE (12: 20: 30, 12: 21: 10) means to judge whether the time variable VehicleTime value in the data stream of a vehicle is earlier than the time period (12: 20: 30, 12: 21: 10).

(2) The object Boolean expression LBsei is

\[ lbsei = x_c \text{EQUALLOC} | x_c \text{OVERLAPLOC} | x_c \text{INLOC} | x_c \text{EASTLOC} | x_c \text{NORTHLOC} | x_c \text{NORTHEASTLOC}, \]

where “EQUAL”, “OVERLAP”, “IN”, “NORTH”, “EAST”, “NORTHWEST”, and “NORTHEAST” are the seven spatial data relations that were previously defined. For example, the expression “VehicleLoc NORTH (985, 266)” determines whether the value of the spatial attribute variable VehicleLoc in a vehicle’s data is located NORTH of (985, 266).

(3) The general numerical attribute Boolean expressions ABsei are described as follows:

\[ absei = x_a = A | x_a > A | x_a < A, \]

where “=” denotes an “equal” relation, “>” indicates a “greater than” relation, and “<” denotes a “less than” relation. For example, the expression “VehicleV>65” returns a Boolean value that represents whether the value of the speed variable VehicleV in a vehicle’s data is greater than 65 km/h.

(4) Event-handling statement:

\[ sei = ibsei | tbsei | lbsei | absei | sei | sei | sei | sei | sei + sei. \]

The event-processing statement consists of four expressions using three operators “,” “||,” and “+.” The operator “,” is a sequence combination operator. The operator “||” is a parallel combination operator, and the operator “+” is the selection operator.

For example, the event-handling statement,

\[ VehicteId = 1253 \] VehicleTime DURING (12: 20: 30, 12: 21: 10) VehicleLocNORTH \[ (985, 358) || (VehicleV < 80) + (VehicleAcc < 15)), \]

determines whether the vehicle that sent data has an identifier VehicleId equal to 1253, whether the data in the time value of the variable VehicleTime is within the time (12: 20: 30, 12: 21: 10), whether the value of the position variable VehicleLoc in the data is NORTH of (985, 358), and whether the value of the speed variable in the data is less than 80 km/h or the vehicle acceleration variable value is less than 15 m/s².
5. Operational Semantics of STEIM

5.1. Setting Property Variables in STEIM. The value of each expression in the model is determined by the value of each attribute variable. These values are currently stored in the temporal and spatial event instance.

The temporal and spatial event state of STEIM is defined as a function of the temporal and spatial EventInstance variable set of the on-board computer system:

\[
\delta: X \rightarrow \text{VehicularEvent}.\tag{24}
\]

The object identification subvariable state is defined as the time-space event instance object identified by a function of the set \(X_{id}\) to the temporal and spatial event instance object identification value set VehicularEventID:

\[
\delta_{id}: X_{id} \rightarrow \text{VehicularEventID}.\tag{25}
\]

The time-dependent variable state is defined as a function of the time attribute value variable set \(X_t\) to the time attribute value set VehicularEventTIME:

\[
\delta_t: X_t \rightarrow \text{VehicularEventTIME}.\tag{26}
\]

The spatial subvariable state is defined as a function of the spatial attribute value variable set \(X_{loc}\) to the spatial value set VehicularEventLOCATION:

\[
\delta_{loc}: X_{loc} \rightarrow \text{VehicularEventLOCATION}.\tag{27}
\]

The state of the numerical subvariables is defined as a function of the numeric attribute value set \(X_a\) to the time-space event instance numeric value set VehicularEventATTRIBUTE:

\[
\delta_a: X_a \rightarrow \text{VehicularEventATTRIBUTE}.\tag{28}
\]

Thus, in state \(\delta\), the value VE of the space event instance variable \(x\) is expressed as \(\delta(x)\). In state \(\delta_{id}\), the value VE\(_{id}\) of the object identifier variable \(x_{id}\) is expressed as \(\delta_{id}(x_{id})\). In state \(\delta_t\), the value VE\(_t\) of the time attribute value variable \(x_t\) is expressed as \(\delta_t(x_t)\). In state \(\delta_{loc}\), the value VE\(_{loc}\) of the spatial attribute variable \(x_{loc}\) is expressed as \(\delta_{loc}(x_{loc})\). In state \(\delta_a\), the value VE\(_a\) of the numeric attribute variable \(x_a\) is expressed as \(\delta_a(x_a)\).

In the STEIM model, event instances of spatial and temporal events in the database are sent to clients. The client database of spatiotemporal events functions as a storage unit for the spatiotemporal event instance set EventInstance\(_B\):

\[
\delta_B: M_B \rightarrow \text{VehicularEvent}_B.\tag{29}
\]

In state \(\delta_{id}\), the IBSEI waits for the calculation and evaluates the results to the Boolean TRUE or FALSE, which is expressed as \(\langle \text{ibsei}, \delta_{id} \rangle \rightarrow B\). In state \(\delta_t\), the time event instance variable \(x\) results are stored in the client event database as \(\delta': \Sigma_B\), which is expressed as \(\langle \text{ibsei}, \delta: \Sigma_B \rangle \rightarrow \delta': \Sigma_B\), where \(\langle \text{ibsei}, \delta_{id} \rangle\) and \(\langle \text{ibsei}, \delta: \Sigma_B \rangle\) are IBSEI expressions.

5.2. Operational Semantics of STEIM

(1) The assignment rule for an object using the Boolean IBSEI expression is expressed as follows:

\[
\begin{align*}
\delta_{id}(x_{id}) &= 1D \\
\langle x_{id} = 1D, \delta_{id} \rangle &\rightarrow \text{TRUE}\tag{30}
\end{align*}
\]

which is the assignment rule for the expression \(x_{id} = 1D\). The variable name of the stored time event instance is \(x\), and the stored child variable \(x_{id}\) stored is the identifier value of the time event instance. If the variable \(x_{id}\) is a constant \(1D\) in the current state \(\delta_{id}\), the value of the expression \(x_{id} = 1D\) is the Boolean TRUE value. When the variable \(x_{id}\) is not a constant \(1D\) in the current state \(x_{id}\), the value of the expression \(x_{id} = 1D\) is the Boolean FALSE value.

(2) The assignment rule of the time attribute for the Boolean expression TBSEI is

\[
\begin{align*}
\delta_t(x_t).\text{end} < T.start \\
\langle x_t\text{BEFORE}T, \delta_t \rangle &\rightarrow \text{TRUE}\tag{31}
\end{align*}
\]

which is the assignment rule for the expression \(x_t\text{BEFORE}T\). The variable name of the storage time-space event instance is \(x\). In addition, the child variable \(x_t\) stores the time value of the time-space event instance. If the variable \(x_t\) is \(\delta_t(x_t).\text{end} < T.start\) in the current state \(\delta_t\), the value of the expression \(x_t\text{BEFORE}T\) is the Boolean TRUE. If the variable \(x_{id}\) is \(\delta_t(x_t).\text{end} \geq T.start\) in the current state \(\delta_t\), the value of the expression \(x_t\text{BEFORE}T\) is the Boolean value FALSE. The remainder of the time-space attribute value Boolean expressions have the same assignment rule as the previous inference rule.

(3) The assignment rule of the spatial attribute for the Boolean expression LBSEI is expressed as

\[
\begin{align*}
\delta_{loc}(x_{loc}) &= \text{LOC} \\
\langle x_{loc}\text{EQUALLOC}, \delta_{loc} \rangle &\rightarrow \text{TRUE}\tag{32}
\end{align*}
\]

which is the assignment rule for the expression \(x_{loc}\text{EQUALLOC}\). The variable name of the stored spatial attribute event instance is \(x\). In addition, the child variable \(x_{loc}\) stores the time value of the spatial attribute event instance. If the variable \(x_{loc}\) is \(\delta_t(x_{loc}) = \text{LOC}\) in the current state \(\delta_{loc}\), the value of the expression \(x_{loc}\text{EQUALLOC}\) is the Boolean TRUE value. When the variable \(x_{loc}\) is \(\delta_t(x_{loc}) \neq \text{LOC}\)
in the current state $\delta_{\text{loc}}$, the value of the expression $x_{\text{loc}} \text{EQUAL LOC}$ is the Boolean FALSE value. The remainder of the spatial attribute value Boolean expressions have the same assignment rule as the previous inference rule.

(4) The assignment rule of a general numerical attribute for the Boolean expression $ABSEI$ is expressed as follows:

$$
\delta_a(x_a) > A \rightarrow \langle x_a, A, \delta_a \rangle \rightarrow \text{TRUE}
$$

$$
\delta_a(x_a) \leq A \rightarrow \langle x_a, A, \delta_a \rangle \rightarrow \text{FALSE}
$$

which is the assignment rule for the expression $x_a > A$. The variable name of the stored spatial event instance is $x$. In addition, the child variable $x_a$ stores the number value of the spatial event instance. If the variable $x_a$ is $\delta(x_a) > A$ in the current state $\delta_a$, the value of the expression $x_a > A$ is the Boolean TRUE value. When the variable $x_a$ is $\delta(x_a) < A$ in the current state $\delta_a$, the value of the expression $x_a > A$ is the Boolean FALSE value. The remainder of the spatial value Boolean expressions have the same assignment rule as the previous inference rule.

6. STEIM Algorithm Based on Petri Net

Finite automata is a basic computer model with extremely limited storage capacity. It is a better model for small storage devices. Push-down automata (PDA) is a better model for infinite storage devices. It has an additional device stack. The stack provides additional unlimited storage, so the device is used as a “last in, first out” stack. It writes symbols on the stack, then reads and writes a symbol and “pushes down” all the other symbols in the stack. Therefore, finite automata can provide theoretical support for some limited state transformations, such as the design of automatic door controller, and push-down automata can provide context-free language recognition. They can play a role in some limited fields. But, these models have proved that some simple tasks cannot be completed, and are too limited to be used as a general model of computers. However, only by improving them on the basis of the two automata models can we get a more mature

Table 1: How the STEIM model handles the four parameter types.

| $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | $\lambda_5$ | $\lambda$ |
|------------|------------|------------|------------|------------|----------|
| Model 1    | 3          | 6          | 12         | 20         | 26       | 1.75     |
| Model 2    | 3          | 6          | 12         | 20         | 26       | 3.66     |
| Model 3    | 3          | 6          | 12         | 20         | 26       | 2.58     |
| Model 4    | 3          | 6          | 12         | 20         | 26       | 2.12     |

Figure 7: The Petri net-processing model of STEIM.
and useful automation model— the Turing machine. The
Turing machine can be defined as a seven-tuple
\((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\), where \(Q, \Sigma, \Gamma\) are all finite sets.
\(Q\) is a state set; \(\Gamma\) is an alphabet on an infinite square. Among
them, the blank symbol \(\epsilon \in \Gamma\); \(\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}\) is
the transfer function, where \(L\) and \(R\) represent their transfer
direction, respectively; \(q_0 \in Q\) is the initial state, \(q_{\text{accept}} \in Q\)
is the acceptance state; \(q_{\text{reject}} \in Q\) is the rejection state, and
\(q_{\text{accept}} \neq q_{\text{reject}}.\) A real system can be defined as a set or sum of
all entities that interact and depend on each other in accor-
dance with certain laws. The system may be natural or
artificial, existing or planned for the future. Modeling of
these complex systems needs to take into account the sys-
tem-related factors involved in system changes, including
raw materials, components, products, personnel, tools,
equipment, data and information, as well as the asynchro-
nous and parallel mode of system state changes. Turing
machine is a very effective modeling tool for simple systems
with serial algorithms, but it is much more difficult to apply
it to complex systems. Petri net is developed from Turing
machine. It provides a powerful theoretical basis and
modeling tools for complex asynchronous, parallel systems,
and modeling. Petri net theory was proposed by Carl A.
Petri, a German scholar, in 1962. Since then, this formal
description language has been widely used in many fields.
Petri net is a graphical tool used to describe the behavior of
complex systems. It is different from other graphical system
description tools (such as flow chart, logic tree), and it can
represent the parallelism, synchronization, conflict, and
causal dependence in the system. Petri net is defined as a
three-tuple \(PN = (S, T; F)\), where elements \(S\) and \(T\) are
called Place and Transition sets of \(PN\), respectively. \(F\) is
Flow-relation. It is a set of ordered pairs composed of ele-
ments \(S\) and \(T\). The necessary and sufficient conditions are
\(S \cap T = \emptyset; S \cup T \neq \emptyset; F(S \times T) \cup (T \times S)(x\) is Cartesian
product); \(\text{dom}(F) \cup \text{cod}(F) = S \cup T\), where \(\text{dom}(F) = \{xy : (x, y) \in F\}\), \(\text{cod}(F) = \{xy : (x, y) \in F\}\), which are, re-
spectively, the definition domain and value domain of \(F\).
Automata, Turing machine, and Petri net have one thing in
common: there is a state transition process within the
system, which can be transformed from one state to another
state, and can be expressed by state diagram and correlation
matrix. Petri nets are equivalent to automation and Turing
machines. Although they are equivalent, automata and
Turing machines are not competent for simulating some
complex systems with parallel, conflict, and synchroniza-
tion. Petri nets have good simulation capabilities for these
systems, and Petri net is also a good model to solve the reu-
se problem of software components.

To use the STEIM model for temporal and spatial event-
stream processing in V2X, we need to compile the
statements into a corresponding executable model.
According to the corresponding spatiotemporal processing
model, we can design an event-stream processing algorithm and
a programming implementation.

A Petri net, which is also known as a place/transition (PT)
net, is one of the several mathematical modeling languages for
describing distributed systems. A Petri net consists of places,
transitions, and arcs. Arcs run from a place to a transition or vice
versa but never run between places or between transitions. The
places from which an arc runs to a transition are referred to as the
input places of the transition. The places to which arcs run from a
transition are referred to as the output places of the transition.
Graphically, places in a Petri net may contain a discrete number
of marks, termed tokens. Any distribution of tokens over the
places represents a net configuration, which is referred to as a
"marking." In an abstract sense, when considering a Petri net
diagram, a transition in a Petri net may "fire" (execute) if it is
enabled.

In the event-stream processing model that is based on
Petri nets, the Petri net input is indicated by the place where
the event has occurred or an attribute value of one of the
basic events. The output is the storage location of the da-
tabase events. Each token in the place indicates whether an
instance of the corresponding event is represented. When an
event instance arrives, the token state in the place that
represents the event will launch the transition. The transition
result is calculated by calculating the transition function.
When the function is satisfied, it causes the transition from
the state of the input to the output until a change in the token
state in the output place is detected.

### 6.1. Spatiotemporal Event Flow-Processing Algorithm

In vehicular networking, space and time are the main functions
of the event-stream processing program. Based on event-
stream processing, the Petri net model returns the event-
stream processing results; the core algorithm is shown in this
section. In this algorithm, all transitions in the input Petri
net model are divided into four classes: TransitionStart,
TransitionMid, TransitionParallel, and TransitionEnd. Among
these classes, the function of TransitionStart is to
detect the attribute value of the received event instance. The
TransitionStart method from the shared place is connected to
TransitionMid and TransitionParallel. The method passes
information to the output place of the TransitionStart, and
the TransitionMid and TransitionParallel places share the
same variable. TransitionEnd is a special instance of Trans-
itionMid for which no subsequent transition occurs.

This algorithm, from TransitionStart to execution, fol-
lows a Petri net structure to TransitionEnd step by step. As
the number of transitions between TransitionStart and

| Model | \(M(P_i)\) | \(M(T_i)\) | \(M(W_i)\) | \(M(W_i)/M(T_i)\) |
|-------|----------|----------|----------|-----------------|
| Model 1 | \(S_i/(88 - S_i)\) | \(1/(88 - S_i)\) | \(S_i/(7656 - 88S_i)\) | \(S_i/88\) |
| Model 2 | \(S_i/(183 - S_i)\) | \(1/(183 - S_i)\) | \(S_i/(33489 - 183S_i)\) | \(S_i/183\) |
| Model 3 | \(S_i/(129 - S_i)\) | \(1/(129 - S_i)\) | \(S_i/(16641 - 129S_i)\) | \(S_i/129\) |
| Model 4 | \(S_i/(106 - S_i)\) | \(1/(106 - S_i)\) | \(S_i/(11236 - 106S_i)\) | \(S_i/106\) |
TransitionEnd in the Petri net is limited, the algorithm can be terminated for a single event instance. In addition, the number of transitions in the Petri net model in this algorithm is $n$, and the time complexity of the algorithm for a single event instance is $O(n)$.

6.2 Petri Net-Processing Model of STEIM. In the Petri net-processing model for a complex event, the input place of the Petri net indicates that the basic events and the output place of the Petri net expresses the complex events. The number of tokens in the place represents the number of event instances. When an event instance arrives, the token in the place will change. By calculating the transition function, the model can determine whether the transition conditions are satisfied. If the transition rule is satisfied, the transition will fire, and the statuses of the input and output places will change. In Figure 7, the event flow transmission rates of the event flow handlers, which correspond to the four STEIM event processing statements, are set to $S_1, S_2, S_3,$ and $S_4$. The processing rates of the event flow handlers are $V_1, V_2, V_3,$ and $V_4$, respectively.

The service intensity of the queuing system is $I = S/V$, and the sufficient and necessary conditions for a stable-state distribution of the queuing system are expressed as follows:

$$I_1 = \frac{S_1}{V_1} < 1,$$

$$I_2 = \frac{S_2}{V_2} < 1,$$

$$I_3 = \frac{S_3}{V_3} < 1,$$

$$I_4 = \frac{S_4}{V_4} < 1. \quad (34)$$

According to the random Petri net of the series structure, the network delay is calculated. We assume that the delay time required for $n$ series transitions involves $n$ independent
random variables. In addition, the parameters are the exponential distribution functions $\lambda_1, \lambda_2, \ldots, \lambda_n$. The average delay times for $n$ transitions are $1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_n$. The total equivalent delay time for the $n$ transitions is $1/\lambda = \sum_{i=1}^{n} 1/\lambda_i$.

In the same manner, the random Petri net delay method of parallel structures is calculated. We assume that the delay time required for $n$ parallel transitions involves $n$ independent random variables. The total equivalent delay times for $n$ transitions is

$$
\frac{1}{\lambda} = \sum_{i=1}^{n} \frac{1}{\lambda_i} - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\lambda_i + \lambda_j} + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \frac{1}{\lambda_i + \lambda_j + \lambda_k}
$$

$$+ \cdots + (-1)^{n-1} \frac{1}{\sum_{i=1}^{n} \lambda_i}
$$

(35)

Figure 7 shows a comparison of the event flow handler’s performance, which corresponds to the four STEIM models. The parameters for the same types of transitions are set to the same processing rate. The parameters are listed in Table 1.

The processing rate of the event flow handler is $V_i = \lambda k$. As the service intensity of the queuing system is $I_i = S_i/V_i = S_i/\lambda k$, the mathematical expectation of the number of pending events in the team system is $M(P_i) = I_i / (1 - I_i) = S_i / (V_i - S_i)$. The formation system that addresses the mathematical expectation of each event’s time is $M(T_i) = 1 / (V_i - S_i)$. The mathematical expectation of the time spent waiting for each event to be processed in a queuing system is $M(W_i) = I_i / (V_i (1 - I_i))$. The wait time ratio for each event in the queuing system is $M(W_i)/M(T_i) = S_i/\lambda k$.

To analyse the dynamic performance of the instance, we construct an application scenario in which the arrival rate of event flow $S$ changes. The processing rate of the computer is constant at $k$ (we set $k = 50$). The dynamic performance of the instance system is shown in Table 2.

Figures 8–11 show the corresponding expected values of the processed events $M(P)$, the expected value of the residence time $M(T)$, the expected values of each event wait time $M(W)$, and an event waiting time scale drawing $(M(W)/M(T))$.

When the computer processing speed is constant and the event-stream transmission rate increases, the expected system event processing slowly increases. The expected residence time of events in the system slowly increases, and the expected waiting time for each event slowly increases. As the event-stream transmission rate increases, the event waiting time ratio slowly increases. Model 1 is the most expensive model, and Model 2 involves the least amount of time spent by the four event flow handlers to process events.

### 7. Experimental Results and Analysis

To test the effectiveness of the system based on the STEIM model for V2X event-stream processing, the following

| Table 3: Values of a spatial and temporal event stream. |
|-----------------|-----------------|---------------|---------------|
| Vehicle | VehicleID | VehicleTime | VehicleLoc | VehicleV |
| VE1 | 11109 | (20:20:20, 20:30) | (998, 120) | 63.5 |
| VE2 | 11109 | (20:20:30, 20:40) | (998, 121) | 69.8 |
| VE3 | 11109 | (20:20:40, 20:50) | (998, 122) | 72.7 |
| VE4 | 11109 | (20:20:50, 20:00) | (998, 123) | 85.3 |
| VE5 | 11109 | (20:21:00, 20:20) | (998, 124) | 77.0 |
| VE6 | 11109 | (20:21:20, 20:30) | (998, 125) | 71.9 |
| VE7 | 11109 | (20:21:30, 20:40) | (998, 126) | 68.3 |

| Table 4: Reasoning results of example. |
|-----------------|-----------------|
| UserName | UserEventBase |
| User1 | $\sum_{i=1}^{3} \delta_i | \sum_{j=1}^{2} \delta_j | \sum_{k=1}^{5} \delta_k |
| User2 | $\sum_{i=1}^{4} \delta_i | \sum_{j=1}^{3} \delta_j | \sum_{k=1}^{5} \delta_k |
| User3 | $\sum_{i=1}^{5} \delta_i | \sum_{j=1}^{4} \delta_j | \sum_{k=1}^{5} \delta_k |
| User4 | $\sum_{i=1}^{6} \delta_i | \sum_{j=1}^{5} \delta_j | \sum_{k=1}^{5} \delta_k |

This theoretical derivation and derivation indicate that the operational semantics of the STEIM language are effective.
discussions address two aspects: theoretical derivation and simulation example verification. The experiment results are analysed and compared.

Figure 12 shows the experimental and test platform of the STEIM model. On the data integration and analysis platform, the data analysis is performed on the industrial personal computer (IPC) and data transmission is conducted via the LTE network.

Figure 13 shows the data test vehicle group. During the experiment, IPC was received by each device in the test vehicle on the project team. According to the output results of the STEIM model, vehicle control based on the CAN bus was performed. All tests were performed on campus and the playground.

7.1. Experimental Design. In V2X, a vehicle with an identifier of “11109” generates a spatiotemporal event instance every 1 s during the driving process. The values for the set time event variables for Vehicle include the Car logo for VehicleID, additional variables, event instance time attribute values of variables for VehicleTime, the event instance space attribute values of variables for VehicleLoc, and the event instance speed values of variables for VehicleV:

\[
\text{Vehicle} = (\text{VehicleID}, \text{VehicleTime}, \text{VehicleLoc}, \text{VehicleV}).
\]  

The temporal and spatial event instance in the instance flow of temporal and spatial events is recorded as EI. Table 2 lists the values of a spatial and temporal event stream.

Four users (User$_1$, User$_2$, User$_3$, and User$_4$) in V2X that are monitoring the vehicle, respectively. As these users are concerned with different events, they use the following four STEIM statements, which are referred to as EventInstance, to process the spatiotemporal event flow generated by the vehicle. Two spatial data sets are established:

\[
\begin{align*}
\text{ROAD}_{32} &= \{(998, 55), (998, 56), \ldots, (998, 150)\}, \\
\text{ROAD}_{65} &= \{(122, 90), (122, 91), \ldots, (122, 180)\}.
\end{align*}
\]  

The STEIM statement of vehicle User$_1$ is EventInstance$_1$:

\[
\begin{align*}
\text{VehicleID} &= 965 \lor \text{VehicleTimeDURING} \\
&\cdot (20: 00, 20: 23: 00) \lor \text{VehicleLocINROAD}_{32} \\
&\lor \text{VehicleV} > 60.
\end{align*}
\]  

The STEIM statement of vehicle User$_2$ is EventInstance$_2$:

\[
\begin{align*}
\text{VehicleID} &= 965 \lor \text{VehicleTimeDURING} \\
&\cdot (20: 00, 20: 23: 00) \lor \text{VehicleLocINROAD}_{65} \\
&\lor \text{VehicleV} > 60.
\end{align*}
\]  

Table 5: Performance analysis.

| Dimension of data report | STEIM | The scheme proposed in [22] | Traditional scheme |
|--------------------------|-------|-----------------------------|--------------------|
| Computational cost of a vehicle during data collection | 10    | 976                         | 700                | 10                 |
|                          | 20    | 1864                        | 1450               | 28                 |
|                          | 30    | 2832                        | 2220               | 41                 |
|                          | 40    | 3657                        | 2955               | 58                 |
| Computational cost of an RSU during data acquisition | 10    | 621                         | 411                | 12                 |
|                          | 20    | 1076                        | 825                | 32                 |
|                          | 30    | 1571                        | 1175               | 45                 |
|                          | 40    | 2113                        | 1586               | 58                 |

Figure 14: Computational cost of a vehicle during data collection.

Figure 15: Computational cost of an RSU during data acquisition.

Table 5: Performance analysis.
VehicleID = 965 \| VehicleTimeDURING
\cdot (20: 30: 00, 20: 33: 00) \| VehicleLocINROAD_{32} \\| VehicleV < 90.
(39)

The STEIM statement of vehicle User_3 is EventInstance_3:

VehicleID = 965 \| VehicleTimeDURINGGmx
\cdot (20: 40: 00, 20: 43: 00) \| VehicleLocINROAD_{65} \\| VehicleV > 70.
(40)

The STEIM statement of vehicle User_4 is EventInstance_4:

VehicleID = 965 \| VehicleTimeDURING
\cdot (20: 50: 00, 20: 53: 00) \| VehicleLocINROAD_{65} \\| VehicleV < 90.
(41)

7.2. Theoretical Derivation Process and Result. According to the operational semantics of the STEIM model, the results of the four STEIM statements are listed in Table 3. Two examples are selected to illustrate the derivation process.

Formula derivation 1: User_3’s event processing statement EventInstance_3 handles the processing of the time event instance EI_3 as follows.

The current temporal and spatial event instance is EI_3. In the current on-board computer system, the state of the
temporal and spatial event variable is $\delta_2$. User$_2$’s client event library status is $\Sigma_{B_2}$. The values of each subvariable in state $\delta_2$ are listed as follows: VehicleID is 965; VehicleTime is (20:50, 20:21:00); VehicleLoc is (998, 123); and VehicleV is 85.3. The following logical process is conducted:

$$\delta_2(VehicleID) = 965 \quad \langle VehicleID = 965, \delta_2 \rangle \rightarrow \text{TRUE}$$

$$\iff \langle VehicleID = 965, \delta_2 : \Sigma_{B_2} \rangle \rightarrow \Sigma_{B_2} \cup \{\delta_2\}$$

$$\delta_2(VehicleTime).\text{end} < 20:20:50 \quad \langle VehicleTime\text{DURING}(20:20:50, 20:21:00), \delta_2 \rangle \rightarrow \text{FALSE}$$

$$\iff \langle VehicleTime\text{DURING}(20:20:50, 20:21:00), \delta_2 : \Sigma_{B_2} \rangle \rightarrow \Sigma_{B_2} \cup \{\delta_2\}$$

$$\delta_2(\text{VehicleLoc}) \subset \text{ROAD}_{47} \quad \langle \text{VehicleLoc}\text{INROAD}_{32}, \delta_2 \rangle \rightarrow \text{TRUE}$$

$$\iff \langle \text{VehicleLoc}\text{INROAD}_{32}, \delta_2 : \Sigma_{B_2} \rangle \rightarrow \Sigma_{B_2} \cup \{\delta_2\} \quad \delta_2(\text{VehicleV}) < 90 \quad \langle \text{VehicleV} < 90, \delta_2 \rangle \rightarrow \text{TRUE}$$

(42)

$$\iff \langle \text{VehicleV} < 90, \delta_2 : \Sigma_{B_2} \rangle \rightarrow \Sigma_{B_2} \cup \{\delta_2\}$$

This derivation, the spatiotemporal event instance $EI_2$ time attribute value does not satisfy the event-processing statement EventInstance$_2$ time expression. Thus, in the EventInstance$_2$ calculation results of the on-board computer system, the spatial and temporal event variable for each attribute variable (VehicleID, VehicleTime, VehicleLoc, and VehicleV) value is empty, and User$_2$ client’s event library remains unchanged.

Formula derivation 2: User$_4$’s event processing statement EventInstance$_4$ handles the processing of the time event instance $EI_4$ as follows.
The current temporal and spatial event instance is EI₄, and in the current on-board computer system, the state of the temporal and spatial event variable is δ₄. Vehicle Userᵢ's client event library status is Σᵢ. The values of each subvariable in state δ₄ are as follows: VehicleID is 965, VehicleTime is (20:20:50, 20:21:00), VehicleLoc is (998, 123), and VehicleV is 85.3, and the following logical process is conducted:

\[
\delta_4(\text{VehicleID}) = 965 \\
\langle \text{VehicleID} = 965, \delta_4 \rangle \rightarrow \text{TRUE}
\]

\[
\implies \langle \text{VehicleID} = 965, \delta_4 : \Sigma_B \rangle \rightarrow \Sigma_B \cup \{\delta_4\} \delta_4(\text{VehicleTime}.\text{start} > 20:20:50 \text{ and } \delta_4(\text{VehicleTime}.\text{end} < 20:21:00) \langle \text{VehicleTimeDURING}(20:20:50, 20:21:00) , \delta_4 \rangle \rightarrow \text{TRUE}
\]

\[
\implies \langle \text{VehicleTimeDURING}(20:20:50, 20:21:00) , \delta_4 : \Sigma_B \rangle \rightarrow \Sigma_B \cup \{\delta_4\} \delta_4(\text{VehicleLoc} \subset \text{ROAD}_{65}) \langle \text{VehicleLocINROAD}_{65}, \delta_4 \rangle \rightarrow \text{TRUE}
\]

\[
\cdot \text{titlearrow} \langle \text{VehicleLocINROAD} \rangle \delta_4 : \Sigma_B \rangle \rightarrow \Sigma_B \cup \{\delta_4\} \delta_4(\text{VehicleV} < 90) \langle \text{VehicleV} < 90, \delta_4 \rangle \rightarrow \text{TRUE}
\]

\[
\implies \langle \text{VehicleV} < 90, \delta_4 : \Sigma_B \rangle \rightarrow \Sigma_B \cup \{\delta_4\} \text{VehicleID} = 965
\]

As shown by this derivation, the spatiotemporal event instance EI₄ time attribute values satisfy the event-processing statement EventInstance₄ time expression. Thus, in the EventInstance₄ calculation results of the on-board computer system, the spatial and temporal event variable for attribute variable (VehicleID, VehicleTime, VehicleLoc, and VehicleV) value is empty. Thus, vehicle Userᵢ client’s event library remains unchanged.

Based on the corresponding operational semantics, other STEIM statements are similar to the reasoning process of the five temporal and spatial event instances listed in Table 3. The set of four user events store the initial state of Σₐ, Σₚ, Σᵢ, and Σₜ. Using the spatiotemporal event instance flow theory in Table 3, the results shown in Table 4 are obtained.

7.3. Performance Analysis. We run experiments in the proposed scheme in terms of computation costs of the RSU and the vehicles and communication overhead of the RSU and the vehicles. The experimental hardware includes an Intel Core i7-7900 processor with a 3.60 GHz clock frequency and 64 GB of memory. The performance results of the proposed scheme and the relevant models are shown in Table 5.

In Figure 14, we plot and compare the computation cost of the vehicle in terms of the data dimension, which ranges from 5 to 40, during the data collection phase. Meanwhile, in Figure 15, we plot and compare the computation cost of RSU in the three schemes in terms of the data dimension that ranges from 5 to 40, during the data acquisition phase. The comparison results show that compared with the [22] scheme and traditional scheme, STEIM can greatly reduce the computation complexity of the involved entities.

7.4. Simulation Results and Analysis. Based on the operational semantics of the STEIM model and the spatiotemporal event flow processing (Algorithm 1), which are based on a Petri net, the following experiment is performed. First, we implemented the STEIM model compiler using the open source tool ANTLRWorks 1.5. ANTLR is a for language recognition that provides a framework for constructing recognisers, interpreters, compilers, and translators from grammatical descriptions that contain actions in a variety of target languages. The experimental hardware includes an Intel Core i7-4790 processor with a 3.60 GHz clock frequency and 32 GB of memory. Figure 16 shows the syntax of the STEIM model based on the ANTLRWorks tool.

In the simulation experiment, a STEIM statement is entered into the event editor interface, and a unique number is set for each STEIM statement. The statement is activated, and the number of events represented by the STEIM statement is displayed in the event query result list. The user can select the number of events to obtain a detailed listing of the events in the event query result details.
Table 6: Example simulation results.

| UserName      | UserEventBase       |
|---------------|---------------------|
| User₁         | VE₁ + VE₃ + VE₄     |
| User₂         | ∅                   |
| User₃         | ∅                   |
| User₄         | VE₁ + VE₂ + VE₆ + VE₇|

The initial state of the event library with 4 users is an empty set. The simulation results of the spatiotemporal event instance flow listed in Table 3 are shown in Table 6.

As shown in Table 6, the spatiotemporal event flow algorithm based on a Petri net is effective. A comparison of the results of Tables 4 and 6 reveals that the results after theoretically deriving the STEIM operation semantics of the model and the experimental results from the simulation program using the structure shown in Algorithm 1 are consistent and support each other. These results demonstrate that processing a vehicular networking spatiotemporal event stream based on the STEIM model proposed in this paper is effective.

8. Conclusion

In this paper, we propose the spatiotemporal operator and the complete grammar of STEIM. This approach effectively expresses the spatiotemporal information of the spatiotemporal event flow in V2X. The formal semantics of the STEIM model is used to design the operational semantics. In addition, we provide a spatiotemporal event-stream processing algorithm that is based on a Petri net model. This algorithm establishes a working mechanism for processing the V2X event-temporal and spatial event stream. In addition, we demonstrate the effectiveness of the system based on STEIM via experiments. Future work should consider protecting the security of information and ensuring the reliable transmission of event streams. As the scale of event flows in V2X increases, conducting distributed event flow processing in vehicular network environments requires further study.

Data Availability

In our research, we made experiments to verify our theory. However, the experimental data in our paper used to support the findings of this study have not been made available because the data will be used commercially.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61932012, 61871039, 61802019, 61906017, and 71804083), the Supporting Plan for Cultivating High Level Teachers in Colleges and Universities in Beijing (Grant no. IDHT20170511), the Characteristic-Disciplines Oriented Research Project in Beijing Union University (KYDE40201702), the Premium Funding Project for Academic Human Resources Development in Beijing Union University (Grant no. BPHR2019AZ01), the Beijing Natural Science Foundation (Grant no. 4184088), and the Beijing Municipal Commission of Education Project (Grant nos. KM20191147001 and KM20171147005), and the Big Data Collaborative Innovation Center for Intelligent Driving (Grant no. CYXC1902).

References

[1] P. A. Drzagbletey, J.-Y. Shim, and J.-Y. Chung, “Quarter-wave balun fed vivaldi antenna pair for v2x communication measurement,” IEEE Transactions on Antennas and Propagation, vol. 67, no. 3, pp. 1957–1962, 2019.

[2] C. Xu, X. Huang, M. Ma, and H. Bao, “A secure and efficient message authentication scheme for vehicular networks based on lte-v,” KSII Transactions on Internet and Information Systems (TIIS), vol. 12, no. 6, pp. 2841–2860, 2018.

[3] B. Di, L. Song, Y. Li, and Z. Han, “V2x meets noma: non-orthogonal multiple access for 5g-enabled vehicular networks,” IEEE Wireless Communications, vol. 24, no. 6, pp. 14–21, 2018.

[4] R.-G. Cheng, Z. Becvar, Y.-S. Huang, G. Bianchi, and R. Harwahyu, “Two-phase random access procedure for lte-a networks,” IEEE Transactions on Wireless Communications, vol. 18, no. 4, pp. 2374–2387, 2019.

[5] S. Chen, J. Hu, Y. Shi, and L. Zhao, “Lte-v: a td-lte-based v2x solution for future vehicular network,” IEEE Internet of Things Journal, vol. 3, no. 6, pp. 997–1005, 2017.

[6] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, “A survey on security aspects for lte and lte-a networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 283–302, 2014.

[7] G. Li, M. Ma, C. Liu, and Y. Shu, “Adaptive fuzzy multiple attribute decision routing in vanets,” International Journal of Communication Systems, vol. 30, no. 4, pp. 564–568, 2017.

[8] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, “Controller placement optimization in hierarchical distributed software defined vehicular networks,” Computer Networks, vol. 135, pp. 226–239, 2018.

[9] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, “Mobile edge computing and networking for green and low-latency internet of things,” IEEE Communications Magazine, vol. 56, no. 5, pp. 39–45, 2018.

[10] S. Kuhlman, H. Lu, A. Festag, J. Kenney, S. Gemsheim, and G. Fettweis, “Evaluation of congestion-enabled forwarding with mixed data traffic in vehicular communications,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 99, pp. 1–15, 2019.

[11] H. Tan, M. M. Ha, H. Labiod, A. Boudguiga, J. Zhang, and P. H. J. Chong, “A secure and authenticated key management protocol (sa-kmp) for vehicular networks,” IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9570–9584, 2016.

[12] J. Cao, M. Ma, and H. Li, “G2zhagroup-to-route handover authentication scheme for mobile relays in lte-a high-speed rail networks,” IEEE Transactions on Vehicular Technology, vol. 66, no. 11, pp. 9689–9701, 2017.

[13] M. Minelli, M. Ma, M. Coupechoux, and P. Godlewski, “Scheduling impact on the performance of relay-enhanced lte-a networks,” IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 2496–2508, 2016.

[14] G. Han, M. Guizani, Y. Bi et al., “Software-defined vehicular networks: architecture, algorithms, and applications: part 2,”
IEEE Communications Magazine, vol. 55, no. 8, pp. 58-59, 2017.

[15] S. E. Li, Q. Guo, L. Xin, B. Cheng, and K. Li, "Fuel-saving servo-loop control for an adaptive cruise control system of road vehicles with step-gear transmission," IEEE Transactions on Vehicular Technology, vol. 66, no. 3, pp. 2033–2043, 2017.

[16] K. P. M. Ashok and V. Vaidhehi, “Anomalous event detection in traffic video based on sequential temporal patterns of spatial interval events,” Ksii Transactions on Internet and Information Systems, vol. 9, no. 1, pp. 169–189, 2015.

[17] B. Liu and A. El Kamel, "V2x-based decentralized cooperative adaptive cruise control in the vicinity of intersections,“ IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 3, pp. 644–658, 2016.

[18] H. Li and C. Yixiang, “Step:a spatial-temporal event processing language for internet of vehicles,” Journal of Frontiers of Computer Science and Technology, vol. 7, pp. 959–974, 2016.

[19] C. Backfrieder, G. Ostermayer, and C. F. Mecklenbrauker, "Increased traffic flow through node-based bottleneck prediction and v2x communication," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 2, pp. 349–363, 2017.

[20] Y. Yang, Z. Wei, Y. Zhang, H. Lu, K.-K. R. Choo, and H. Cai, "V2x security: a case study of anonymous authentication," Pervasive and Mobile Computing, vol. 41, no. 11, pp. 259–269, 2017.

[21] B. Brecht, D. Therriault, A. Weimerskirch et al., “A security credential management system for v2x communications,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 1–22, 2018.

[22] Q. Kong, B. Lu, M. Ma, and H. Bao, “A privacy-preserving sensory data sharing scheme in internet of vehicles,” Future Generation Computer Systems, vol. 92, pp. 644–655, 2019.

[23] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. Van De Wetering, "Visual traffic jam analysis based on trajectory data," IEEE Transactions on Visualization and Computer Graphics, vol. 21, no. 12, pp. 2159–2168, 2013.

[24] R. Sarraf and M. P. McGuire, “A data driven approach for safe route planning,” International Journal of Applied Geospatial Research, vol. 9, no. 1, pp. 1–18, 2018.

[25] J. J. So, A. Stevanovic, and P. Koonce, "Estimating performance of traffic signals based on link travel times," Journal of Advanced Transportation, vol. 50, no. 5, pp. 786–801, 2016.

[26] J. Li, C. Shen, X. Zhu, Y. Xie, and S. A. Cummer, "Nonreciprocal sound propagation in space-time modulated media," Physical Review B, vol. 99, no. 14, pp. 144–151, 2019.

[27] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3913–3926, 2019.

[28] N. Ma and D. Wang, "Extracting spatial-temporal characteristics of frequency dynamic in large-scale power grids," IEEE Transactions on Power Systems, vol. 34, no. 4, pp. 2654–2662, 2019.

[29] Y. Zhang, D. Hou, D. O’Connor et al., “Lead contamination in Chinese surface soils: source identification, spatial-temporal distribution and associated health risks," Critical Reviews in Environmental Science and Technology, vol. 49, no. 15, pp. 1386–1423, 2019.

[30] X. Fan, B. He, and P. Brazillon, "Context-aware big data analytics and visualization for city-wide traffic accidents," in Proceedings of the International Interdisciplinary Conference on Modeling Using Context, pp. 395–405, Paris, France, 2017.

[31] G. A. Shafabakhsh, A. Famili, and M. S. Bahadori, "GIS-based spatial analysis of urban traffic accidents: case study in Mashhad, Iran," Journal of Traffic and Transportation Engineering (English Edition), vol. 4, no. 3, pp. 290–299, 2017.

[32] B. Li, W. Guo, H. Zhang, C. Zhao, S. Li, and A. Nallanathan, "Spectrum detection and link quality assessment for heterogeneous shared access networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1431–1445, 2019.

[33] B. Jin, W. Zhuo, J. Hu, H. Chen, and Y. Yang, "Specifying and detecting spatio-temporal events in the internet of things," Decision Support Systems, vol. 55, no. 1, pp. 256–269, 2013.

[34] K. Lee, L. Liu, B. Palanisamy, and E. Yigitoglu, “Road network-aware spatial alarms,” IEEE Transactions on Mobile Computing, vol. 15, no. 1, pp. 188–201, 2016.

[35] L. Zheng, D. Xia, X. Zhao et al., "Spatial-temporal travel pattern mining using massive taxi trajectory data," Physica A: Statistical Mechanics and Its Applications, vol. 501, pp. 24–41, 2018.

[36] D. Wang, H. Bao, and F. Zhang, "Clt-dnnet: effective circular traffic light recognition with a deep neural network," International Journal of Pattern Recognition and Artificial Intelligence, vol. 31, no. 11, pp. 1–15, 2017.

[37] Q. Guan, H. Bao, and Z. Xuan, “The research of prediction model on intelligent vehicle based on driver’s perception,” Cluster Computing, vol. 20, no. 1, pp. 1–13, 2017.

[38] Y. Zheng, H. Bao, and C. X. “A method for improved pe-destrian gesture recognition in self-driving cars,” Australian Journal of Mechanical Engineering, vol. 16, no. 1, pp. 78–85, 2018.

[39] C. Xu, X. Huang, M. Ma, and H. Bao, “An anonymous handover authentication scheme based on lite-a for vehicular networks,” Wireless Communications And Mobile Computing, vol. 2018, Article ID 6251219, 26 pages, 2018.

[40] X. Huang, C. Xu, P. Wang, and H. Liu, “Lns: a security model for electric vehicle and charging pile management based on blockchain ecosystem,” IEEE Access, vol. 6, pp. 13565–13574, 2018.

[41] C. Xu, X. Huang, M. Ma, and H. Bao, “Gakav: group authentication and key agreement for lite/lite-a vehicular networks,” in Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications: IEEE 15th International Conference on Smart City: IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 412–418, Bangkok, Thailand, 2018.

[42] C. Xu, X. Huang, M. Ma, and H. Bao, “A privacy-preserving and cross-domain group authentication scheme for vehicular in lite-a networks,” Journal of Communications, vol. 12, no. 11, pp. 604–610, 2017.

[43] C. Xu, H. Liu, Y. Zhang, and P. Wang, “Mutual authentication for vehicular network in complex and uncertain driving,” Neural Computing and Applications, vol. 32, no. 1, pp. 61–72, 2020.

[44] C. Xu, H. Liu, P. Li, and P. Wang, “A remote attestation security model based on privacy-preserving blockchain for v2x,” IEEE Access, vol. 6, pp. 67809–67818, 2018.

[45] J. F. Allen, “Maintaining knowledge about temporal intervals,” Readings in Qualitative Reasoning About Physical Systems, vol. 26, no. 11, pp. 361–372, 1990.