IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 July 2024, accepted 10 August 2024, date of publication 19 August 2024, date of current version 29 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3446274

RESEARCH ARTICLE

CSM-H-R: A Context Modeling Framework in Supporting Reasoning Automation for Interoperable Intelligent Systems and Privacy Protection

SONGHUI YUE[1], XIAOYAN HONG[2], (Senior Member, IEEE), AND RANDY K. SMITH[2], (Senior Member, IEEE)

[1]Department of Computer Science, Charleston Southern University, North Charleston, SC 29483, USA
[2]Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487, USA

Corresponding author: Songhui Yue (syue@csuniv.edu)

ABSTRACT The automation of High-Level Context (HLC) reasoning across intelligent systems at scale is imperative because of the unceasing accumulation of contextual data, the trend of the fusion of data from multiple sources (e.g., sensors, intelligent systems), and the intrinsic complexity and dynamism of context-based decision-making processes. To mitigate the challenges posed by these issues, we propose a novel Hierarchical Ontology-State Modeling (HOSM) framework CSM-H-R, which programmatically combines ontologies and states at the modeling phase and runtime phase for attaining the ability to recognize meaningful HLC. It builds on the model of our prior work on the Context State Machine (CSM) engine by incorporating the hierarchies, relationships, and state transitions to take care of the dynamic aspects of context. The design of the framework supports the sharing and interoperation of context among intelligent systems and the components for handling CSMs and the management of hierarchy, relationship, and transition. Case studies are developed for IntellElevator and IntellRestaurant, two intelligent applications in a smart campus setting. The prototype implementation of the framework experiments on translating the HLC reasoning into vector and matrix computing and presents the potential of using advanced probabilistic models to reach the next level of automation in integrating intelligent systems. The implementation of the framework is available at https://github.com/songhui01/CSM-H-R. Additionally, a built-in feature of privacy protection support is discussed in the application domain by anonymization through indexing and reducing information correlation.

INDEX TERMS Automation, context dynamism, context modeling, context reasoning, intelligent system, interoperability, privacy protection, system integration.

I. INTRODUCTION

Intelligent systems take advantage of context-aware computing in reasoning automation and decision-making in context-rich fields such as smart healthcare [1], [2], emergency response [3], smart cities [4], and smart transportation [4], [5]. Intelligent systems are thus empowered to adapt to the changing environment, learn from experiences, and often make decisions without explicit human intervention [6]. A broad range of technologies involved include rule-based expert systems [7], probabilistic-model-based [8], [9], and modern artificial intelligence (AI) techniques such as deep learning [10], and reinforcement learning [3], [11].

With the advancement of sensor technologies [12], [13], contextual information has evolved from spatial and temporal data to complex facts (the states of various related entities for an application). These facts can usually be calculated from multi-source information (data from sensors deployed for entities) and historical contextual data. The variety and heterogeneity bring significant challenges in the context computing of intelligent systems to fulfill different criteria, such as interoperability, consistency checking, support of

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Mehmood.

sharing and reasoning [6], [14], [15], privacy protection [2], [16], and requires academia and industry to explore various approaches or a combined approach to handling the context from different domains or applications.

A context modeling framework, CSM-H-R, is proposed in this research to address three challenges. First, as contextual data are multi-sourced, they can be synthesized and then exploited by many intelligent systems simultaneously through proper data sharing and privacy protection mechanisms, with the purpose of reducing the overall data processing effort [17]. This can be achieved particularly with the help of cloud-based architectures and distributed systems [18]. Taking a smart campus as an example, applications of Intelligent Elevator [19] can be developed. Contextual data collected within a smart campus and those applications can contribute to each other and even to larger areas like a smart city, and vice versa. There could be many intelligent systems co-existing for public services or personal interests within different sizes of areas, and they are interoperable if they work with the same level of contextual information, as illustrated in Figure 1. We refer to the contextual information that can be used directly by those intelligent consumers as High-Level Context (HLC) in this study, in comparison to raw data such as images with RGB pixel matrices or sensor data of accelerometers.

Secondly, context sharing within intelligent systems and semantic interoperability of the shared context [20] help reasoning automation in support of decision-making. Especially as data accumulation progresses and the context is prone to changes, when both happen, it is impractical to rely on human experts to identify and embed the explicit and implicit logic within the context in the design of decision-making processes of intelligent systems. Thirdly, privacy protection is one of the key factors in the success of intelligent systems [2], [16], [21].

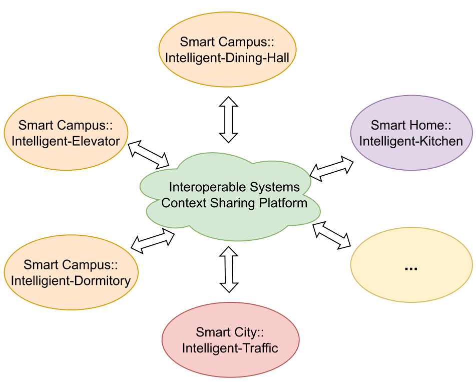


FIGURE 1. **Interoperable intelligent systems.**

CSM-H-R is based on ontology models of the static contextual information and Context State Machines (CSM) proposed in our prior research [22], where we apply a state-based approach, particularly in modeling for proactive behaviors and dynamic aspects of context changes [23]. Most of the current unimodal context modeling approaches, such as ontology and object-oriented methods, focus on the representation of the current context, like the current values of location (e.g., Home, Office, Gym), timestamp, activity, and the relationship between them [12], [24]. We proposed that explicitly putting forward states of high-level context can be beneficial in intriguing new angles of context understanding and modeling activities for dynamic state changes and proactive decision-making. Furthermore, the state, as the fundamental element of ontologies' attributes, also serves as the fundamental element of objects in object-oriented design. Based on these reflections, CSMs were devised to simulate state changes in context attributes and situations.

In summary, CSM-H-R is designed to facilitate reasoning automation through context sharing and interoperating among intelligent systems. The key aspects of our research contribution encompass the following:

1) Design of a HOSM framework CSM-H-R, which supports the sharing and interoperation of context among intelligent systems and the components for handling CSMs and the management of hierarchy, relationship, and transition.
2) The framework builds on the model of our prior work on the Context State Machine (CSM) engine by incorporating the H (*H*ierarchy) and R (*R*elationship and t*R*ansition) dimensions to take care of context dyanmism.
3) An implementation of CSM-H-R, providing two designs of input data formats to build core models for different smart campus intelligent systems, demonstrating the sharing of the context within intelligent systems and semantic interoperability of the shared context.
4) Indexing is applied to reduce the information correlation. Anonymization in transmission and reasoning is discussed in response to privacy concerns.

The subsequent sections are structured as follows: Section II presents the related research. Section III outlines the foundational concepts and hypotheses. Section IV elucidates the methodology of the framework. Section V presents the components of CSM-H-R and their interrelationships. Section VI describes the evaluation of applying the framework. Section VII is future work and discussion. Lastly, section VIII provides the concluding remarks.

II. RELATED WORK

Hossein and Aniket [25] introduce a design of a microservice software framework for implementing automation in the IoT-Fog-Cloud ecosystem for functions from data filtering and transferring to task workflow and scheduling. Elizabeth and Jeff [26] present Contextion, a programming framework for creating context-aware mobile applications. The framework is particularly designed for the rapid addition of new sensor technologies on a mobile device. Ralph et al. [27] presents ProCAKE, a generic framework for building structural and process-oriented case-based reasoning applications. It implements many syntactic and semantic similarity measures for various data types. The framework in this study promotes automation of reasoning, first emphasizing the data collection and transferring data into a format that makes the data more interoperable based on combinational ontology-state modeling so that they can work together to make predictions using various reasoning methods. The implementation of the framework experiments with basic probabilistic/evidence reasoning.

State machine, which is a lightweight, human-readable, and easy-to-parse approach, has been used for activity recognition [28], [29]. Teixeira et al. [29] present an activity-recognition system for assisted living applications and smart homes, using a lightweight hierarchy of finite state machines (FSMs) to detect actions and activities (sequences of actions). Rodriguez-Benitez et al. [28] use a mealy machine to realize action recognition in video sequences. The State Transition Data Fusion Model is proposed by Lambert [30], and it is "a unification of Sensor and Higher-Level Fusion." The model uses states and transitions to represent real-world status.

Ontology-based context reasoning approaches typically use Resource Description Framework (RDF) and Web Ontology Language (OWL) to build ontology models and then a set of reasoning rules defined to reason on them [31]. Nguyen and Choi [32] use a graph-based approach to conduct context reasoning. Machine learning techniques are frequently used for intelligent system-level context reasoning [33], [34]. Mourchid et al. [35] utilize the Bayesian Network (BN) for knowledge representation and reasoning under conditions of uncertainty. Hong et al. [36] present a recommender system for Places Of Interest (POI) using Markov Chain State Models (MCSM). The system leverages contextual information to provide more relevant POI recommendations. Our approach generalizes the usage of states, especially for modeling context, by introducing a state-based conceptual model to systematically model context and promote the computing of context using probabilistic ways such as BN, MCSM, neural networks, or basic CSM matrices.

Ramakrishnan et al. [37] propose correlation mining algorithms based on Kullback-Leibler (KL) divergence and frequent set mining that exploits correlated contexts to enable unsupervised self-learning. These algorithms help to identify alternate sources for a context and semantically describe the previously unseen contexts in terms of already known contexts. Gao et al. [38] tackle situational assessment for intelligent vehicles using the stochastic environmental model and Gaussian distributions in the dynamic traffic environment with unexpected obstacles, sensor failures, or communication losses. The collision risk under detection uncertainty is particularly assessed via the trajectory prediction in their study. The implementation of the framework in this study mainly addresses software integration and data interoperability, and the design for reasoning automation for selected intelligent-system scenarios targets basic stochastic methods for demonstration purposes. More advanced learning methods, as well as application scenarios, will be explored and evaluated in our future work. Lee and Helal [39] propose a modeling method for sensor data in smart space to instantiate and generate a context model and predict the context changes. Compared with Lee's work, our study targets a higher level of contextual data after synthesizing contextual-related information from various sources.

Hoang et al. [21] propose a privacy-preserving blockchain-based data sharing platform for the InterPlanetary File System (IPFS), a content-addressable peer-to-peer storage system, to alleviate the concerns also lie in denying of service of centralized systems and the traceability of systems through traditional access control. Mehdi et al. [40] propose a three-module framework named "Ontology-Based Privacy-Preserving," with three modules to store the privacy information using ontologies, find abnormal patterns, and provide a privacy rule manager. Arachchige et al. [41] introduce a framework named PriModChain that enforces privacy and trustworthiness on industrial IoT data by amalgamating differential privacy, federated machine learning, Ethereum blockchain, and smart contracts. Our work is built on an ontology-state-based modeling approach when dealing with systems' data and supports privacy protection through anonymization in data modeling, communication, and synthesizing. In this paper, we discuss privacy protection as the built-in feature of the proposed framework, and the detailed system design and evaluation will be explored in future work.

III. PRELIMINARIES

This section lays the groundwork for CSM-H-R by providing our observation, research hypothesis, and necessary preliminaries.

A. OBSERVATIONS AND HYPOTHESES

Our motivation for this study is based on the following observations and hypotheses.

1) We observed the limitations in the possible numbers of the attributes' states that are used with intelligent system reasoning. Even though for some entities' attributes, the data can be continuous, like the temperature value, a continuous range can be divided into different segments and make the states to be discrete. Some research about working with continuous context can be found in [42] and [43].

2) A hypothesis is that the number of meaningful states for some problem domains is even smaller and small enough to be computational. For example, after the discretization of temperature values using the rounding method, integers between −20°F and 120°F can be obtained, which represent a possible range for a habitable environment. Considering a specific area, like San Fransisco, the range can be between 32 F to 100 F. After proper scaling, states such as 'Cold,' 'Cool,' 'Moderate,' 'Warm,' and 'Hot' can be selectively retained.

3) For human-centered intelligent systems within IoT, contextual data can change frequently, and so do the decision processes. Thus, contextual reasoning logic needs to adapt to the changes accordingly.

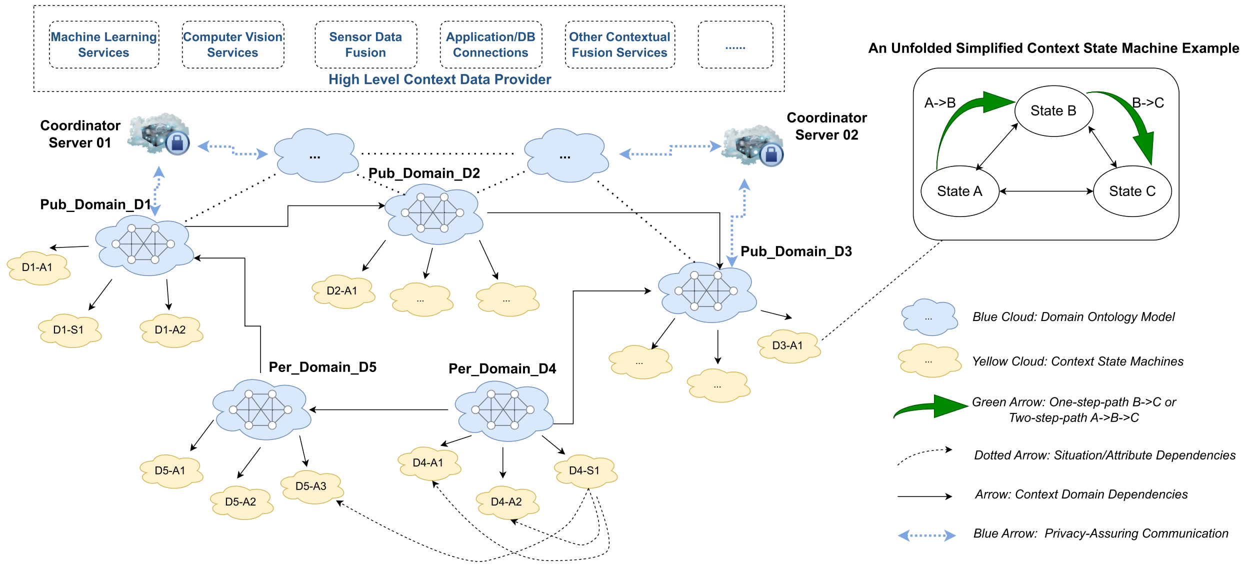


FIGURE 2. Aerial view of context sharing in conjunction with modeling methods.

4) Users and designers of intelligent systems sometimes need to understand the context and reasoning rationale. Modeling the state transitions (possibly with different scaling techniques) can help achieve this requirement.

B. CONTEXT MODELING AND CSMS

Context modeling is the core of context processing in supporting context reasoning, context sharing, and semantic interoperability of heterogeneous systems [44]. Classical context modeling techniques include key-value, markup, logic-based, ontology-based, object-oriented, and graphical modeling. Among them, ontologies are widely regarded as a highly effective solution for knowledge representation and facilitating information interoperability among applications operating in dynamic and diverse environments [20].

Context State Machines (CSMs) are based on the concept of ontologies and promote the states and state transitions. A finite state machine, more specifically, a mealy type state machine, is applied to represent the states' sequences and transitions. The two types of CSMs proposed in our prior work are Context Attribute State Machine (CASM) and Context Situation State Machine (CSSM). Figure 2 presents an aerial view of context distribution, communication, and modeling for intelligent systems of public domains (with public services related APPs) and personal domains (for applications that are customized for different personal needs).

As shown in Figure 2, each domain can administer contextual information of the entities that belong to the domain, which implicitly contains an ontology model of those entities, and each entity can have different attributes [19]. The combinations of the states of some correlated attributes and entities form various situations [12]. Each domain can have an attribute like D5-A1, which means the domain D5 has an attribute A1, and a situation like D4-S1, which means domain D4 has a situation space S1. For simplicity, entities in between the domains and attributes, as well as situation spaces, are not listed in the notations. Additionally, a simplified example of CSM can be found for D3-A1. More examples and details about building them can be found in [22].

The steps representation and coordinator servers in Figure 2 will be referred to in the following methodology sections and discussion for the topics of matrix representation and a privacy protection mechanism.

C. HCL DATA EXAMPLES

HCL data can be used as input for intelligent applications or context reasoning middleware. Vaizman et al. [45] in their research use around ten sensors to collect behavior information such as "Stairs - going up," "Stairs -going down," "Elevator," "Cleaning," and "Singing." Hande et al. [46] uses 20 binary sensors to generate 27 different activities. The same level of contextual information is used in this study, and some examples are given in Table 1.

D. CONTEXT DYNAMISM

Context dynamism in this study refers to the dynamic aspects of entities, attributes, and states and the corresponding relationship changes within intelligent system domains in the smart city setting. Below are some examples; however, additional instances could exist in real-world scenarios.

1) New Joining of Entity, Attribute, or State, e.g., a new student is joining a smart campus.

2) Relationship Changes, e.g., a student enrolls in a class provided by a professor.

3) Hot-spot Situation Forming, e.g., a student makes a decision only with the accompany of the happening of another fact.

4) Cycles and steps of transitions affecting decision-making: a student will visit a place or make a decision only after a pattern of visiting some other places.

TABLE 1. HLC data examples.

Object Category	Object	Attribute	State	Conditions/Complement	Sources
Person::Default	Adam	BloodSugar	Low	Location:::Timestamp	Body Sensors
Person::Default	Adam	Action	Open-CarDoor001	Location:::Timestamp	Body Sensors, Car Sensors, GPSs
Person::Default	Adam	Action	Driving	Location:::Timestamp	Body Sensors, Car Sensors, GPSs
Vehicle::Bus	Bus_01	Action	Running	Location:::Timestamp	Accelerometer
Vehicle::Bus	Bus_01	Speed	45mph	Location:::Timestamp	Accelerometer
Person::Professor	Bob	Action	Walking	Location:::Timestamp	Sensor
Person::Professor	Bob	Direction	South	Location:::Timestamp	Sensor
Person::Professor	Bob	Speed	5mph	Location:::Timestamp	Sensor
Person::Student	Donnie	Action	Arrives	Building003 ::: Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	location	Building003	Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	location	Building003::Corridor-Segment01	Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	Action	TakingElevator	Complement:::Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	location	Building003::classroom012	Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	Action	Leaves Building003	Location:::Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	Action	Eating	Complement:::Location:::Timestamp	GPS, Camera, History Movement Data
Person::Student	Donnie	Action	Talking	Complement:::Location:::Timestamp	GPS, Camera, History Movement Data

Notes: 1) Double colon used in the Context Category column is to denote the level relations. In the implementation of this paper, the first level is used as the category. The descendants are normally the labels of an object. 2) Triple colon is to denote a parallel relation. 3) A timestamp example: 2023-01-02 15:02:23. 4) Complement: Take "Eating," for example; its complement can be a food type or a specific food name or object. 5) A complement can refer to another context object in a different domain. For example, the complement of "talking" can refer to a Zoom meeting. Concretely, the meeting's URI can be the unique Zoom ID.

IV. MODELING

As a HOSM-based framework, CSM-H-R centers around the core components of concepts in ontology and states. This section presents this study's high-level design, including the core class diagram of CSM-H-R, encoding and privacy protection, and leveraging context hierarchy, transition, and relations.

A. A SIMPLIFIED CLASS DIAGRAM OF CSM-H-R

States in the definition of CSM are meaningful values for an attribute of an object from our prior study [22]. According to the advocator of the object-oriented (OO) approach, the overall context can be viewed as an object [47]. In the state-based approach, the universal conceptual atoms include object, attribute, and state, especially attributes are taken out of objects, and states of attributes are taken out of attributes. In this way, it preserves the best extensibility for modeling at both the concept level and programming level.

Ontology is extensively used to represent the knowledge bases in intelligent systems and AI [48], [49]. It typically include concepts, instances, properties, relationships, rules, and constraints [50], [51]. CSM-H-R is a state-based modeling approach that includes the key concepts of ontology and organizes them using a class diagram that any object-oriented programming language can implement. The mapping of the concepts in ontology and the concepts in CSM-H-R are as follows:

- Concept ≡ Context Category
- Instance ≡ Context Object
- Property ≡ Context Attribute
- Relationship ≡ Context Relationship & Matrices
- Rules and constraints can be recursively represented using objects and matrices as needed.

In Figure 3, the class diagram of the CSM-H-R core model is demonstrated. It extends the CSM core model in our prior work [19] in three aspects:

1) Adding references from "ContextAttribute" and "ContextAttributeState" to "ContextObject" so that it supports the building of hierarchies for context objects (refer to subsection IV-C4).

2) Using "Matrices" to represent both the context relationships and state transitions.

3) The utility class of "ObjectURIMapping" is used to manage the mapping between indexes of objects and the URIs.

Most classes inherit from the "Ontology" class by default. To simplify and improve readability, this inheritance is abstracted away, and the focus is placed on composition relationships and references that illustrate hierarchical relations in the class diagram.

There are different context objects in each context category (e.g., a person, a device, or an application). Figure 4 shows a real domain of objects and attributes, where "Person" is a category. "Student" and "Professor" are the occupations mapped to a nominal attribute. Each of the ordinal attributes can be related to a context attribute state machine if necessary.

B. INDEXING AND PRIVACY PROTECTION

Our current implementation uses indexes to represent the entities in the program and frequency embedding of state transitions in generating multi-dimensional matrices of CSM (refer to the next section of the lower-level design and implementation). To simplify the implementation for proof of concept, numbers from 0 to n are used as indexes for context objects, assuming there are n entities in the context domain. Future implementations may include alternative encoding mechanisms.

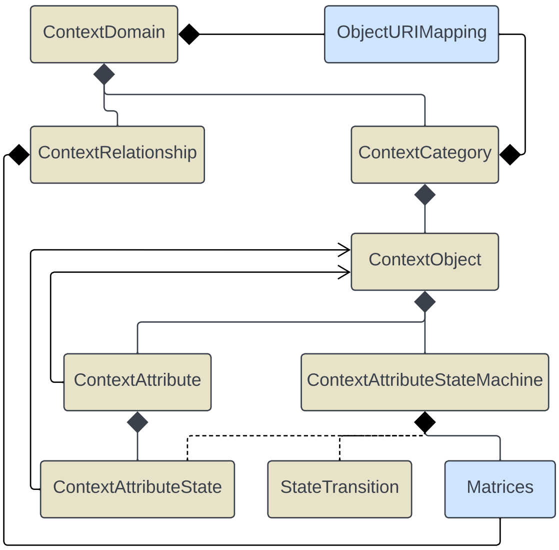


FIGURE 3. CSM-H-R core model (an extension of the CSM core model [19]).

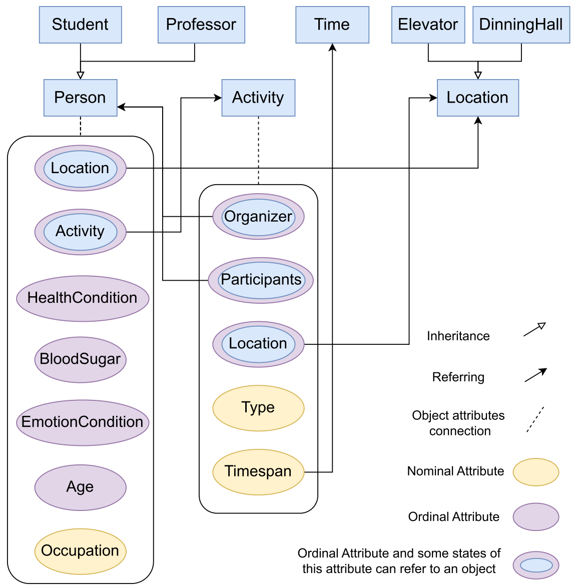


FIGURE 4. An example of a real domain of objects and attributes.

Privacy protection support is achieved by anonymization through indexing and reducing information correlation. A utility class of "ObjectURIMapping" stores a list of URIs of the entities, and each of them is mapped with its index, which is used in the CSM state transition matrices to represent the corresponding entities. A coordinator server is in charge of storing the URI and the real entity information, as shown in Figure 2. Sensitive data and their owners' information can be shipped to the target computing platforms separately to reduce information correlation. With the help of traditional encryption techniques, the chance of information interception and recovery can be reduced.

Although privacy protection is not the main focus of this work, it is still important to particularly mention this feature since it is provided through the built-in anonymization capability of the context modeling and it is a very crucial aspect that affect the broad application of data sharing and system interoperation. The evaluation of this feature is planned for future work.

C. CONTEXT HIERARCHY, TRANSITIONS, AND RELATIONSHIPS

H and R are two special letters of CSM-H-R. The letter H symbolizes the Hierarchies, and R stands for Relationships and tRansitions in CSM multidimensional state transition matrices. They are also treated as two hyperparameters when initializing the domain with the framework: H is how many hierarchies this domain is considering, and R is how many transition steps this domain is considering.

1) HIERARCHY

Hierarchy has two meanings: one refers to the reference relation between "ContextAttributeState" and "ContextObject" as shown in Figure 3, and the other refers to different granularities of context attributes in creating CASMs [22]. The H in CSM-H-R refers to the first when considered as a hyperparameter.

2) TRANSITION

A transition refers to a state change, which is also an important aspect of context information [22]. Implementing the framework in this study starts with considering 1 transition, and the number of transitions can be 2, 3, or even more, depending on the data analysis and reasoning results evaluation. A two-step transition example A→B→C is shown in Figure 2. Figure 5 demonstrates the data structure after the embeddings, where the data n in [D1.1, D2.3, D3.4] means the frequency is n to reach a state at index 4 through a state at index 1 and a state at index 3.

CSMs are graph representations of states and transitions, which means there could be circles if the transition number is more than 1. A circle number with its ID can be the next level of transition granularity.

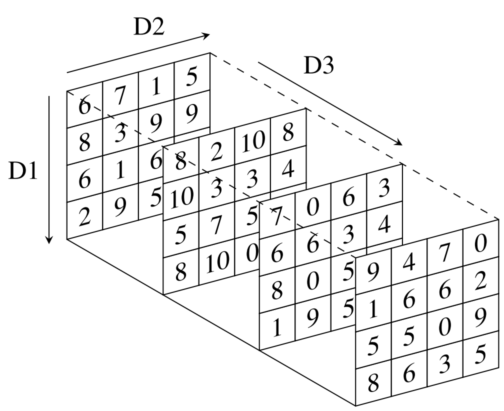


FIGURE 5. A 3-dimensional CASM matrix.

3) RELATIONSHIP

A person's situation could include the friends', colleagues', or other related entities' current states. So, to extract a person's situation, it is necessary to have a data structure to store the relationship information. In our design, a relationship matrix is used to store this type of information. Firstly, it has an array of persons, where each person is assigned an index used as the coordinate in the matrix, along with an array of relationship types. A 3-dimensional matrix is employed to organize the data, where the first and second dimensions represent the persons' indexes, and the third dimension corresponds to the relationship types.

4) RELATIONSHIP AND HIERARCHY

Relationships and hierarchies together can provide richer context than simple text information. Relationship refers to the class relationships or entity relationships in Ontology concepts. It involves the start of a relation (relation registration), updating a relation, and deleting/disabling a relation.

Relationships can be extracted from relational databases and human inputs or automatically identified through correlations such as co-location and co-timing. Additionally, relationships can also be automatically identified through mining with the hierarchies. An illustration of this process using hierarchical mining is provided below:

A person is going to a building; an intelligent system is to reason whether the person will take an elevator. The hierarchy structure is presented in Figure 6. Suppose the person object is Person001, the attribute is location, the building is Building001 as a state of the location, and Building001 contains a lab called Lab001. According to the CSM-H-R core model, Building001 is an object referred to by the state. Lab001 is an attribute of Building001, and it has a label called owner, or an attribute of a user list, and the label or this list refers to the person Person001. There is a strong bi-directional relationship observed. The state of Lab001 will influence the person's decisions, resulting in higher weights assigned to Lab001 for related decision-making. Further evaluation will enable adjustments to these weights to refine the model.

After transferring the contextual data into objects within an object-oriented program, the data stored in CASMs and CSSMs can be used to formulate additional matrices, such as those for decision-making, to support context reasoning. Contextual data are transferred into numerals in objects and CSMs, so algorithms to reason on the numeral data can be designed based on fuzzy logic, probabilistic logic, or rules.

V. CSM-H-R: SUPPORTING CONTEXT SHARING, REASONING, AND PRIVACY PROTECTION

This section presents the components of CSM-H-R, as shown in Figure 7. In order to support context sharing and interoperability, the framework is built with a message broker, which consists of channels for context and message tranmission and connecting the intelligent systems and the other context processing components shown in yellow rectangles.

A. CSM ENGINE

The CSM engine was primarily introduced in our prior research [19], and it is designed and developed to let

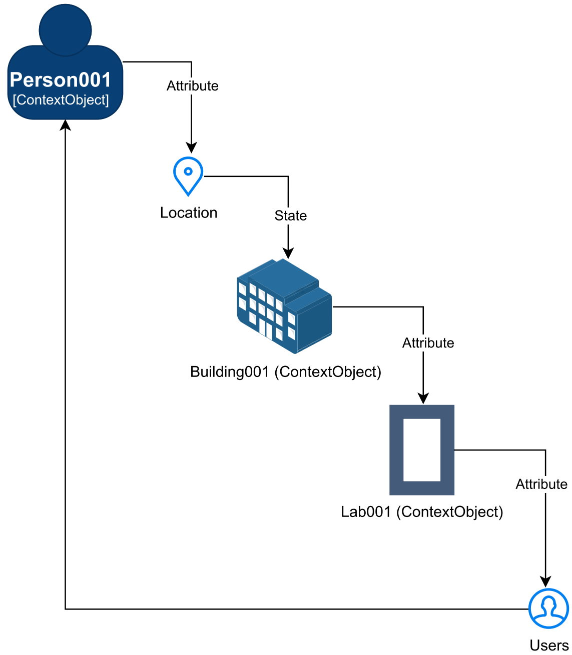


FIGURE 6. An example of hierarchy discovery.

the contextual data live in the states and context state machines. It contains a "core" module, highly related to the state-based modeling of contexts. It defines the basic classes for the concepts in context state machines, including context domain, context category, context object, context attribute, context attribute state, etc. Matrix is the fundamental computing paradigms in this implementation. State transition matrices are used to represent the CASM, CSSM and the relationships.

To facilitate the data processing for building and configuring CASMs, two types of triple are implemented as conversion data formats. One is Triple-H-R, a triple of "object," "attribute," and "state," with complements and conditions as shown in Table 1. Algorithm 1 is designed to transfer this type of triples into CSM/States representation. In order to simplify the content of the pseudocode, object-oriented language style representation is used if the notations are self-explanatory or easy to comprehend. The other is an extension of the Triple concept from the Resource Description Framework (RDF) [19]. A Triple in RDF is composed of a subject, a predicate, and an object. Meanwhile, an extended triple Triple-RDF in our framework includes decisions and conditions. The condition can contain several triples to represent the condition details. Algorithm 2 presents the process of relationship registration using "relationMatrix" and "relationClosenessMatrix." As shown in the algorithms, iterations are widely used to search for some related objects in the two demonstration algorithms. Methods that lead to more efficient searching and updating can be applied in the future to further improve the implementation efficiency. The style of the two algorithms represents the general idea of how to transform data from different formats into the same numeral state-based representation, and thus, the framework

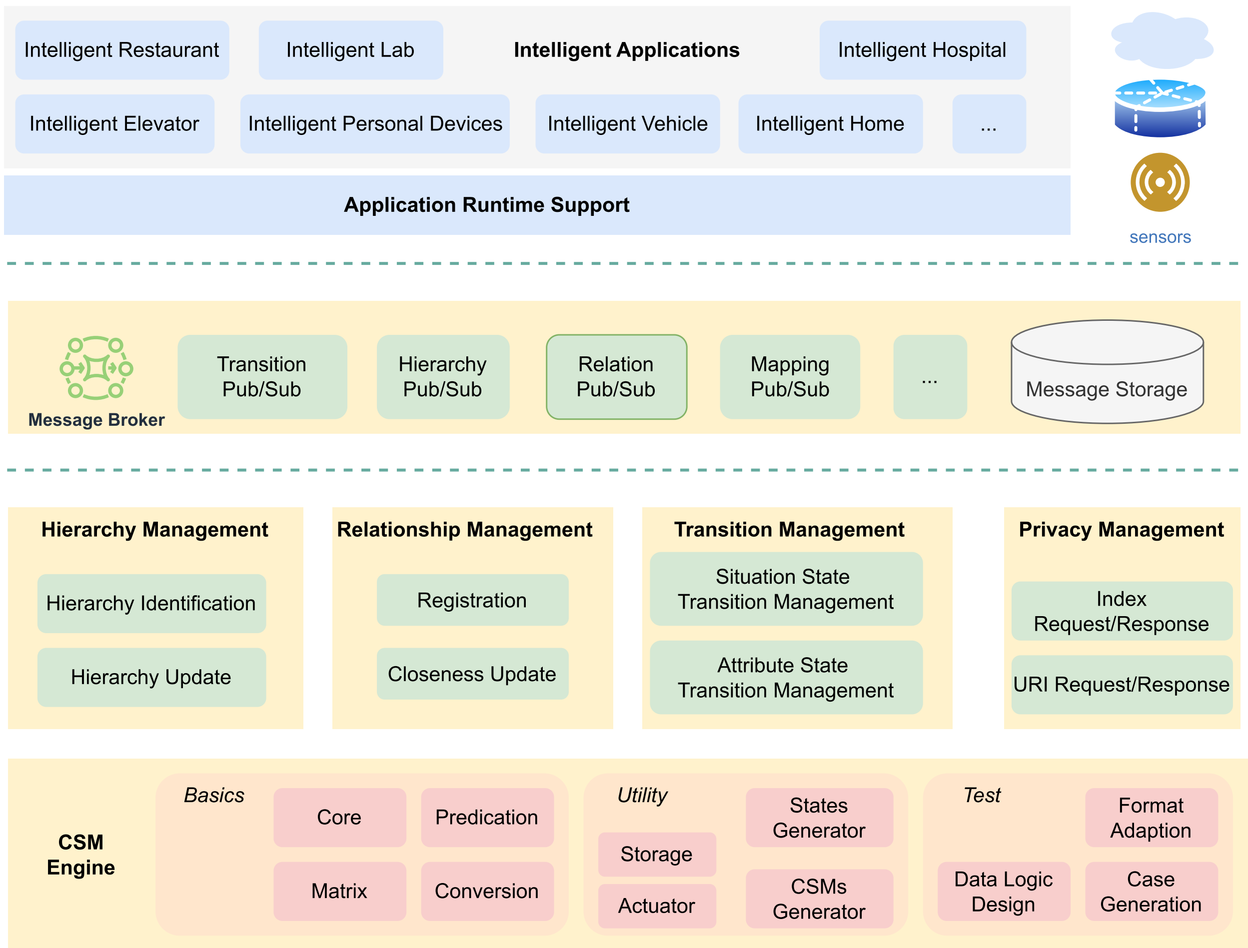


FIGURE 7. The CSM-H-R framework for interoperable intelligent systems and privacy protection.

supports the interoperability that is required when dealing with heterogeneous systems.

The prediction module is designed to contain context reasoning-related classes and attributes such as context reasoning function and threshold. It also includes internal methods to run the CSMs' context reasoning functions. Matrix storage and retrieving are convenient using the JSON-related packages of the Java language. However, processing the matrix data for particular purposes depends on the algorithms that need to be designed and explored. This following subsections introduces the basic matrix computing methods to calculate the probabilities of state transitions and decision-making. Quantile-based decision functions and thresholds for common cases or different scenarios can be developed based on these methods.

Matrix computing can be applied when intelligent systems use translated numerals to compute and make context reasoning. For example, one CASM is put into a matrix, and trigger functions can be put into a different matrix. CASM matrices can be divided into different types according to the time range of states. The first type is for all states in history, the second is for one week, the third is for one day, and the last is for the most recent. This way, the related data are changed into integers, significantly saving storage space.

1) CASM MATRIX REASONING

States are used to model context. Meanwhile, putting forward the states as the first citizen of modeling supports the calculation of probabilities through logic in the granularity of states and state transitions. For example, to calculate the probability of transiting from a state S_x to a specific state S_y in a CASM (suppose the total number of states is n, and the CASM matrix name is M; M_{xy} refers to the value of the spot of row x and column y), the formula of the probability is:

$$P_{x->y} = M_{xy} / \sum_{i=0}^{n} M_{xi} \tag{1}$$

A "decisionMatrix" is used for context reasoning about which decision a person will most probably take. The third dimension of the matrix is the decision types. To calculate the probability of a decision type is chosen when the entity transit from state S_x to state S_y (suppose the total number of decision types is d and the decision matrix name is

Algorithm 1 An Iterative Algorithm to Transfer Triples Into CSM/States Representation

```
input  : A list of triple objects TO and the initilizaed
         context domain object CD
output: An updated context domain object CD with
         CASMs and CSSMs assembled
cc ← NULL ;          /* context category */
co ← NULL ;            /* context object */
col ← NULL ;    /* context object list */
foreach triple object t_i ∈ TO do
    foreach context category object cc_i ∈ CD do
        Check if the cc_i has the same type of the object
        t_i;
        if the cc_i has the same type of the object t_i then
            cc ← cc_i;
        else
            continue;
    if no category is finally identified then
        functionToHandleErrorOrCategoryRegistration();
    Initilize a new object co_temp using t_i's object
    information;
    co ← co_temp;
    o_exist ← false;
    foreach context object co_i ∈ cc_i.col do
        if co.URI == co_i.URI then
            o_exist ← true
    if o_exist then
        Search the context object, attribute etc. and
        update them using the current data ;
    else
        // The object is a new one.
        addNewStateToRelationshipMatrix();
        cd.col.add(co); cc_i.col.add(co);
        Initilize a new context attribute ca using t_i's
        attribute information;
        Initilize a new context attribute state cas using
        t_i's state information;
        if cas is not in a maintained context attribute
        state list caslof the ca then
            ca.casl.add(cas);
        Initilize a new context attribute state machine
        casm using a default constructor;
        handle hmstates of casm to update the map of
        the states;
        handle matrixes of casm to update the casm;
        co.add(casm);
        handle conditions or constraints;
```

D; D_{xyz} refers to the value of the spot of (x, y, z)), the formula is:

$$P_{z|x->y} = D_{xyz} / \sum_{i=0}^{d} D_{xyi} \tag{2}$$

Algorithm 2 An Algorithm to Register Relationships

```
input  : A list of triple objects TO and the initilizaed
         context domain object CD
output: An updated context domain object CD
closeness ← 0;
relationName, relationURI, subjectURI, objectURI;
foreach triple object t_i ∈ TO do
    relationName = t_i.predicate.predicateName;
    relationURI = t_i.predicate.predicateURI;
    relation_exist = false;
    foreach RelationPair
    rp ∈ CD.contextRelation.relationTypeList do
        if relationURI.equals(rp.relationURI) then
            relation_exist ← true;
            break;
    if !relation_exist then
        CD.contextRelation.relationTypeList
        .add(newRelationPair(
        relationName, relationURI));
    index = CD.contextRelation
    .relationTypeList.indexOf(rp);
    subjectURI = t_i.subject.subjectURI;
    objectURI = t_i.object.objectURI;
    closeness = t_i.object.closeness;
    x = CD.col.indexOf(subjectURI);
    y = CD.col.indexOf(objectURI);
    CD.contextRelation.relationMatrix
    .get(x).set(y, index);
    CD.contextRelation.relationMatrix
    .get(y).set(x, index);
    CD.contextRelation.relationClosenessMatrix
    .get(x).set(y, closeness);
    CD.contextRelation.relationClosenessMatrix
    .get(y).set(x, closeness);
```

The obtained probabilities can then be decided to be "Obvious" or not by comparing them with a decision threshold like "0.8". If a probability is "Obvious," then the related decision would probably be taken by the examined person or other entities.

2) USING RELATIONSHIP MATRIX TO EXTRACT SITUATIONS AND BUILD CSSM

For building a person's CSSMs, the system needs a mechanism to find the current situation and update the situation. The relationship matrix can be used to extract situations and build CSSMs. A situation comprises the person's states and last states (transitions), the person's related entities' states, and the last states (transitions). In order to obtain the information of the latter, the system will process all triple inputs. For each triple, the system will traverse the relations to see if the triple's subject has a relationship with the person. If so, the system will add the "subject," its state, and the latest transition to the situation. For example, a person entity P1's location attribute can maintain a current state and a last state. Further information about the person's situation can be obtained from P1's relationship matrix: a list of related persons can be identified and iterated to get their states and last states or transitions.

If there are too many related entities for one entity, the computing process could be refined in the future by focusing only on the situations of the closest entities. Closeness could be defined based on factors such as the time entities spend together or other relevant features.

The CSM engine maintains an array of situations. When a new state of a situation is introduced, a new number is assigned to the situation list, and the new number maps to the new situation. This way allows for constructing a CSSM using a mechanism similar to that used for building a CASM, with the probability calculation for transitioning from one situation to another being analogous to the process within a CASM.

3) CSSM DECISION REASONING

Since the recommended decision is related to a person's activities, and probably that the persons' activities are implicitly related to their related entities' status, the relationship between a person's situations and their decisions can be explored. This forms the basis for decision reasoning using CSSM. To calculate the probability of a decision type is chosen when the entity transit from a state S_x to a specific state S_y in a CSSM, the method is similar to Formula 2 (Suppose the decision matrix name is SD):

$$P_{z|x->y} = SD_{xyz} / \sum_{i=0}^{d} SD_{xyi} \tag{3}$$

B. UTILITY AND TEST

The utility module contains the code file that can be used to provide utility functions for initialization, generation, storage, and testing. Particularly, the "State Generator" module is designed to load data from various sources. Then, the data can be used for object, attribute, and state extraction. The "CSM Generator" module uses objects, attributes, and states' information to build CSMs for context objects. The "Actuator" module can receive CSMs from or distribute to other servers. It also contains functions for "file" to "machine" (CSM object) and "machine" to "file" transformation. The "file" can be in XML format or from database tables.

One important task in developing the framework prototype is testing it using mock data, which should be generated according to defined cases and data logic. The data may need to be updated to a format the utility functions can process.

In our implementation of storage functions, JASON files are used to store the CSM data. An object contains all the data for an entity, including the attributes, the states of those attributes, and the state transition information. Due to its lack of a fixed schema, it aligns well with NoSQL databases.

C. HIERARCHY MANAGEMENT

Hierarchy management requires a submodule to identify the hierarchies. It happens when a new entity joins the ontology graph of an intelligent application, an attribute is updated, or a new state of an attribute emerges.

The Hierarchy Update submodule is responsible for sending the newly identified ones to other modules that utilize the hierarchy relationship, such as the Relationship Management module. It also receives messages through the broker to collect changes in the hierarchies by calling the hierarchy identification submodule.

D. RELATIONSHIP MANAGEMENT

The relationship types and related entities, such as a friend in a system, are from registration and identification processes; the registration refers to the relations extracted from databases or other sources, and identification refers to an automatic method of relation finding through correlations such as co-location or co-timing (refer to subsection IV-C4). At the implementation level, if a new person is encountered in the context, the registration process can register the person in the person array. Similarly, if a new relationship type is discovered in the context, the process can register the relation type in the relation type array.

The closeness of the relations can be updated according to the types of relations, the information from hierarchies, and the transitions. In our implementation, a number from 0 to 100 is used to represent the closeness, and when the number is updated to 0, it can mean that the relationship is disabled.

E. TRANSITION MANAGEMENT

Our study categorizes two types of state machines for situation and attribute, and the situation is described by a composition of attributes from one or more objects. The transitions of the states can represent essential changes in the systems.

State transition identification can include two parts: building a new type of transition and identifying the occurrence of a real transition. New transitions can be added when new states are added to CSMs. When an object's attribute states are changed, the real transition happens.

Other types of transitions involve steps and circles. If a circle emerges, a new type of transition can be created. The meaning of that type of transition may or may not have a special meaning; however, if some special meaning has been identified, then the mapping between the transition type and the meaning will be stored and maintained by this module.

Besides, state transitions are depicted using the matrix representation. When a number in a matrix changes and a threshold of that number has been registered to trigger some action, this module will monitor the messages for that number in the message broker. Thus, one transition may trigger some actions in different intelligent systems.

F. PRIVACY PROTECTION

To protect privacy, domain-specific IDs from indexes enable this framework to hide details of a Person's Identity (PI) when building behavioral patterns. Since all the states are indexed and those indexes are used when making context reasoning, modeled data and sensitive information can be decoupled at the physical level. For example, they can be transmitted through a different HTTP request or other types of requests or even passed through a USB device. Beyond the encryption of transmission data, this separation mechanism adds another level of protection by design if the CSM-H-R is applied.

G. CHANNELS IN THE MESSAGE BROKER

We abstract a message broker layer to show the data types of the framework's data exchanges. A message broker may be the best practice, but if the system is lightweight and there is not a demanding requirement for decoupling, scalability, and fault tolerance, an implementation without a broker may also be used.

Modules in the below components can send messages to each other through the broker to decouple message handling, and they can also receive new context updates from applications, sensors, databases, and other cloud devices.

H. APPLICATION RUNTIME SUPPORT

This part mainly contains the adaptors. The runtime support can be in the devices that run applications or an independent process of a server serving many applications. If a new transition happens, the applications can take some actions according to the changes in context, possibly by triggering registered call-back functions.

VI. PRIMARY EVALUATION OF THE FRAMEWORK IMPLEMENTATION

The framework is implemented in this study using a general-purpose language. Among the components presented in Figure 7, we implemented the modeling and some fundamental functionalities, including the CSM Engine, the basics for supporting hierarchy discovery, relationship management, and privacy. We will continue the implementation of other essential functionalities in our future work. The code is maintained at https://github.com/songhui01/CSM-H-R.

Our prior study showed that the modeling could support context reasoning for system decisions in both basic and complex scenarios [19]. The evaluation in this study involves partial core functions supported by the framework. Firstly, CSMs were built for two intelligent systems, namely IntellElevator (Intelligent Elevator) and IntellRestaurant (Intelligent Restaurant), within a smart campus setting. Secondly, execution time was evaluated across various data levels. Thirdly, the compression rate of CSMs was compared with DEFLATE, a lossless compression method.

A. CONTEXT MODELING FOR INTELLELEVATOR AND INTELLRESTAURANT

The framework is a context modeling framework. For evaluation purposes, two sets of HLC data are designed: one for IntellElevator and one for IntellRestaurant. These two sets use different formats, and the processing logic is tailored to each format accordingly. So currently, our implementation of the framework supports two types of input format (even though the format of input still means that the input should follow a specific protocol). The input of different formats can be transmitted to one unique form in model representation, which can be further used for reasoning as presented in our prior study [19].

1) INTELLELEVATOR

A utility script is used to generate random location changes and decision-making for 50 people in a smart campus setting. Initially, a location is set for each person. An analogy is that 50 people were born within those five buildings, including Home, GYM, DINNINGHALL, COFFEESHOP, and an academic building called the Science and Engineering Complex (SEC).

A number of records can be generated using the format "index, person ID, person name, person type, date, decision, action, action URI, action type, location URI, location name, location type." These records are transformed into tripleRDF records, which will be further processed into CASMs and CSSMs within a context domain object.

2) INTELLRESTAURANT

When working with a different input format of Triple-H-R, a triple of "object," "attribute," and "state," a utility script is developed to generate data in that format. Then, using those generated data records, a context domain object was created along with CASMs, with an emphasis on presenting the enhancement of usability by supporting another format of input. Our future work will connect the models generated by those two formats by carefully designing the correlations among the data.

The data generated for IntellRestaurant follows more complex logic. In one of the test cases, 2,000 students and 500 professors are generated as entities, with 23 store names and 10 restaurant names used as location states. In addition to the location attribute, six other attributes are designed: health, blood sugar, emotion, age, gender, and occupation. The activity attribute state is randomly generated from five lists of options, such as "work activities," which includes "Meeting," "Coding," "Planning," "Presenting," and "Researching."

B. EXECUTION TIME

Our prior work proves that the execution time of the modeling and the reasoning process is within a time budget that an intelligent system in literature can tolerate. A Trace-driven simulation was conducted to show that the execution time is short enough for the CPT module to respond to the ICU module of IntellElevator [19]. As shown in Figure 8, the simulation feeds mocked data into the CSM engine through the Triple-RDF format and uses different amounts of contextual data to generate the time consumption results.

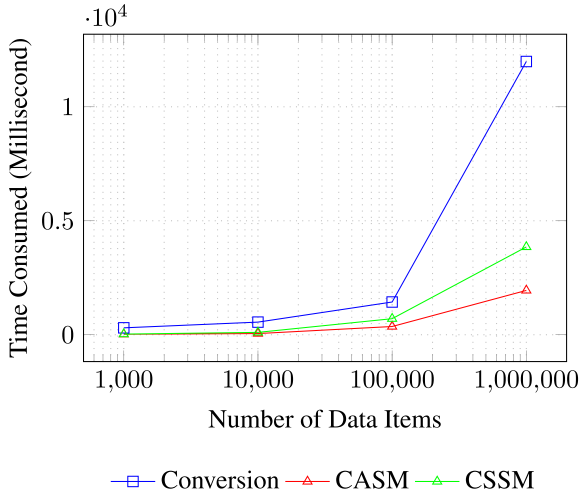


FIGURE 8. Execution time for data conversion, building CASM and CSSM for IntellElevator.

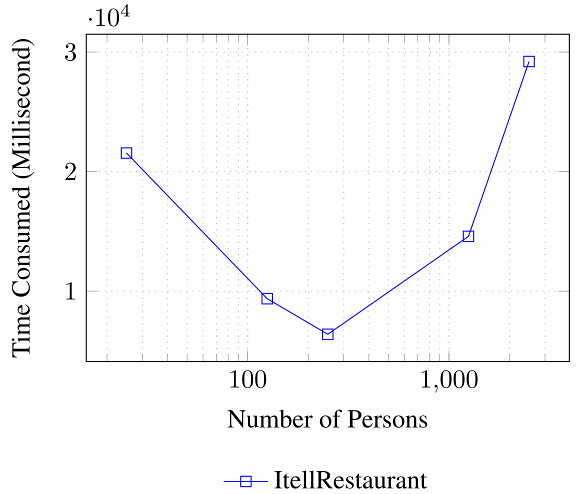


FIGURE 10. Execution time for building CASM for IntellRestaurant w.r.t. number of context objects.

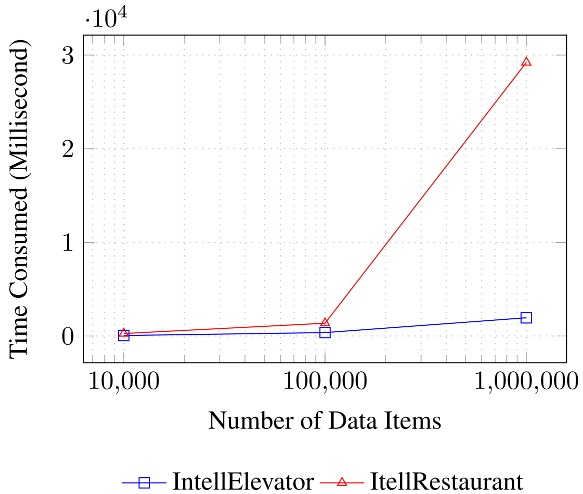


FIGURE 9. Comparison of execution time for building CASM.

This study further the examination of the execution by comparing the running time for generating CASMs for IntellElevator and IntellResturant, as shown in Figure 9. Generating CASMs for IntellRestaurant, when the number of entities changes from 25 to 125, 250, 1250, and 2500, the shape of time used is a convex function, as shown in Figure 10. The three evaluation figures presented in this subsection apply a logarithmic scale for the x-axis, where it grows slowly at first and faster when the number is larger.s

The execution environment was a commercial computer with an "Intel Core i7 CPU @2.60 GHz" using Eclipse with a heap size limit of 8.00 GB. Valuable insights can be gained from the execution results:

1) The most relevant finding endorsing the implementation's fitness for real-world usage is that when the data is within 100k level, the time spent is within 10 seconds in total for the processes of data generation, conversion, and CSMs construction, as shown in Figure 8.
2) More execution time will be consumed when the entities have more attributes and states to process from comparing the CASMs building for IntellElevator and IntellRestaurant, as shown in Figure 9.
3) When the number of total data items stays the same, the execution time does not necessarily increase linearly when the number of entities increases, as shown in Figure 10, which suggests the depth of various matrices will affect the efficiency and can be studied for special optimization purposes.

C. DATA COMPRESSION

Data compression rate can greatly affect the transmission performance in exchanging context information among processes in both local or distributed system settings.

The data compression result compared with the DEFLATE algorithm is shown in Table 2. DEFLATE is a lossless compression method that uses a combination of LZ77 and Huffman coding [52] and is used in formats like ZIP, PNG, and HTTP compression. Our intention is not to directly contrast the compression results with the optimal algorithms in terms of compression ratio. Instead, we aim to provide an overview of how the CSM engine can alleviate the transmission burden by reorganizing the data information.

CASM meta file and CASM compressed file, as the output of data transformation through CSM Engine, can be used to rebuild the original information. The ratio of raw output-file/input-file is 13.41, which is larger than 2.13, the ratio of ZIP/uncompressed. However, after the compression of the CASM meta file and CASM compressed file using DEFLATE, the total size (346KB + 239KB) is smaller than the size of the file compressed from the input file (1186KB) only using DEFLATE.

VII. DISCUSSION AND FUTURE WORK

Core concepts of the framework and some functionalities are ready to be used as part of the integration and reasoning

TABLE 2. CASM compression compared with DEFLATE compression in processing data for 2000 students and 100 professors in the intelligent restaurant application.

	Input File	Output Files		Ratio (Output Files/Input File)
		CASM Meta File	CASM File	
Uncompressed	55669KB	5951KB	1512KB	13.41%
After Zip (DEFLATE)	1186KB	346KB	239KB	49.33%
Ratio (ZIP/Uncompressed)	2.13%	5.81%	15.81%	

support for a context-sharing platform or middleware. This paper mainly presents the overall design of the framework based on our proposed CSM-H-R model. The fundamental modeling and functionalities are implemented. For future development, the framework will be enriched with additional features, including a crafted API, functions supporting hierarchy discovery, transition and relationship management, privacy protection, and various encoding methods to accommodate machine learning and deep learning algorithms.

A. STRENGTHS AND LIMITATIONS

Besides expressing the essence of context, facilitating context reasoning automation, and supporting privacy protection, our design and implementation of CSM-H-R in this research demonstrate benefits in four extra specific areas.

Firstly, numerals are used in this research to represent the textual data and object relationships. This way, plain JSON files can be used directly to store contextual data, requiring less storage space than a relational database. This approach can also specifically help machine-to-machine communication [10] by minimizing the representation of data: using numerals to represent everything and degrade the data sizes when using different network protocols.

Secondly, the implementation of the CSM engine is designed to be extendable in adding entities and constraints of different system domains for supporting even more complex context-aware scenarios. Taking care of context dynamism is discussed in the next subsection.

Thirdly, data integration from multiple sources can be achieved through first recognizing the entities using ID and URI, and then the data for a specific entiy can be merged or split as required. Related data merging and splitting algorithms need to be developed in the future.

One disadvantage is due to the vector/matrix representations, and a generated model can lack the semantic details of the data. So, to use a model made of attributes with different granularities, a new data model-building request will need to be initiated. The second disadvantage is the possible states' combinational explosion issue, given the granularity mechanism is not applied, and the meaningful states are not recognized effectively.

Additionally, this paper's main objective is context modeling and framework design. Although case studies are developed for two intelligent applications in a smart campus setting to evaluate the work, more real-world case studies remain pivotal in substantiating the applicability of the framework. Privacy protection is a built-in feature of the proposed framework; however, it is at the design and discussion level in the current phase of the work, and more detailed design and evaluation will be implemented in the future.

B. TAKING CARE OF CONTEXT DYNAMISM

The design and implementation of the framework in this study automate the handling of related context dynamism within intelligent systems in the smart city setting.

1) New Joining of Entity, Attribute, or State: A related object will be created or updated as long as a new one is identified.

2) Relationship Changes. The change of the states will be captured during the processing of the data, and the closeness of the related relationship can be updated.

3) Hot-spot Situation Forming. When a new situation forms, it can be captured in the model for making decisions related to an intelligent system.

4) Cycles and steps of transitions affecting decision-making. Steps of transition are designed as a hyperparameter of the model to reflect the handling of this type of dynamism. Implementation of transition circles is in the future work.

By modeling the dynamic aspects of context and enhancing interoperability, stochastic models and various machine-learning techniques can be adopted. As a result, predictions can be made without designers of intelligent systems to fully master the dynamisms of the entities and events, and the reasoning process only pays attention to the suggestions of the generated mathematical models.

C. CSM-H-R AT SCALE

Although an effective data representation in the IoT area can serve as the backbone of the data for machine learning and data mining techniques usage, one of the concerns of the CSM approach when involving big data is that it can occupy excessive storage and computing resources if there are too many entities and states, or considering extravagant hierarchies and transition steps. However, if the intelligent system is not time-sensitive, the CSMs can be built or updated asynchronously when the server is idler to save computing resources, and the data can be archived after CSMs are built. In addition, the distribution of related tasks to different servers in the cloud can resolve computing bottlenecks and improve responsiveness. The approaches to alleviating the CSM engine server pressure, such as parallel and distributed computing, pends exploration.

D. EXPLAINABILITY

We argue that the framework proposed in this study can provide explainability of context reasoning by supporting a gray-box [53] learning paradigm: on the one hand, through index and frequency embeddings, the reasoning of context has been transformed to numeral computing, and the reasoning engine does not need to know the meaning of the states. Therefore, the usage of numerals helps in the interoperability and sharing of context during context reasoning. On the other hand, the states are mapped to meaningful ontologies, which gives it the potential to reveal the logic of decision-making as one direction of our future work.

E. GENERALIZATION

Theoretically, CSM-H-R, as a modeling and automatic reasoning approach, can be utilized not only with HLC but also with lower-level knowledge as long as objects, attributes, and states can be recognized. Constraints may vary compared with the HLC-based study in this research and pends exploration.

VIII. CONCLUSION

In order to facilitate the building of interoperable intelligent systems with various context dynamism through context sharing and support privacy protection, a context modeling framework CSM-H-R is proposed in this paper. As an extension of our prior work of applying CSMs to smart elevators, the framework is generalized to support context-rich intelligent systems that require context reasoning and sharing. As presented in this study, the framework's benefits include supporting automatic reasoning, interoperability, context-sharing, privacy protection, and explainability. It takes care of context dynamism and can be applied to context data that is not limited to HLC.

The framework is implemented using a general-purpose programming language. The code is available at https://github.com/songhui01/CSM-H-R. This study presents a primary evaluation of an implementation of CSM-H-R for supporting CSM building using two customized formats (each for one type of intelligent system), execution time, and data compression. The evaluation results of the novel framework demonstrates that CSM-H-R is feasible in modeling context for smart applications, the time efficiency for the data processing, and the facilitation of context sharing through data compression. Additionally, the research discusses the inherent attribute of CSM-H-R related to its support for privacy protection.

REFERENCES

[1] A. A. Abdellatif, A. Mohamed, C. F. Chiasserini, M. Tlili, and A. Erbad, "Edge computing for smart health: Context-aware approaches, opportunities, and challenges," *IEEE Netw.*, vol. 33, no. 3, pp. 196–203, May 2019.

[2] E. Batista, M. A. Moncusi, P. López-Aguilar, A. Martínez-Ballesté, and A. Solanas, "Sensors for context-aware smart healthcare: A security perspective," *Sensors*, vol. 21, no. 20, p. 6886, Oct. 2021.

[3] F. Kiani and Ö. F. Saraç, "A novel intelligent traffic recovery model for emergency vehicles based on context-aware reinforcement learning," *Inf. Sci.*, vol. 619, pp. 288–309, Jan. 2023.

[4] R. Zhu, S. Wu, L. Li, P. Lv, and M. Xu, "Context-aware multiagent broad reinforcement learning for mixed pedestrian-vehicle adaptive traffic light control," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19694–19705, Oct. 2022.

[5] S. Chavhan, D. Gupta, B. N. Chandana, A. Khanna, and J. J. P. C. Rodrigues, "IoT-based context-aware intelligent public transport system in a metropolitan area," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6023–6034, Jul. 2020.

[6] S. Yue, S. Yue, and R. Smith, "A survey of testing context-aware software: Challenges and resolution," in *Proc. Int. Conf. Softw. Eng. Res. Practice (SERP)*, 2016, p. 102.

[7] Larry Bielawski and Robert Lewand, *Intelligent Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies*. Hoboken, NJ, USA: Wiley, 1991.

[8] Q. Deng and D. Soffker, "A review of HMM-based approaches of driving behaviors recognition and prediction," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 1, pp. 21–31, Mar. 2022.

[9] N. Deepa, B. Prabadevi, P. K. Maddikunta, T. R. Gadekallu, T. Baker, M. A. Khan, and U. Tariq, "An AI-based intelligent system for healthcare analysis using ridge-adaline stochastic gradient descent classifier," *J. Supercomput.*, vol. 77, no. 2, pp. 1998–2017, Feb. 2021.

[10] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," *Neural Comput. Appl.*, vol. 32, pp. 16205–16233, Jan. 2020.

[11] Z. Ning, P. Dong, X. Wang, and J. J. Rodrigues, and Feng Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–24, 2019.

[12] A. Khattak, N. Akbar, M. Aazam, T. Ali, A. Khan, S. Jeon, M. Hwang, and S. Lee, "Context representation and fusion: Advancements and opportunities," *Sensors*, vol. 14, no. 6, pp. 9628–9668, May 2014.

[13] S. van Engelenburg, M. Janssen, and B. Klievink, "Designing context-aware systems: A method for understanding and analysing context in practice," *J. Log. Algebr. Methods Program.*, vol. 103, pp. 79–104, Feb. 2019.

[14] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, "Semantic interoperability in heterogeneous IoT infrastructure for healthcare," *Wireless Commun. Mobile Comput.*, vol. 2017, pp. 1–10, Jan. 2017.

[15] A. Cimmino, M. Poveda-Villalón, and R. García-Castro, "EWoT: A semantic interoperability approach for heterogeneous IoT ecosystems based on the web of things," *Sensors*, vol. 20, no. 3, p. 822, Feb. 2020.

[16] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: State of the art and technical challenges," *Pervas. Mobile Comput.*, vol. 17, pp. 159–174, Feb. 2015.

[17] E. de Matos, R. T. Tiburski, C. R. Moratelli, S. J. Filho, L. A. Amaral, G. Ramachandran, B. Krishnamachari, and F. Hessel, "Context information sharing for the Internet of Things: A survey," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106988.

[18] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure EHRs sharing of mobile cloud based E-health systems," *IEEE Access*, vol. 7, pp. 66792–66806, 2019.

[19] S. Yue and R. K. Smith, "Applying context state machines to smart elevators: Design, implementation and evaluation," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–9.

[20] A. Pliatsios, K. Kotis, and C. Goumopoulos, "A systematic review on semantic interoperability in the IoE-enabled smart cities," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100754.

[21] V.-H. Hoang, E. Lehtihet, and Y. Ghamri-Doudane, "Privacy-preserving blockchain-based data sharing platform for decentralized storage systems," in *Proc. IFIP Netw. Conf. (Netw.)*, Paris, France, 2020, pp. 280–288.

[22] S. Yue, R. Smith, and S. Yue, "A state-based approach to context modeling and computing," in *Proc. 14th IEEE Int. Conf. Ubiquitous Intell. Comput.*, 2017, pp. 1–6.

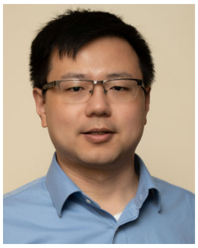
[23] A. Bousdekis, N. Papageorgiou, B. Magoutas, D. Apostolou, and G. Mentzas, "A probabilistic model for context-aware proactive decision making," in *Proc. 7th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2016, pp. 1–6.

[24] A. Bucchiarone, A. Marconi, M. Pistore, and H. Raik, "A context-aware framework for dynamic composition of process fragments in the Internet of services," *J. Internet Services Appl.*, vol. 8, no. 1, pp. 1–23, Dec. 2017.

[25] H. Chegini and A. Mahanti, "A framework of automation on context-aware Internet of Things (IoT) systems," in *Proc. 12th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, Dec. 2019, pp. 157–162.

[26] E. Williams and J. Gray, "Contextion: A framework for developing context-aware mobile applications," in *Proc. 2nd Int. Workshop Mobile Develop. Lifecycle*, New York, NY, USA, Oct. 2014, p. 27.

[27] R. Bergmann, L. Grumbach, L. Malburg, and C. Zeyen, "ProCAKE: A process-oriented case-based reasoning framework," in *Proc. ICCBR Workshops*, vol. 2567, 2019, pp. 156–161.
[28] L Rodriguez-Benitez, J Moreno-Garcia, JJ Castro-Schez, C Solana, and L Jimenez, "Action recognition in video sequences using a mealy machine," *Int. J. Comput. Inf. Eng.*, vol. 2, no. 5, pp. 1383–1389, 2008.
[29] T. Teixeira, D. Jung, G. Dublon, and A. Savvides, "Recognizing activities from context and arm pose using finite state machines," in *Proc. 3rd ACM/IEEE Int. Conf. Distrib. Smart Cameras (ICDSC)*, Aug. 2009, pp. 1–8.
[30] D. A. Lambert, "A blueprint for higher-level fusion systems," *Inf. Fusion*, vol. 10, no. 1, pp. 6–24, Jan. 2009.
[31] S. Meyer and A. Rakotonirainy, "A survey of research on context-aware homes," in *Proc. Australas. Inf. Secur. Workshop Conf. ACSW Frontiers*, vol. 21, 2003, pp. 159–168.
[32] T. Van Nguyen and D. Choi, "Context reasoning using contextual graph," in *Proc. IEEE 8th Int. Conf. Comput. Inf. Technol. Workshops*, Jul. 2008, pp. 488–493.
[33] I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," *Expert Syst. Appl.*, vol. 97, pp. 205–227, May 2018.
[34] H. Seo, J. Park, M. Bennis, and M. Debbah, "Semantics-native communication via contextual reasoning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 9, no. 3, pp. 604–617, Jan. 2023.
[35] F. Mourchid, J. Ben Othman, A. Kobbane, E. Sabir, and M. El Koutbi, "A Markov chain model for integrating context in recommender systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[36] J. Hong, E.-H. Suh, J. Kim, and S. Kim, "Context-aware system for proactive personalized service based on context history," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7448–7457, May 2009.
[37] A. K. Ramakrishnan, D. Preuveneers, and Y. Berbers, "Enabling self-learning in dynamic and open IoT environments," *Proc. Comput. Sci.*, vol. 32, pp. 207–214, Jan. 2014.
[38] H. Gao, J. Zhu, T. Zhang, G. Xie, Z. Kan, Z. Hao, and K. Liu, "Situational assessment for intelligent vehicles based on stochastic model and Gaussian distributions in typical traffic scenarios," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 3, pp. 1426–1436, Mar. 2022.
[39] J. W. Lee and A. Helal, "Modeling and reasoning of contexts in smart spaces based on stochastic analysis of sensor data," *Appl. Sci.*, vol. 12, no. 5, p. 2452, Feb. 2022.
[40] M. Gheisari, H. E. Najafabadi, J. A. Alzubi, J. Gao, G. Wang, A. A. Abbasi, and A. Castiglione, "OBPP: An ontology-based framework for privacy-preserving in IoT-based smart city," *Future Gener. Comput. Syst.*, vol. 123, pp. 1–13, Oct. 2021.
[41] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6092–6102, Sep. 2020.
[42] S. Ji, J. He, A. S. Uluagac, R. Beyah, and Y. Li, "Cell-based snapshot and continuous data collection in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 9, no. 4, pp. 1–29, Jul. 2013.
[43] B. Hidasi and D. Tikk, "Approximate modeling of continuous context in factorization algorithms," in *Proc. 4th Workshop Context-Awareness Retr. Recommendation*, Apr. 2014, pp. 3–9.
[44] A. Padovitz, S. W. Loke, A. Zaslavsky, B. Burg, and C. Bartolini, "An approach to data fusion for context awareness," in *Int. Interdiscipl. Conf. Modeling Using Context*. Cham, Switzerland: Springer, 2005, pp. 353–367.
[45] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Comput.*, vol. 16, no. 4, pp. 62–74, Oct. 2017.
[46] H. Alemdar, H. Ertan, O. D. Incel, and C. Ersoy, "ARAS human activity datasets in multiple homes with multiple residents," in *Proc. 7th Int. Conf. Pervasive Comput. Technol. Healthcare Workshops*, May 2013, pp. 232–235.
[47] J. Bhogal and P. Moore, "Towards object-oriented context modeling: Object-oriented relational database data storage," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 542–547.
[48] A. L. Antunes, E. Cardoso, and J. Barateiro, "Incorporation of ontologies in data warehouse/business intelligence systems—A systematic literature review," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 2, Nov. 2022, Art. no. 100131.
[49] N. W. Rahayu, R. Ferdiana, and S. S. Kusumawardani, "A systematic review of ontology use in E-learning recommender system," *Comput. Educ., Artif. Intell.*, vol. 3, Jan. 2022, Art. no. 100047.
[50] S. Mishra and S. Jain, "Ontologies as a semantic model in IoT," *Int. J. Comput. Appl.*, vol. 42, no. 3, pp. 233–243, Apr. 2020.
[51] I. Osman, S. Ben Yahia, and G. Diallo, "Ontology integration: Approaches and challenging issues," *Inf. Fusion*, vol. 71, pp. 38–63, Jul. 2021.
[52] D. Venu, A. V. R. Mayuri, S. Neelakandan, G. L. N. Murthy, N. Arulkumar, and N. Shelke, "An efficient low complexity compression based optimal homomorphic encryption for secure fiber optic communication," *Optik*, vol. 252, Feb. 2022, Art. no. 168545.
[53] E. Pintelas, I. E. Livieris, and P. Pintelas, "A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability," *Algorithms*, vol. 13, no. 1, p. 17, Jan. 2020.



SONGHUI YUE received the B.E. degree from the Software Engineering Institute, East China Normal University, in 2009, and the Ph.D. degree in computer science from the Department of Computer Science, The University of Alabama, in August 2019. He is currently an Assistant Professor with the Department of Computer Science, Charleston Southern University. He has over five years of industry experience in web and mobile-based application development and distributed data processing. His research interests include intelligent software engineering, context reasoning for the IoT, intelligent systems, smart city, artificial intelligence, and machine learning.

XIAOYAN HONG (Senior Member, IEEE) received the M.S. degree from the Computer Science Department, Zhejiang University, China, and the Ph.D. degree in computer science from the University of California at Los Angeles, in 2003. She is currently an Associate Professor with the Department of Computer Science, The University of Alabama. She is also involved with a project for the NSF Research Experience for Undergraduates National Academy of Engineering Grand Challenges Program. Her research interests include mobile and wireless networks, future wireless internet, and high-performance networks. Her current research interests include mobile ad hoc networks, wireless mesh networks, vehicular networks, and delay-tolerant networks.

RANDY K. SMITH (Senior Member, IEEE) is currently an Associate Professor with the Department of Computer Science, The University of Alabama. He is also the Director of the Center for Transportation Operations, Planning, and Safety, The University of Alabama. He also serves as a Principal Investigator for the Center for Advanced Public Safety, The University of Alabama. He is a Faculty Affiliate with Alabama Transportation Institute. He has over 20 years of experience working at the intersection of computer science and transportation safety. He has worked with multiple states' Department of Transportation and currently oversees a portfolio of research projects ranging from crash data curation to near real-time digital twins for smart corridors.