CHAPTER 4

EXTENDED ENTITY HIERARCHICAL RELATIONAL AXIOM REPRESENTATION OF CONTEXT

4.1 INTRODUCTION

Humans have always used their understanding of circumstances or context to navigate the world around them, to organize information, and to adapt to conditions. The behaviour of a context-aware system depends not only on its internal state and user interactions, but also on the context sensed during its execution, as cited by Ejigu et al (2008).

The existing models of context representation do not fully address the basic issues related to context data due to variations in the types of context information they can represent. Some models take the user's current situation, while others consider the physical environment. The challenge is to put a more generic approach in context modeling capture, and represent various features of context information including a variety of context information and their dependencies.

Context representation needs a model that represents the data and its meaning. Semantic representation, support for processing, conversion of data into the necessary knowledge and action, are important for context-aware reasoning and decision support.
In this chapter, a new approach for a context representation is proposed. The proposed context meta model uses context entities, their hierarchies, relationships, axioms and metadata as basic building blocks for context representation.

4.2 CONTEXT DEFINITION

Based on Dey’s (2000) generalized definition and Winograd’s (2001) specification on the definition of context, context was redefined by Ejigu et al (2008). Here, context is defined as an operational term. It depends on the intention for which it is collected. It also interprets the operations involved in an entity at a particular time and space, rather than the inherent characteristics of the entities and the operations themselves.

Consider a smart medical ward in a hospital where patients, nurses, physicians, etc. are involved. Assume that the ward is equipped with context sensor technologies (hardware and software) in its rooms, corridors and garden at the disposal of the individuals involved. Patients admitted to the hospital may need intensive follow up which may create staff shortage, and may result in inappropriate care to the needy ones due to overloading. Human interventions are needed only when alerted by the system. Live multimedia recording and transmission of an event are used for monitoring purposes. Personalized attendances are given to those who are concerned, as cited in OpenGALEN project (2007).

Classes of generic context entities for the above scenario and its sub-entities are shown in Figure 4.1.
Figure 4.1 Smart Hospital Scenarios

Context data is collected for each participating entity using hardware or software tools. The figure shows the relation between entities including the subEntityOf relation. At the root of the hierarchy is a global entity named ContextEntity. Figure 4.1 also shows the major classes of context entity descriptors, such as personal, device, physical environment, network, activity, service, and location contexts. Personal entity provides contexts like a person’s identity, address, service preference, device ownership, activity, location etc. Device entity provides contexts like hardware properties, software properties, display properties, device capabilities etc. Network entity contexts are expressed in terms of properties like delay and error characteristics, data rate, transport protocols, etc. Physical
environment entity provides contexts like illumination, noise level, humidity, temperature etc. Activity entity context shows whether an activity is scheduled or not / it needs special location or not / type of activity etc. Location entity provides contexts about its containment and situation with respect to other entities. Service entity provides contexts about where the service is located, type of service (data service, audio service, video service, and application service), service availability, etc.

Context can be defined on generic and domain levels as shown in Table 4.1. The column on the left side of the Table 4.1 shows a generic level definition of the relationship between entities. This is equivalent to defining the domain and range of a relation. The column on the right side of Table 4.1 shows the use of the same relationship in a specific domain of application, as discussed above in the smart hospital scenario.

Table 4.1 Generic and Domain based definition of relationships

<table>
<thead>
<tr>
<th>Generic Level Definition</th>
<th>Domain Level Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person isEngagedIn Activity</td>
<td>Physician isEngageIn Patient – treatment</td>
</tr>
<tr>
<td>Location isLocatedIn Location</td>
<td>Canteen isLocatedIn Campus</td>
</tr>
<tr>
<td>Person isLocatedIn Location</td>
<td>Physician isLocatedIn Canteen</td>
</tr>
<tr>
<td>Network hasDataRate xxx</td>
<td>ConnectionX hasDataRate low</td>
</tr>
</tbody>
</table>

The statements in Table 4.1 need a higher-level statement. These statements can be expressed in terms of a meta-statement or axioms. Information like the time of occurrence, precision, source of data, etc., can be part of meta-information. For example, “physician isLocatedIn canteen” at a
given time $t$ and the precision of the statement “a network connection has low speed” is 90%. Time $t$ that is associated with “physician isLocatedIn canteen” and the precision of 90% that is associated with “connection hasSpeed low”, refer not to the individual components of the statements, but to the entire statements. Regarding the axioms, if a relation LocatedIn is transitive then LocatedIn obeys the transitivity axiom: for example, “canteen isLocatedIn campus” and “physician isLocatedIn canteen” means that “physician isLocatedIn campus”; similarly, if owned-by is an inverse of owns then: “device owned-by person” means “person owns device”.

4.3 PROPOSED EEHRAM MODEL FOR CONTEXT REPRESENTATION

The proposed EEHRAM is a layered context representation meta-model that uses a set of extended entities (EE), a set of hierarchies (H), a set of relations (R), a set of axiomatic relations (A), and a set of metadata (M), to represent the context data and its semantics. The proposed model EEHRAM is composed of the following components:

- EE is a set of extended entities for which context is to be captured.
- H is a set of hierarchical relations that forms an inversely directed acyclic graph (inverted DAG) on entities. These relations are binary in nature. The Nodes of the graph represent entities, and the arcs of the graph represent hierarchical relations. The root entity at the top of the hierarchy graph is a global entity known as ContextEntity.
• R stands for the union of the sets of binary relations $R_e$ and $R_a$. $R_e$ is the set of binary relations having both its domain and range from the set $EE$. $R_a$ is the set of binary relations defined from the set of entities $EE$ to the set of literals, representing the entity attributes. The domain of the relation $R_a$ is the set of $EE$ while its range is a set of literal values.

• A is a set of axiomatic relations. An axiomatic relation is a relation about relations. For all $r_i$, an element of a set of relations $R$, have a zero or more $a_j$, an element of set of axioms $A$ that $r_i$ obeys. For example, define a relation $r_1$ as a transitive relation then $r_1$ obeys the transitivity property (axiom):

$$(e_1, r_1, e_2) \text{ and } (e_2, r_1, e_3) \Rightarrow (e_1, r_1, e_3).$$

• Similarly, if a relation $r_2$ is defined as a symmetric relation, then $r_2$ obeys the symmetric property (axiom):

$$(e_1, r_2, e_2) \Rightarrow (e_2, r_2, e_1).$$

• M is a set of metadata about a defined relation instance. The set of Metadata together with the set of Axioms enhances the EEHARAM model for handling the semantics of the context data.

Hierarchy is an important structure to organize and classify context entities and relations into two layers. The layered organization helps to classify and tag context data as domain dependent or domain independent (generic). Figure 4.2 shows a layered graphical representation of the proposed
EEHRAM structure that shows hierarchies, entities, entity relations, attribute relations, axiomatic relations, metadata and the layers: (Layer (a) is the generic layer and layer (b) is the domain layer).

Figure 4.2 Layered representation of the EEHRAM Components

4.3.1 EEHRAM Model using a Real Life Example

Consider an application in a medical domain, where the context data come from medical entities, like patients, doctors, activities and events in a hospital, devices, locations, etc. The representation of the components of the EEHRAM model, using a few example data from the application in a smart medical domain, is given in Figure 4.3.
Entity and Hierarchy

Activity, Person and Device are examples of generic entity classes and they are presented in the generic layer. They are high-level context entities from which specific entities can be derived, and they are common to all domains of applications. All entities in this category have a hierarchical relation named isa with the root entity known as ContextEntity. Meeting, Patient, Doctor and Phone are examples of domain entity classes in a medical application, and they are presented in the domain layer. Bob, Pascal and Sphone0095 are examples of entity instances in the medical application. Examples of hierarchical relations from Figure 4.3 are (Device, isa, ContextEntity), (Person, isa, ContextEntity), (Activity, isa, ContextEntity), (Doctor, isa, Person), (Pascal, instanceOf, Doctor), etc.
Relation

Relations like \((\text{Activity, hasStartTime, time})\), \((\text{Person, isEngagedIn, Activity})\), \((\text{Person, locatedWith, Person})\), \((\text{Person, locatedWith, Device})\), \((\text{Device, hasMemory, memory})\) are defined in the generic layer. Such generic relations can be inherited down in the hierarchy, by the sub-entities and instances in the specific domain of application. Similarly, relations like \((\text{Meeting, hasEndTime, time})\) and \((\text{Patient, hasDoctor, Doctor})\) from the medical application are presented in the domain layer. They restrict the domain and the range of the relations that are inheritable down in the hierarchy by entity instances. Finally, relations like \((\text{Bob, hasBodyTemp, 39.5})\), \((\text{Pascal, Owns, SPhone0095})\), \((\text{Bob, hasDoctor, Pascal})\) and \((\text{SPhone0095, hasMemory, 400})\) defined on entity instances represent the basic context definition formalism.

Relations like \(\text{hasEndTime, hasMemory, hasStartTime, and hasBodyTemp}\) are elements defined as attribute relations \(\left( R_a \right)\) because the range of these relations is the set of literal values. Relations like \(\text{hasDoctor, locatedWith and owns}\) are elements defined as entity relations \(\left( R_e \right)\), because they are defined from entity to entity, i.e., both their domain and range are drawn from the set of entities.

Some relations in Figure 4.3 are defined to have associated axioms and some have metadata. Examples of relations with associated axioms are \((\text{Person, locatedWith, Device})\) and \((\text{Pascal, owns, SPhone0095})\). In Figure 4.3, \(\text{locatedWith}\) is defined to be symmetric, and therefore it obeys the symmetric-axiom property, which means the relation \((\text{Device, locatedWith, Person})\) automatically holds true. Similarly, because \(\text{owns}\) is defined to be an inverse of \(\text{hasOwner}\), it obeys the inverse-axiom property, which means the relation \((\text{SPhone0095, hasOwner, Pascal})\) automatically holds true. The
relation \((Person, \text{engagedIn}, \text{Activity})\) has a metadata that informs about its precision represented by \(\text{hasPrecision}\).

4.3.2 Layers, Axioms and Metadata

Generic layer in the EEHRAM model consists of classes representing basic entities. Such classes have the \textit{generalization} relation with the base classes called \textit{ContextEntity} that represents the EEHRAM root entity. All association relations and attributes defined on these entities apply to all sub-entities lower down in the hierarchy. They are defined independent of the domain of application. For example, if a relation \(\text{hasAddress}\) applies to an entity class Person, i.e. \(\text{hasAddress}(\text{Person, Address})\), then this relation applies to all sub entities and instances of Person. The domain layer represents entities that define specific domains of application. In the hierarchy graph, the domain layer consists of all entities that do not have a direct generalization relation with the root entity. In addition to their defined relations, they inherit relations from the parent entities.

An axiom is a sentence, proposition or rule that is taken as valid, and serves as a necessary starting point and formal logical expression, for reducing and inferring logically consistent statements. Axiomatic relations can be defined both at the generic and domain layer of the EEHRAM model. Descriptions of some of the generic level axiomatic relations, \textit{sameAs}, \textit{inverse}, \textit{symmetric} and \textit{transitive} are given as follows:

\begin{align*}
\forall r \in R & \quad \text{symmetric} (r) \iff (\forall e_1, e_2 \in E, r(e_1, e_2) \Rightarrow r(e_2, e_1)) \\
\forall r \in R & \quad \text{transitive} (r) \iff (\forall e_1, e_2, e_3 \in E, r(e_1, e_2) \land r(e_2, e_3) \Rightarrow r(e_1, e_3)) \\
\forall r_1, r_2 \in R & \quad \text{inverse} (r_1, r_2) \iff (\forall e_1, e_2 \in E, r_1(e_1, e_2) \Rightarrow r_2(e_2, e_1)) \\
\forall r_1, r_2 \in R & \quad \text{sameAs} (r_1, r_2) \iff (\forall e_1, e_2 \in E, r_1(e_1, e_2) \Rightarrow r_2(e_1, e_2))
\end{align*}
Similarly, application domain based axiomatic relations are used to state axioms and rules that are used to deduce further knowledge for reasoning. A statement “under normal condition, a patient is always treated by the same doctor” can be considered as an axiom (assumption) in a medical domain. Using this assumption, another domain based deduction rule can be created as follows:

\[
\forall d \text{ instanceOf Doctor}, p \text{ instanceOf Patient}: \text{hasDoctor}(p,d) \land \\
\text{engagedInActivity}(p, \text{takeTreatment}) \Rightarrow \text{engagedInActivity}(d, \text{giveTreatment})
\]

Some degree of accuracy in such axioms or rules can be associated as metadata.

*Metadata* is data about data. Metadata in context modeling is important, to associate quality, precision, source, time stamps, and other information to the context data. Such information is important to prepare the context data for reasoning and decisions. In the EEHRAM model, metadata is a relation that describes another relation instance. For example, consider a given context information that says “patient is located in the garden”; it is possible to make other statements about this statement to answer questions like: *Who* reported this information? *Which* service is used to report this information? *When* did it happen? *How* accurate is the information? *Why* is the subject in this situation? *What* will happen next? etc.

The different data representations like the Unified Modeling Language (UML), binary relation, and Resource Description Framework (RDF) model for transforming the proposed EEHRAM conceptual model into data representations are discussed below.
### 4.3.3 Mapping the Hierarchical Conceptual Model to the UML

Unified Modeling Language (UML) is used to formalize hierarchical as a conceptual context representation model. The UML is a standard specification language for object modeling. It is used as a tool for ORDBMS designing. Incremental mapping of the concepts in EEHRAM to the UML is given below:

- *Extended Entity* in EEHRAM can be represented as a UML class.

- The concept of *hierarchical* relation in EEHRAM can be represented as *generalization* relationship in the UML.

- *Entity relations* in EEHRAM can be represented as *association relationships* in UML, and attribute relations in EEHRAM can be represented using *attributes* in the UML class.

- *Axiomatic relations* in EEHRAM can be represented as *association classes* in the UML. The concept of metaclass can also be used to represent axiomatic properties like *symmetric property*, *inverse property*, etc.

- *Metadata* in EEHRAM can be represented using *association classes* in the UML.

The above mapping of EEHRAM and UML is also shown in Table 4.2.
Table 4.2 Mapping the proposed EEHRAM with the UML

<table>
<thead>
<tr>
<th>Hierarchical</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Entity</td>
<td>(EE)</td>
</tr>
<tr>
<td>Hierarchy Relation</td>
<td>(H)</td>
</tr>
<tr>
<td>Entity Relation</td>
<td>(R_e)</td>
</tr>
<tr>
<td>Attribute Relation</td>
<td>(R_a)</td>
</tr>
<tr>
<td>Axioms</td>
<td>(A)</td>
</tr>
<tr>
<td>Metadata</td>
<td>(M)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hierarchical</th>
<th>UML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Entity</td>
<td>UML Class</td>
</tr>
<tr>
<td>Hierarchy Relation</td>
<td>Generalization Relationship</td>
</tr>
<tr>
<td>Entity Relation</td>
<td>Association Relationship</td>
</tr>
<tr>
<td>Attribute Relation</td>
<td>Attributes in UML Class</td>
</tr>
<tr>
<td>Axioms</td>
<td>Association Classes</td>
</tr>
<tr>
<td>Metadata</td>
<td>Association Classes</td>
</tr>
</tbody>
</table>

The limitations of using UML as a modeling tool to represent the EEHRAM modeling components are that, attribute relationships, context metadata and axiomatic relations cannot be adequately represented in the UML. Representing axiomatic relations in the EEHRAM model as association classes in the UML has the limitation of repeating the same set of axiomatic relations for every occurrence of the instance of the entity. For example, the inverse axiomatic relation, hasCustomer that is defined on hasDriver using an association, class should be independent of every instance of the association, but refer to the association itself.

The limitations of UML to mapping to the proposed EEHRAM meta-model can be overcome, by using the extended UML features like meta-classes and association-classes. There is limited support for the representation of the semantic aspect of context data. The UML is used as it supports the proposed ontology based Object Relational database management system. UML modeling tools can be used to formalize the representation of the EEHRAM model.
4.3.4 Mapping the Proposed EEHRAM and Relational Models

Context and context metadata can be represented using notations and concepts in a binary relation. The binary relation \( R \) is a subset of a Cartesian product \( X \times Y \), where \( X \) and \( Y \) are arbitrary sets. The sets \( X \) and \( Y \) are called the domain and range of the relation respectively. The statement \( (x, y) \) is a subset of \( R \), is read "\( x \) is \( R \)-related-to \( y \)" , and is denoted by \( xRy \) or \( R(x,y) \). The order of the elements in each pair is important: if \( a \neq b \), then \( R(a, b) \) and \( R(b, a) \) can be true or false, independently of each other.

Based on this definition of relations, given the sets of context entities \( E \) and the set of values \( V \) drawn from the set of context entities and literal values, a relation \( R \) is the subset of the Cartesian product of the sets of \( E \) and \( V \) as shown in Equation (4.1)

\[
R \subseteq \{(e_i,v_j): e_i \in E, v_j \in V\} \quad (4.1)
\]

Consider all meaningful set of statements,

\((e_i,v_j) \in R\) that can also be represented as:

\[\{R(e_i,v_j): (e_i,v_j) \in R\}\] or in a more linear form

\[\{(e_i,R,v_j): (e_i,v_j) \in R\}\]

This triple can be used to define a context (\( C \)) as shown in Equation (4.2)

\[
C \equiv \{(e_{i,k}, r_k, v_{j,k}) : e_{i,k} \in E_k, r_k \in R, v_{j,k} \in V_k\} \quad (4.2)
\]

This can be extended to define context and context metadata (\( CM \)) using the basic context \( C \), meta-relation (\( RM \)) and meta-value (\( VM \)) as shown in Equation (4.3) and Equation (4.4)

\[
CM \equiv \{(c_i, rm_k, vm_j) : c_i \in C, rm_k \in RM, vm_j \in VM\} \quad (4.3) \text{ Or}
\]
\[
CM \equiv \{((e_i, r_k, v_j), rm_i, vm_p): e_i \in E, r_k \in R, v_j \in V, rm_i \in RM, vm_p \in VM\} \quad (4.4)
\]
Basic context data (medical application):

(Schedule, isA, Service), (TimeTable, instanceOf, Schedule),

(Meeting, isA, Activity), (Meeting005, instanceOf, Meeting),

(Meeting005, hasStartTime, #201211081400)

(Doctor, isA, Person), (Pascal, instanceOf, Doctor)

(Pascal, isEngagedIn, Meeting005)

Context with metadata:

((Pascal, isEngagedIn, Meeting005), hasSource, Agenda)

((Pascal, isEngagedIn, Meeting005), hasPrecision, yy%)

N-ary relation (relation of degree N) as a base of the relational database model inherits its properties from binary relations.

Relational models can be used to represent the entity, hierarchy, relations and metadata components of the EEHRAM model. They can also be extended to represent axioms in the EEHRAM model using definitions like (locatedWith, is, symmetric), (locatedWith, is, transitive), (owns, inverseOf, hasOwner), etc. However, this is not sufficient to represent fully the semantic aspect of the context data represented in the EEHRAM.

4.3.5 Mapping the EEHRAM and RDF Data Model

RDF models have been in use to represent semantic metadata in different application domains. The work by Bouzeghoub(2004) uses the RDF semantic description model, to allow the reuse and assembling of learning objects that represent pedagogical materials available on the web. The core element is the representation of semantic metadata that allows the description
of the domain model, user model and learning object model. In this section, the RDF and its extensions to build a generic context meta-model, are discussed.

The primary characteristic of a context data is that, it possesses an actor or a subject (an entity). The context value defined on the subject is expressed in terms of multiple properties. Further, the terms predicate and object are used to represent the situation of the subject with respect to a specific property of a context data. This convention goes with the RDF-triple representation formalism, $<\text{subject}, \text{predicate}, \text{object}>$, which in turn, maps all types of relations in the proposed EEHRAM model. Additional context metadata can also be included as part of the context data. In addition to the subject, predicate and object triples, context modeling requires context metadata to extend the context model to the historic, probabilistic, or confidence-carrying models.

Such attributes are meaningful only when they refer to a particular instance of the triple, not to each individual element. To describe this situation, the RDF and its extension, called RDF reification W3C (2007), are used.

Reification is used to represent facts that must then be manipulated in some way; for example, to compare logical assertions from different witnesses to determine their credibility. The message "John is six feet tall" is an assertion of truth that commits the sender to the fact, whereas the reified statement, "Mary reports that John is six feet tall" defers this commitment to Mary.

In the same way, a reified RDF data contains each original statement as a resource and the other additional statements made about it. The four properties used to model the original statement as the RDF resources are,
subject, predicate, object and type. A new resource with these four properties represents the original statement and can be used as a subject or an object of other statements with an additional statement made about it.

Figure 4.4 shows the demonstration of a context metadata representation, using statement reification. This figure shows an example of triple statement: “Bob is located in the Library”. This statement can be reified by additional meta-information like “is reported by sensor#5”, “has accuracy of 88%”, “has occurred at 11:40 today”, etc.

![Figure 4.4 Context metadata represented using reification](image)

Figure 4.4 shows an equivalent graphical representation of the RDF data model for the reified context data. The RDF reification principle can be used to represent additional context attributes to the basic context triples.
The RDF is one of the major building blocks in a formalism, to represent ontology, which has features for defining and representing axioms. Therefore, axioms can be represented using RDF/OWL formalisms. Details on mapping the EEHRAM to ontology are discussed in chapter 5.

4.4 CONCLUSION

In this chapter, the context representation model EEHRAM is proposed. Components of the EEHRAM model (entity, hierarchy, relations, axioms and metadata) are derived directly from the definition of the context. A generic EEHRAM graph is used to represent an abstract conceptual model. This makes it simple, as it follows the abstraction and conceptualization of
context data and its semantics, in the form of axioms and metadata. The different data representation formalisms like the UML, binary relation with its extended meta-form and the RDF model can be used to convert the EEHRAM conceptual model into concrete data representation. A more comprehensive approach, which deals with the mapping of the proposed EEHRAM model into standard data management structures, that support the storage and processing of both the context data and the context semantics, are presented in the next chapter.