CHAPTER 3
REVIEW OF MODELLING AND EXTRACTION OF CONTEXTS AND SITUATIONS

3.0 CHAPTER INTRODUCTION

Human brain has implicit and clear understanding about background of interaction that is, “context”, which makes the interaction amongst humans rich. However, user-device interactions are not as rich and clear as human – human interactions. This is because devices are not capable of utilizing contexts to adapt to user’s behavior and environment.

Capturing contexts comes naturally to human beings, but machine based extraction of context is challenging due to its abstractness. Mechanisms are being developed to make contexts explicit. For example, contexts have been defined to be location, identity, time, season, temperature, emotional state, locomotion physical and physiological state of user, surrounding objects and persons etc. [89 - 90]. Once a definition of context is adopted, it can be captured by utilizing various data gathering sources including sensors, transaction logs, calendar entries etc. [91]. Methods of context extraction from sensors have been exclusively studied here.

Isolated pieces of contextual data are not useful for decision making. These need to be integrated to reflect comprehensive contextual knowledge as “Situations” of interest [92]. Access to situations encapsulates all the sensor and context level algorithms from end user interface, thus simplifying communication and computation at that end. Abstraction of situations increases adaptability and autonomicity of sensor based services [93]. Representation, recognition and autonomous inference of situations are few issues to be addressed for situation extraction. These issues and related current state of art is subject matter of this chapter. The chapter is divided into four sections. First two sections discuss definitions and extraction of contexts and situations. In third section, proposed framework for computation of contexts and situations from sensor data has been elaborated.
The first section is about definition of context and its machine based processing. Concept of context with respect to machine based extraction has been discussed through analysis of various widely accepted definitions of context [19-21][89][94]. Formal models of context representation are crucial for their interoperability across devices [95]. Some common modeling methods being used in applications are list based, object based, role based and ontology based [90- 91][96]. Each modeling mechanism is supported by its reasoning method to extract contexts from data. Mechanisms catering to uncertainty of input data have been studied [97].

In Section 3.2, situation definitions, relevant in computing definitions, have been studied. Several specification and data based methods for situation representation and extraction have been proposed in literature [98 - 101].Extraction of situations from contexts has been studied also.

In Section 3.3, proposed framework to integrate concepts of sensor data management, contexts and situations have been described. Purpose of this integration is to obtain double benefits of solving sensor data handling issues along with enabling useful applications by extracting contexts and situation of the monitored systems.

The chapter has been concluded in Section 3.4 with brief outline of literature surveyed in this chapter about the abstract concepts of contexts and situations. It also summarizes the findings and conclusions obtained from the study.

3.1 CONTEXTS AND THEIR COMPUTATION

3.1.1 Definition of Context

Context is an abstract term used variably in layman’s language. It encompasses wide meanings. Webster’s dictionary defines context as "The whole situation, the background or the environment relevant to a particular event, personality, creation etc." Researchers have also given formal definitions of context applicable to its computation. Few prominent ones that have gained wider acceptance as follows:-
• [19] defined three important aspects of context as where you are, who you are with and what resources are nearby. Context was exemplified by location, lighting, noise level, network connectivity, communication costs, bandwidth etc.  

• Schmidt et al. [20] defined context as “knowledge about user’s and device’s state, including surroundings, situation, and to a less extent, location”. They categorize contexts in two categories namely human factors and physical environment  

• [89] defined context as “any information that can be used to characterize the situation of an entity”. An entity here may be a person, place or object that is considered relevant to the interaction between a user and an application. He divided context into three aspects namely Computing/ User/ Physical context.  

• [94] defined context as “the set of environmental states and settings that either determines an application’s behavior or in which an application event occurs and is interesting to the user”.

[89] [94] define context by synonyms like “description of environment” and “description of situation”. Definitions by synonyms are too generic and don’t assist in making contexts tangible entities.

Few others define context by example [19 - 20]. Some common examples of contexts that can be relevant to many applications are:

*Location:* - This context is useful for exact location of tracking, presence and absence in monitoring. Location identification coordinates may be absolute or discrete categorical values like room no., room type (For example, kitchen, meeting room etc.).

*State:* - The state of an entity represents that part of context which is directly related to the mind and body of the entity itself. The relevance of information to an individual can be determined by what the person is doing and/or what physical, mental, and emotional state she is in. Thus motion, activity, physical condition, and emotional condition are state related contexts.

*Surroundings:* - Another aspect of context that influences entity’s action and requirements is its surroundings. Examples of surroundings are data about the light conditions, temperature, noise level, or concentration of carbon dioxide in environment etc. Applications can utilize these contexts to trigger an alarm when some indicator data exceed a certain threshold.
Other elements of context: -

There are few contexts that can be identified from additional information about the system. These are:

• **Identity:** - Knowledge of the identity of relevant entities can be useful in context-aware application. This may include the identity of the entity, nearby persons or objects.

• **Time and history:** - Another prominent context element is date and time. This element provides information that answers the question of when a particular context has occurred. Time information is available on every computing device. Historical data about interactions between entities or of an entity’s system usage are elements of context as well. Historical information is useful to deduce correlations between past and present contexts of user.

• **Personal preferences:** - The preferences, interests, schedules etc. of an application’s users also belong to a type of relevant context.

Context awareness has been recognized as an important technology basic to modern computing applications. Conceptual difference between a context aware and regular application is shown in a simple manner in Figure 3.1.

![Figure 3.1: Conceptual difference between a Regular and Context Aware application](image)

A regular application always interacts in a predefined manner with the end user. A context aware application utilizes surrounding context to adapt its interaction accordingly.
Some humane applications that can benefit from becoming context aware and their relevant contexts are shown in Table 3.1. In column 1, contexts that can be relevant in human health tracking, remote elderly monitoring, in-home patient care are exemplified. Human monitoring is a well-researched domain [100][102 - 103]. The contexts in this domain have been further classified as physical, physiological, location, temporal and identity. Some contexts relevant for military, disaster management and weather prediction are listed in subsequent columns.

**Table 3.1**: Relevant Contexts in Various Humane Applications

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical Context</strong></td>
<td>Location, Identification of intrusion presence/absence of enemy Plan of action Control Automatic Vehicles</td>
<td>Presence of Humans Evacuation Route</td>
<td>Meteorological Context Emissions – Atmospheric Process</td>
</tr>
<tr>
<td>Moving-&gt; (Walking, Running) Not Moving-&gt; Posture (standing, sitting, lying)</td>
<td></td>
<td></td>
<td>Hydrological Context Temperature, Dew Point, Wind Speed and Direction, Wind Chill, Wet Bulb Temperature, Precipitation Rate, Storm Total Precipitation</td>
</tr>
<tr>
<td><strong>Location Context</strong></td>
<td></td>
<td></td>
<td>Oceanographic Context Humidity, Heat Index, Mixing Ratio, Barometric Pressure, Pressure Trend,</td>
</tr>
<tr>
<td>Absolute Coordinates e.g. GPS Logical Location e.g. Indoor(in house -&gt; Kitchen, Toilet, Bedroom), In mall, In office (On desk, in meeting, gym, car), Outdoor (In Market, Park, Bus station, On road)</td>
<td></td>
<td></td>
<td>cloud cover, cloud top temperature, sea surface temperature, snow cover, cloud motion vector, outgoing long wave radiation</td>
</tr>
<tr>
<td><strong>Temporal Context</strong> Absolute Time -&gt; Symbolic Time (Day, Night)</td>
<td>Infiltration Detection , Mine Detection</td>
<td>Availability of Aid</td>
<td></td>
</tr>
<tr>
<td><strong>Identity Context</strong> Identity – Human -&gt; Name -&gt; Age -&gt; Gender</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Physiological Context</strong></td>
<td>New problems/ updated low level context  (\rightarrow) new tactics</td>
<td>Impact of Disaster Severity of Disaster</td>
<td></td>
</tr>
<tr>
<td>Body Vital Parameters, Physiological signals related to specific problem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Surrounding Context</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nearby Resources – Persons, Facilities (Hospital, Medical Aid, Police)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Once contexts relevant to an application domain are specified, the next task is to define their sources and extraction methods. Adaptation to context opens up opportunities for the creation of new services to end users. E.g. If the context “user is in meeting” is detected, then profile adaptation service may change phone to silent mode. A health alert application uses activity context like exercising or fallen down to judge if the increase in heartbeat is fatal or normal. In next section mechanisms of formal representation, modeling and reasoning of context have been described.

3.1.2 Modeling and Reasoning about Context

Formal context models describe framework of representation and reasoning of contexts for implementation across applications [104]. A context model is important for structured representation of variety of Contextual Information (CI). Contexts modeled by machines should be in high agreement to those perceived by human beings. Table 3.2 provides categorization of context modeling approaches proposed in literature. Most commonly used context representation models are role – based, spatial-temporal and ontology based models [100] [105].

*List based models* are most simplistic models to represent the context information in key-value pairs. [106] famously used this model to describe contexts in their context toolkit. Such models are efficient for simple and small number of contexts but not suitable for complex data. Context retrieval, visualization and scalability are poor in such models. In simple list based models, there were issues of handling the wide variations in information, complex relationships in context information and temporal information of context.

One of the earlier approaches that break away from flat representations was *graphical models* [22]. These models overcome the lack of formality and generality in the previous list based models. The graphical representation overcomes limitations of list based models to some extent. The graphical representation also provided a formal basis for representation and reasoning on diverse context information.
<table>
<thead>
<tr>
<th>Context Modeling Method</th>
<th>Main Feature</th>
<th>Context Capturing capability</th>
<th>Reasoning of Contextual Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Based Models [106]</td>
<td>Describe contexts and its description as a vector</td>
<td>Key Value pairs</td>
<td>Directory Lookup Methods, Description Logic,</td>
</tr>
<tr>
<td>Graphical Models [22]</td>
<td>Diagrammatic representation of context design</td>
<td>Informal entities to be formalized during development</td>
<td>Unified Modeling Language (UML)</td>
</tr>
<tr>
<td>Object Role Model e.g. Context Markup Language [107]</td>
<td>Semantics of contexts as objects and roles played by them</td>
<td>Objects as bearers of context</td>
<td>UML and Object oriented concepts of reusability and encapsulation</td>
</tr>
<tr>
<td>Spatial context model [108]</td>
<td>Use Geometric or symbolic coordinates for location determination</td>
<td>Annotates spaces as exact locations</td>
<td>Query on position of a particular entity, find objects within given range, find nearest neighbors of an entity</td>
</tr>
<tr>
<td>Formal Logic Based Models [109]</td>
<td>Creation of hierarchy of contexts and their meta information</td>
<td>Represent context as Predicates, composition as rules.</td>
<td>First Order Predicate Logic</td>
</tr>
<tr>
<td>Ontology Based Models [110 - 111]</td>
<td>Use concepts and their relationships</td>
<td>Can capture hierarchical context data using specific logical operators</td>
<td>Classify events/objects to most likely classes. Detect inconsistencies in CI. Ontological reasoning for inference of new CI</td>
</tr>
<tr>
<td>Hierarchical Models [112]</td>
<td>Define context definition and composition from low level abstractions</td>
<td>Cues from sensors and contexts from cues</td>
<td>Markup Languages</td>
</tr>
<tr>
<td>Domain Specific Models e.g. in [113]</td>
<td>Light weight and tailor made for specific applications</td>
<td>Not specific – e.g. ContextML uses XML based schema</td>
<td>Context Schema and Description Logic</td>
</tr>
<tr>
<td>Hybrid Models [96]</td>
<td>Combine features of more than one model described above</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The **Object Oriented** models are graphical models enhanced by object oriented programming, representation and role assigning principles. This approach handles the super class and sub class relationships among different contextual information. Context entities composition and sharing (COSMOS) is a popular model of this category, it was proposed in [114]. Different levels of abstractions of sensor are implemented as contexts and communicate through different interfaces and methods. **Domain Specific** models represent contexts relevant to a particular domain. These
are less flexible and have complex context operators. Service-Oriented Context Aware Middleware (SOCAM) is a domain specific model proposed by [115]. Spatial models are special category of domain specific models used for localization of real world objects. These are used to model contexts for location aware scenarios mainly. “Middlewhere” proposed by [109] is one such model.

The formal logic based models allow context values to be represented formally as simple axioms and rules that relate various contexts. First order logic has been popular to represent contextual propositions and relations [91]. Gaia is a widely used example of this category of context model [116]. Logic based models are difficult to implement due to lack of toolkit support.

Ontology Based Models use ontology for designing contextual information representation and handle the semantics among different context information. Ontologies are application specific ways to represent and reason complex domain information. It provides explicit formal specifications of the contexts in a domain and relations among them. Wide adoption of ontology enables the reuse of previous works and the creation of common and shared domain vocabularies [111]. Once domain ontology is agreed upon, it is very efficient in enhancing interoperability and heterogeneity aspects [117]. Context ontologies streamline the whole process of collecting and managing low-level sensor data, transforming to middle-level information fusion, and to high-level activity recognition. However, there is lack of standardization and limited handling of uncertainty in existing ontological representations. For constrained devices like sensors, ontology based approach may not be suitable in its original formalization and light weight methods are required.

Hierarchical models can be any of the above models with context information defined in logical hierarchies. This model can be efficient in handling context semantics and relationships but it needs to enhance the maintenance of dynamicity in environment of the user's context. [112].

Hybrid model integrates the important properties of different representation models to obtain more flexible and general model. The hybrid approaches usually perform very well in most of the cases as these tend to combine benefits of more than one model [96].
There is a fundamental need for a generally applicable context model which ensures the processability and consistency of context throughout applications. The available models are more of meta-models and do not give sensor specific methods of context extraction. Sensor based CI may be incorrect, incomplete, or inconsistent. From the studies done here, it has been observed that imperfection of CI is handled in very few systems. These problems, however, can be addressed in context extraction mechanisms instead of representation.

Reasoning or extraction mechanisms proposed to be used with discussed models are listed in last column of Table 3.2. Unified Modeling language based and first order logic based approaches and their variants are commonly used for extraction of context using models. Description logic and its generalized form in ontological reasoning are used to reason on models represented as markup languages and ontology.

Context models can represent CI in a formal manner to facilitate its sharing and interoperability across systems [97]. Reasoning mechanisms are useful for extraction of current contexts from knowledge encoded in models. Specific challenges to create and use sensor based CI is discussed in next section.

### 3.1.3 Sensors as Context Data Sources

The idea of using sensors for simple context detection has been existent since long [54]. Simple threshold based sensors in automatic doors and cut-off switches are examples of these. Sensors are becoming preferred sources for context awareness, as compared to other medium due to following advantages:

1. Can be deployed easily in harsh environments unreachable by human beings.
2. Can provide detailed micro level information of the environment continuously for long time
3. Have convenient size and affordable cost for large scale deployment
4. Unobtrusiveness, due to wireless transmission and small form factor, makes them suitable for use with human beings
5. Easily acceptable by people, unlike video monitoring
6. Can be embedded in environment, integrated into everyday objects and worn by humans.

Scores of sensor-based devices have already become part of our day to day lives.

Sensors can be used to capture a broad range of CI:

- **Environment**: temperature, humidity, barometric pressure, light, and noise level in an ambient environment; usage of electricity, water, and gas
- **User**: Location, schedule, motion data like acceleration of different parts of body, and physiological data like heart rate and blood pressure etc.
- **Interaction**: With objects in surroundings.

<table>
<thead>
<tr>
<th>Physical Environment Sensors</th>
<th>Physiological Sensors</th>
<th>Logical Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Body Temperature</td>
<td>Time of the Day</td>
</tr>
<tr>
<td>Humidity</td>
<td>Heart Beat Rate</td>
<td>Date</td>
</tr>
<tr>
<td>Pressure</td>
<td>Blood Pressure</td>
<td>Scheduled Event / Activity</td>
</tr>
<tr>
<td>Ambient Light</td>
<td>Accelerometers</td>
<td>Diary Entries</td>
</tr>
<tr>
<td>Ambient Sound</td>
<td>ECG</td>
<td>Age</td>
</tr>
<tr>
<td>Location</td>
<td>EEG</td>
<td>Gender</td>
</tr>
<tr>
<td>CO2 Sensors</td>
<td>RFID Tags for Identity</td>
<td>Information of co-residents</td>
</tr>
<tr>
<td>Camera</td>
<td>Passive Infrared Sensors</td>
<td>Visitor Information, if any</td>
</tr>
<tr>
<td>Microphone</td>
<td>Orientation</td>
<td>Medical Histories</td>
</tr>
<tr>
<td>Illumination</td>
<td>Skin Conductivity Sensors</td>
<td>Any other habit related information</td>
</tr>
</tbody>
</table>

Table 3.3 lists few common physical and logical sensors to sense contexts relevant to human monitoring applications. Wearable sensors provide physiological sensing like ECG, blood pressure, heart rate and pulse oximeter etc. Other sensors are worn on body and installed in objects to classify daily activities of the patient. Environmental sensors provide a spatial context and enable location tracking. Logical contexts directly derivable from sensors are listed in Table 3.4.

Different sensors produce different types of sensor data, including binary, continuous numeric, and featured values. The type of data will have an impact on techniques chosen to analyze the same. A binary value is the simplest type of sensor data: true (1) or false (0). RFID sensors produce a binary reading: an object with an RFID tag is detected by a reader or not. Continuous
numeric values are produced by most sensor types, including positioning sensors, accelerometers and all the ambient sensors

Table 3.4: Some Contexts and Related Sensors

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Contexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Weather</td>
</tr>
<tr>
<td>CCTV</td>
<td>Video Monitoring</td>
</tr>
<tr>
<td>Light Sensor</td>
<td>Light level, frequency</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>Tilt, acceleration</td>
</tr>
<tr>
<td>Moisture</td>
<td>Dampness / dryness</td>
</tr>
<tr>
<td>Water-level</td>
<td>How full / empty a container is</td>
</tr>
<tr>
<td>Switch or button</td>
<td>If something is touching / pressing it</td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature</td>
</tr>
<tr>
<td>Light</td>
<td>Light / dark</td>
</tr>
<tr>
<td>Pressure</td>
<td>Pressure (e.g. someone standing on it)</td>
</tr>
<tr>
<td>Movement</td>
<td>Specific movement patterns</td>
</tr>
<tr>
<td>Proximity</td>
<td>How close / far something is</td>
</tr>
<tr>
<td>Contact Sensors</td>
<td>For knowing status of a device e.g. open / close</td>
</tr>
</tbody>
</table>

Featured values are typically produced from relatively more sophisticated sensors such as a camera. These values are more difficult to process autonomously by machine for context extraction.

Identification of contexts of interest and appropriate sensors is critical to context extraction for an application. Methods of sensors to context transformation are discussed in next subsection.

3.1.4 TECHNIQUES FOR EXTRACTION OF CONTEXT FROM SENSOR DATA

Availability of different types of sensors is instrumental in extracting rich CI. Contexts derivable from a single sensor are straightforward to extract by simple threshold based method. For extraction of contexts derived from more than one sensors(similar or heterogeneous) advanced methods will be required. For example, physical context of a person can be obtained by data from many accelerometers worn on various parts of human body defining locomotion [118]. The location context can be determined by fusing data from sensors like GPS, ambient light and ambient noise etc. [119].

Fusion of sensor data as contexts requires consideration of their heterogeneous characteristics and capabilities. Data from diverse sensors exhibits high complexity (different modalities, huge volumes, and inter-dependency relationships between sources), dynamism (real-time update) and
uncertainty (in accuracy, precision and timeliness) [120 - 121]. Wireless transmission of raw data by sensors invariably introduces unwanted noise in it. This leads to uncertainty issues in sensor data, which can be out of date, incomplete, imprecise and contradictory. Noisy sensor data may result in misunderstanding of context, which will lead to incorrect application behavior. In presence of sensor uncertainties, conventional models of context modeling, reasoning and data mining that work well on complete data may become unusable [122].

Many sensor based context awareness techniques are enabled by machine learning probabilistic methods [95]. Simple threshold based rules can be used at sensor nodes for providing low level contexts like symbolic location or classify temperature as hot/cold. For handling source uncertainty, various approaches used in literature are probabilistic logic, fuzzy logic, Bayesian networks, Hidden Markov Models and Dempster- Shafer theory of evidence [123]. For exploratory data analysis of unknown unlabeled data, unsupervised classification algorithms like Kohonen Self Organizing Maps (KSOMs) have been used [102].

Various methods used by different researchers and their purpose are summarized in Table 3.5. The applications discussed in first half of Table 3.5 use context awareness to improve functioning of any of the basic protocols of WSN. This has been termed as “smart sensing”. The second half presents extraction of user relevant contexts from sensor data to make applications context aware and thus “smart”.

The first half of table provides references on the protocol level introduction of context awareness to improve the functioning and lifetime of sensor networks. [48] used a hierarchical architecture for energy saving routing and communication plan by having a context aware duty cycling plan. One approach used by Elnaharaway is prediction of data generation [124]. They take advantage of density of nodes and overlapping of sensed information to predict missing values as well as in pointing the reasons for information collected contrary to hypothesis.

The second half introduces current work in designing novel applications that can be built upon sensor based context awareness along with algorithms enabling context processing. WSNs are useful for monitoring applications.
Table 3.5: Examples of use of various machine learning methods in WSNs

<table>
<thead>
<tr>
<th>Smartness Aspect</th>
<th>Method used</th>
<th>Purpose of study &amp; Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Sensing</td>
<td>K-Self Organizing Maps</td>
<td>Intelligent Routing [125]</td>
</tr>
<tr>
<td></td>
<td>Swarm Intelligence</td>
<td>Adaptive Routing [126]</td>
</tr>
<tr>
<td></td>
<td>Fuzzy-ART neural network</td>
<td>Maximize Network Lifetime [127]</td>
</tr>
<tr>
<td></td>
<td>Self-Organizing Maps (SOMs)</td>
<td>Intelligent Aggregation [47]</td>
</tr>
<tr>
<td></td>
<td>Fuzzy logic control</td>
<td>QoS Management [128]</td>
</tr>
<tr>
<td></td>
<td>Bilinear pairings</td>
<td>Clock Synchronization in WSNs [129]</td>
</tr>
<tr>
<td>Sensing for Smartness</td>
<td>Context Awareness</td>
<td>Context Aware Sensor Grid for Agriculture [130]</td>
</tr>
<tr>
<td></td>
<td>Pattern recognition and learning</td>
<td>Continuous Remote Health Monitoring [131]</td>
</tr>
<tr>
<td></td>
<td>OWL- a method for defining ontology</td>
<td>Intelligent Meeting Room [132]</td>
</tr>
<tr>
<td></td>
<td>Back Propagation Methods</td>
<td>Intelligent Vehicular Networks for Telematic Apps [133]</td>
</tr>
<tr>
<td></td>
<td>Temporal Patterns</td>
<td>Ambient Intelligence [13]</td>
</tr>
<tr>
<td></td>
<td>Multi Agent Systems</td>
<td>Commercial Lighting Control [134]</td>
</tr>
<tr>
<td></td>
<td>Hidden Markov Models</td>
<td>Real Time Daily Activity Classification [135]</td>
</tr>
</tbody>
</table>

A monitored situation can be judged remotely by physiological data and external surrounding factors. Sensors act as input to know the context of the user. Actuators make it possible to perform actions in the physical environment of a user.

Remote monitoring for precision agriculture [130], intelligent vehicular networks for telematics [134] and commercial lighting control [133] are examples of outdoor applications in which machine learning based monitoring solutions have been proposed. Major category of indoor applications is in the field of ambient intelligence. [13] suggested use of temporal patterns to implement it. In office environments, an intelligent meeting room application can enable lot of savings in energy. One such application has been implemented by [132] using ontology based model. Automatic identification and classification of human intent is a challenging application. Many researchers have approached it as current activity classification using temporal methods like Hidden Markov Models [135].

Processing methods for context extraction listed in Table 3.5 has been dealt in literature by two methods namely unsupervised, in which there are no known labeled data and supervised in which example data with ground truths are available and future patterns are to be identified and
labeled. KSOMs and Naïve Bayes Classifiers are found to be useful methods in WSNs for unsupervised and supervised learning respectively [136].

In all these applications, raw sensor data was transformed in application specific ways. There is hardly any algorithmic approach for mapping of sensors to specific contexts. Finally, most models don’t support hierarchical abstractions. Challenge undertaken here is to design a common set of abstractions as contexts such that similar machine learning methods can be used to extract these abstractions from different sensors. However, at sensor level processing methods may differ. The right type of sensor and accuracy of sensor data are critical to context extraction. Moreover, following issues make sensor based context extraction challenging:

- Sensors emanate data of multiple rates and types.
- Raw source data does not directly give desired information. The context aware application need to have a provision for mapping raw data from one or more sensors to define a context.
- Uncertainty inherently gets introduced due to wireless transmission and fidelity issues of sensors.
- Same sensor data can be interpreted to different contexts according to the requirements of application.
  - For example, based on the location data for a number of users, we can define
    - user-centered contexts (meeting — the users are gathering in a meeting room)
    - location-centered contexts (occupied — a room is occupied).
- The data sources may be transient. At some times during operation some of them may not be available.

The objective of context extraction mechanism proposed in this thesis has been to address these identified concerns.

CI obtained from sensors is abstract and more comprehensible than raw data. However, the complete overall picture is still not representable by individual contexts. In next section, concept of situations and methods to deduce them from available contexts has been discussed.
3.2 SITUATIONS AND COMPUTATION OF SITUATIONS

Contexts are not as intuitive to humans as situations [137]. CI should be interpreted into a higher, domain-relevant concept called situation, which is an abstract state of affairs interesting to applications.

It was in early eighties when Jon Barwise and John Perry coined the term situation semantics and started publishing research work under the title situation in logic [138]. They worked towards providing a logical and formal basis for reasoning about common-sense and real world situations, particularly for theoretical linguistics, philosophy and natural languages. Later situation descriptions gained significance in the field of Human – Computer Interaction (HCI) [21] and then in pure computing applications.

Role of situation abstraction in data fusion has been identified in data fusion models. Situation awareness has been kept as second level processing in popular JDL sensor data fusion model designed for military communication [58]. Recently it has become significant in information fusion research for civil human oriented commercial applications also [139]. Methods are being developed for machine based automatic situation recognition from available information.

3.2.1 Definition of Situation

Simply speaking, situations are descriptions of “What is happening right now”? For a human, answers to “Where he is and what is he doing” can be good indicators of his current situation. • Merriam-Webster dictionary puts definition of situation as “relative position or combination of circumstances at a certain moment”.

For computing purposes, many researchers have defined situations to formalize them. Few such definitions are:

• McCarthy, 1969 [140]: “A situation is a finite sequence of actions.”
• Endsley, 1988[23]: “the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future”
Dietrich, 2003[24]: “...extensive information about the environment to be collected from all sensors independent of their interface technology. Data is transformed into abstract symbols. A combination of symbols leads to representation of current situations...which can be detected”

Yau, 2006 [25]: “A situation is a set of contexts in the application over a period of time that affects future system behavior”

McCarthy [140] defined situation as a sequence of actions. Such assumption may not hold in real life situations. Occurrence of situation may be characterized by a particular set of contexts but not necessarily in a particular sequence. Similarly, Endsley and Yau [23][25] associated duration with particular situation. This assumption is again considered against the dynamisms of real life situations. A situation may occur for different duration in different scenarios. Definition by Dietrich [24] confirms to the hierarchical information fusion paradigm. It talks about transformation of raw data to symbols which will be contexts in this case. Situation is ranked higher in abstraction than individual contextual information and defined as its combination.

Data from multiple sensors and their fusion to provide contexts in same time frame can be used for producing near real-time situation pictures. For computation purpose, a situation can be defined by collecting relevant contexts, uncovering meaningful correlations between them, and labeling them with a descriptive name. A logical expression of correlated contexts is called a logical specification of a situation. With this, situation abstraction bridges gap between sensor data and applications. For example, if the location context “studyRoom” conjuncts with an interaction context “keyboardAccessed” then their correlation form a logical specification of a situation with its descriptive definition as ‘working’, meaning that if a user is in the study room and accessing the keyboard of the desktop, he or she is considered in a ‘working_on_computer’ situation. An application can be defined on this situation; for example, adjusting the sound level of the background music. In remote patient monitoring, time spent in situations like entertainment, sleeping, exercising and daily routine contexts can be major indicators of health [141].

Formally, a situation can be described as \((c_1, \ldots, c_n) : \varphi\), where \(\varphi\) is a logical expression with free context variables \(c_1, \ldots, c_n\). Two situations can be regarded as mutually exclusive from
each other if they cannot co-occur at the same time in the same place on the same subject; for example, a user cannot be in a cooking situation and a sleeping situation at the same time.

The situation of interest may be composed of contexts describing state of the physical space, the users in it and the computation resources. Some of the issues in defining a situation in terms of its contexts are:

- How to form a situation’s logical specification, which can be acquired by experts or learnt from training data?
- How to infer situations from a large amount of imperfect contextual data; how to reason on situations’ relationships?
- How to address uncertainty of real life situation composition?
- Availability of only subsets of contexts may lead to ambiguity in situation classification.

The situation definitions mechanisms thus need to address following issues: minimize false positives and negatives during situation recognition in real time, provide consistent descriptions across various context uncertainties and be fast enough to provide real time processing. Formal methods of situation modeling for sake of automatic extraction are discussed in next subsection.

### 3.2.2 Modelling and Reasoning about Situations

Situation modeling is about providing a simple, human understandable representation of sensor data to applications, whilst shielding end users from the complexities of sensor readings, sensor data noise [142]. Situations have been modeled using specification-based approaches [143 - 145]. These approaches represent expert knowledge as logical rules and apply reasoning engines to infer situations from current contexts. Specification based methods work well when contexts are easy to interpret and the relationships between contexts and situations are easy to establish.

With their powerful representation and reasoning capabilities, ontologies have been widely applied for modeling situations using specifications [146 - 147]. Ontologies provide a standard vocabulary of concepts to represent domain knowledge, specifications and semantic relationships of application specific situations.
Specification based situation modeling is feasible only for limited number of contexts. For large number of contexts and larger training data, situations can be modeled using data based machine learning approaches. A series of Bayesian derivative models are popularly applied, including naïve Bayes, Bayesian networks and Dynamic Bayesian Networks [148]. Hidden Markov Models and Conditional Random Fields have also been found useful in encoding temporal relationships among CI [149].

Inspired from language modeling, grammar-based approaches like context free grammars are applied in representing the complex structural semantics of processes in hierarchical situations [150]. Decision trees, Neural Networks, and Support Vector Machine methods built on information entropy have also been used to classify contexts into situations [151 - 153].

Even though the above learning techniques have achieved good results in situation identification, they need a large amount of training data to set up a model and estimate their model parameters. When training data is not easily available, specifications are obtained using web mining techniques to uncover the common-sense knowledge between situations and objects by mining the online documents; that is, what objects are used in a certain day to day situation and how significant the object is in identifying that situation. Some unsupervised data mining techniques including suffix-tree [103] and Jeffrey divergence [154] have been applied to this domain. Other hybrid approaches for modeling situations in terms of contexts have also been proposed. [145] defines events from sensor networks and various situations to be composed of events. Situation representation is done in terms of Generalized Event Monitoring (GEM) language. GEM event trees and finite state machines were designed for recognition. Oh et al demonstrate an indoor context management framework to monitor home environment with multiple habitants [155]. They refer the (6W) - What, Who, When, Why, Where and How model of instantaneous surrounding description. This model is used as reference for definition of constituent contexts of situations in this work.

Thirunarayan et al. [156] explained synthesis of high- level, reliable information for situation awareness by querying low-level sensor data. Srivastav et al. [157] define framework for
abstraction of situations from objects and their usage information. The relational dependencies among objects are modeled as cross-machines called relational Probabilistic Finite State Automata (PFSA) and modeled using xD-Markov machine construction. These PFSAs are mapped to situations.

Other different works of enabling pervasive computing in general have also studied situations and their description mechanisms. Yau et al. in [25] analyzed the semantics of situations and gave them formal representations. They consider context as any instantaneous, detectable, and relevant property of the environment, the system, or users. An atomic situation is composed of contexts related as context operators, including function, arithmetic or comparison operators and time constraints. This helps application designers to specify situations using formal expressions and manipulate them.

[98] proposed representation of situations by decoupling the inference procedures of reasoning about context and situations from the acquisition procedure of sensor readings in context-aware systems. They apply logic programming approach to characterizing situations, which helps the system designer in naturally individualizing situation descriptions in an application. [99] studied classification of situations in terms of their composition. A situation can be an intrinsic, formal, or relational context situation, derived from a single, double and multiple pieces of context respectively.

Thomson et al [158] provide reusable library of situation specifications. They expressed different levels of granularity of a situation through specification inheritance. New specifications created as variations of existing ones enable interpretation of same situation at different levels of abstraction.

It is concluded from studies of situation modelling techniques that specification techniques typically require hand-crafting of situation specifications and inference rules and heavily rely on domain knowledge. Learning based approaches on the other hand require lot of costly training. It is thus proposed to develop a combination of specification- and learning-based approaches to support successful situation identification in a variety of environments and scenarios.
Conclusively, situation model should be such that it can be plugged with sensor to context ecosystem.

The goal of a situation representation model is to provide a theoretical foundation for building a situation-aware system with following capabilities:

1. Formally representing logical specification of situations
2. Verifying the integrity and consistency of situation specifications according to current context
3. Establishing the sufficiency of existing CI in describing target situations.

From our discussion of situation modelling approaches, a hybrid approach with representation clarity of specification based approaches and reasoning power of learning based approaches should be sought for unexplored environments.

3.2.3 Recognizing Situations From Contexts

Sensor derived CI is to be integrated for detecting situations. For example, contexts generated by internal condition of patient's body, e.g. physiological values like blood pressure, sugar level and heartbeat rate are to be seen along with external contexts like who is near him, objects being used, movement etc.

One of the main requirements of a situation model is to support online recognition from contextual data in real time [154]. A system may host a large number of actuations depending upon different situations. These actuations need to be triggered in time constraints. The situation model must not only be able to handle uncertainty, but also be descriptive about inference results; that is, what situations are most likely to happen while what situations are possible or impossible to happen [145].

The situation recognition problem is about searching a state space, where states are composed of set of CI defining situations. Main task then becomes to match the input stream with any of the states. There are at least two issues in this matching. First, set matching is not straightforward due to problems of partial matching and ordering. Second, presence of a context may not be either crisply true or false for a particular situation.
[144] extracted situations from video sensor contexts using regular grammars to match online abnormal behavior situation of occupant of building. Fuzzy logic, with its strength in dealing with imprecision and evidence theories like Dempster–Shafer theory has been used to solve the uncertainty problem of contextual composition. Christos [159] suggested use of fuzzy inference rules for situation determination from contexts.

Recognition of specific situations like “human activity in indoor environment” has been suggested by various authors [145] [160]. [145] propose a layered architecture of situation representation and recognition along with sensor management and communication protocols. Situations are represented as events and composite events. Finite state machines and distributed commitment machines are worked out as recognition algorithms. [160] formulated activity recognition as a pattern-based classification problem, and proposed an Emerging-Pattern based approach to recognize both simple and complex activities in a unified framework. They proposed a segmentation algorithm based on feature relevance to segment the boundary of two adjacent activities. Good classification performance on sequential and interleaved activities was reported.

[143] extracted situations from video sensor contexts, regular grammars are used to match online abnormal behavior of any occupant of building as suspicious situation. [28] described defining and using situation lattices for situation description and inference. Kasteren et al [149] addressed similar problem using Hierarchical Hidden Markov Model and Conditional Random Fields. [156] used prolog type reasoning to convey confidence in sensor data interpretation and eliminate inconsistencies in situation descriptions. An integrated actual implementation framework requires modules for converting sensors data to semantic web notations like XML. For reasoning, the inference engine is able to implement the rules in XML syntax or the data to be reasoned is to be brought in the form of PROLOG. Another pioneer work in the area of activity recognition using ambient sensors was reported in [161]. The study was done by deploying large number of simple sensors in real environments. The work has become baseline for many further developments in this domain.
[141] described situation recognition as classification problem, where input data source is a camera sensor and knowledge base of common sense of that field. The sensor data is segmented and classifications done on individual segments. The classified low-level labels are input to a domain specific “concept net”. A shortest path algorithm is then used to recognize high-level situation. Experiments were done in classifying a) general types of location, (like road, square, park, shop, cinema, mall, restaurant, gym. (b) Domestic (kitchen, living room, bathroom, bedroom, garden) (c) working environments (meeting room, office, corridor, leisure room)(d) vehicles used (bike, car, bus, train). The database and concept net are user defined and hence may not be complete.

[162] did multi sensor fusion for health care monitoring. Distributed inference was done using graphical models—that characterize the relationship between variables. They suggested three types of distributed inference at data, algorithm or decision level for activity prediction. Bayesian Framework for feature selection is used for feature reduction and outlier detection.

[146] gave a conceptual framework for situation characterization, abstraction, recognition, and projection from sensor data. Snapshot of all sensor data at a point of time is considered as scene and over a period is considered as episode. Further, the data is quantitatively processed as an event. The quantitative relations are aggregated as qualitative relations, which are strongly connected to activities.

[163] designed integrated systems with simple algorithms for events like noise, spike reporting instead of raw data. The decision about an event is taken within the sensor only. Before making a new physical sensor with these capabilities, a virtual sensor in a computer can be implemented to test the intelligence algorithms.

Situation recognition mechanisms unlike existing black box machine learning methods should be transparent and understandable by user. Besides computation, it should also detect the inefficiency of existing contexts to identify the target situations. These challenges were kept as objectives while developing C1 to situation extraction methods in this thesis.
3.3 INTEGRATION OF SENSOR DATA, CONTEXTS AND SITUATIONS

The ideas of context and situation abstraction will be leveraged and expanded in this thesis for problem of sensor data management and abstraction in HeWSN discussed in Chapter 2. It was concluded from the study of data fusion models that a hierarchical distributed system is suitable for scalable fusion.

However, the form of information at each level of hierarchy was not established. After study of definitions of Context and Situations in this chapter and their relevance in modern computing applications, it was realized that middle level fusion can be in form of “Contexts” and higher level can be in terms of “Situations”. As a result of this integration of concepts, sensor data will be managed in a hierarchical manner such that in the process, usefulness of data will be increased and its size will be decreased as shown in Figure 3.2 (a).

![Figure 3.2](image)

**Figure 3.2:** (a) Hierarchical Relationship between Sensor, Context and Situations (b) Form of Data at each level in reverse order

Situations are more meaningful, stable, and certain, so they are considered more crucial than individual pieces of context in determining automated actuation. The steps towards situation abstraction from sensor data are depicted in Figure 3.2 (b). In first step, processed data from various types of sensors is gathered, which is organized as contexts to represent abstracted information from the domain under study in the second step.
As we do these abstraction steps, the size of the data to be handled is reduced and usefulness towards automated decision making is increased.

### 3.3.1 Proposed Framework

The framework to implement proposed integration takes as input data from uncertain sources like sensors and outputs abstractions of interest as situations.

The flow of abstraction for some target set of situations to be recognized will be:

- Define or learn from data, the logical specification of situations as co-presence of several contexts.
- Define probabilistic mechanisms for automatic, real time extraction of middle level abstractions, that is, contexts from low level abstraction, that is, processed WSN data.
- Identify appropriate sensors to be used for context extraction.
- Design local processing methods for heterogeneous sensors to adapt the multi rate & type of data from each sensor type and achieve low level abstraction.

For environments already laced with sensors, steps described above will be done in reverse order. Based on the studies done on sensor data fusion models in last chapter and context, situation extraction models in this chapter, a need for unified model for seamless transformation of sensor to situation was sought.

A three stage logical model is designed as shown in Figure 3.3. Generic classes of algorithms considered to be suitable at each stage have been shown in framework. Various stage wise components to be developed are:

*For Stage 1:* Local Processing Algorithms for Sensors for obtaining low level abstractions

*For Stage 2:* Definition of Generic Context Model, Methods of sensor to individual context mapping, Inference mechanisms of contexts from uncertain sensor data

*For Stage 3:* Modeling of Situations in terms of contexts, Inference of Situations from Contextual Information – Soft and Exact

The proposed framework has a broad scope conceptually. It can incorporate any type of sensors. Following design goals need to be met while designing algorithms at each level:

1. Design a hybrid (specification cum learning based) approach for generic context to situation description.
2. Handle inherent uncertainty due to ambiguities of dynamic environments in composing situations.
3. Identify insufficiency of contexts to define target situations.
4. Reliable way of mapping raw data to contexts.
5. Online context matching from stream data.
6. Reliable with least human intervention: Minimum False positives, as actuation may be automatic.
7. Meet real time constraints: Applications like Military demand real time reporting of a situation.
8. Withstand uncertainty due to unavailability of context sources or missing data due to environmental factors.

**Figure 3.3:** Proposed Sensor Data to Situation Fusion Framework
All these issues are to be addressed while implementing the proposed framework. Designed or chosen algorithms for each stage and results obtained have been described in later chapters. The algorithms have been put to task on different real sensor network data where ground truths on contexts and situations are known. Data from physical sensors like light, temperature, pressure, wearable sensors, motion sensors and logical sensors like time of the day, duration etc. has been tested.

3.4 CHAPTER SUMMARY

Identification of contexts from sensor based devices without human intervention has been a research area since quite some time. For purpose of extracting and utilizing context, existing context models were surveyed in this chapter. It gave rise to few major findings. Firstly, available models are more of meta-models and don’t provide sensor specific methods of context extraction. There is no available approach for mapping data sources of specific contexts. Finally, most of the models don’t support hierarchical abstractions. Extracting only trivial contexts from data is not very useful for end user. Another abstract concept of automated situation recognition and situation modeling was also surveyed. Methods to represent and identify situations from contexts have been studied.

A framework has been proposed in this chapter to integrate these two concepts hierarchically to compute situations from sensors via intermediate contexts. Computational methods for real time determination of context values from sensors and situation values from contexts were also explored.

Few interesting questions to which the designed framework seeks answers are following. How are the sensors distributed? How to synchronize the format, type and accuracy of the collected data? Which sensors produce similar data and can be clustered? Fusion techniques of first and second stage consider answers to these questions and identify set of current contexts. These are used to find overall situation and confidence in it.

In subsequent chapters, the designed framework has been implemented and evaluated to answer these questions.