CHAPTER 3

RELEVANT CONCEPTS

This chapter presents the various concepts relevant to this research work.

3.1 Ontology Engineering

The Web is continuously evolving towards Web 3.0 after going through Web 2.0 (O’Reilly.T 2005), since it was created with hyperlinks and multimedia capabilities. The Semantic Web, which is the prime component of Web 2.0 and 3.0, is an evolving growth of the World Wide Web in which the semantics of information and services on the Web are being defined. This facilitates the Web to understand and satisfy the requests of people and machines to use the Web content (T. Berners-Lee et al. 2001).

“The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” (T. Berners-Lee et al. 2001).

“The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications” (M. Dean et al. 2004).
These definitions highlight the following:

- Well-defined meaning of the information
- Machines use the information (automation)
- Collaboration through data integration and reuse of data

The Semantic Web aims to enable machines to access the available information in a more efficient and effective way. The researchers and developers are working on two different choices towards the realization of their vision.

- In the first choice, the machine reads and processes a machine-sensible specification of the semantics of the information.
- In the second choice, web application developers embed the domain specific knowledge into the software which will enable the machines to perform assigned tasks perfectly.

3.1.1 Semantic Web Architecture:

The preceding ideas and standards to complete the Web are being put into practice under the supervision of the World Wide Web Consortium (W3C). To lessen the quantity of standardization required and increase reuse, the Semantic Web technologies have been organized into several layers revealed in figure 3.1. The two base layers are inherited from the former Web. The rest of the layers aim to build the Semantic Web.
The Semantic Web layers are organized following an increasing level of difficulty from bottom to top. Higher layers functionality relies upon lower ones. This design approach facilitates scalability and encourages using the simpler tools for the purpose at hand. All layers are detailed in the subsequent subsections.

### 3.1.2 URI and UNICODE:

The two technologies that verify this layer are directly taken from the World Wide Web. URI (Uniform Resource Identifier) offers global identifiers and UNICODE (Unique, Universal and Uniform Character enCoding) is a character-encoding standard. It supports worldwide characters.

In short, this layer offers the global perspective, already present in the World Wide Web, for the Semantic Web.

### 3.1.3 XML and Namespaces:

The Semantic Web should smoothly incorporate with the Web. Consequently, it must be mingled with Web documents. HTML (Hyper Text Mark-up Language) is not adequate to capture all that is going to be expressible in the Semantic Web. XML (eXtensible Markup Language) is a superset of HTML that may be used the serialization syntax for the Semantic Web. XML was first of all tried, but in recent times other possibilities have been developed.
The XML assists integrating Semantic Web documents in the current XML/HTML web. The other options are N-Triples and Notation 3 syntax\(^3\).\(^1\)

Namespaces were introduced in XML to increase its modularization and reuse of XML vocabularies together with XML Schemas. They are also used within the Semantic web for the identical purpose.

3.1.4 RDF Model and Syntax:

The RDF (Resource Description Framework) Model and Syntax (M. Dean et al. 2004) defines the building blocks to realize the Semantic Web. This is the prime layer that was specifically developed for it. This specification defines:

- RDF abstract syntax and
- RDF graph model.

The RDF graph model defines a structure consists of nodes and directed edges between nodes. The structure of nodes and directed edges conform directed graphs. Those directed graphs model the network of terms and associations between terms of the Semantic Web. The nodes and its relations are called resources and are recognized by URIs. Each node has its own URI and its relations also have its own URI. They are called properties. Figure 3.2 shows an example of RDF graph model.

---

\(^{3,1}\)http://www.w3.org/DesignIssues/Notation3.html
Particular edges are recognized by the *triad* composed by the source node, property and target node. *Triads* are named *triples* or *RDF statements* and they are the *RDF* abstract syntax.

Graphs can be arranged as a set of triples, one for each edge in the graph. Both representations are identical so that the graph model can be reconstructed from the set of *triples*. *Triples* can also be assigned as an explicit identifier that is an *URI*. This procedure is called *reification*.

### 3.1.5 RDF Schema:

*RDF* Schema extends *RDF* and is a vocabulary for describing classes and properties and of *RDF*-based resources, with semantics for generalized hierarchies of such classes and properties. Simple *RDF* gives the tools to build semantic networks (knowledge representation technology). Nevertheless, there is still a lack of several semantic network facilities not existing with *RDF*. There are no defined taxonomical relations. They are labeled in the *RDF* Schema specification (D. Brickley et al. 2004).

*RDF* Schema specification gives some primitives from semantic networks to specify metadata vocabularies. *RDF* Schemas implement metadata vocabularies in a modular way, like *XML* Schemas.
➢ **type:**

It is a property that links a resource to a class to which it pertains.

The resource is labeled as a member of this class and thus it has its features.

➢ **class:**

It is a set of things that share some features; they have a common conceptual abstraction. A class models the concepts present at the referential semantic level.

➢ **subClassOf:**

This property holds the taxonomical relations between classes. For Example, if class $B$ is a subclass of class $A$, then class $B$ has all the typical features of class $A$.

➢ **subPropertyOf:**

This property builds the taxonomy of properties. For Example, if property $B$ is a sub property of property $A$, then whenever it is mentioned that the property $B$ holds between two resources it can be inferred that $A$ also holds.

➢ **domain & range:**

Both are properties that relate other properties to classes. They constraint the classes to which the related properties can be connected.
Domain defines all classes to which the subject resource of the triples where property appears must belong. The same is applicable for range but constraining the object resource.

3.1.6 **Ontology:**

Ontologies are necessary when the expressiveness achieved with semantic network-like tools is not enough. Metadata vocabularies defined by RDF Schemas can be considered simplified ontologies. The tools included in this layer raise the developed vocabularies to the category of ontologies. Figure 3.3 shows ontology schema.

- **The Web Ontology Language (OWL):**

Semantic Web relies heavily on ontologies. Concretely, ontologies based on Description Logics (DL) paradigm include definitions of concepts, OWL classes, OWL properties, roles, and individuals.

The most common language to formalize Semantic Web ontologies is OWL a proposal of the W3C. The goal of this standard is to formalize the semantics that was created in old frame systems and semantic networks.

The Web Ontology Language (OWL) is a family of knowledge representation languages for representing ontologies. OWL gets its formal semantics from description logics (DL), a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally easiest way.
The *OWL* language consists of three sub-languages of increasing expressive power, such as *OWL Full*, *OWL DL* and *OWL Lite*.

- **OWL Full:**

  It is entirely compatible with *RDF* and it includes full *OWL* language primitives.

- **OWL DL:**

  It is the part of *OWL full* with the benefit of efficient reasoning support. However it has less compatibility with *RDF*.

- **OWL Lite:**

  It is subset of *OWL DL* with the advantages of easier implementation and understandability. However it has limited expressivity.

- **Annotations:**

  The next step of *OWL* is *OWL2* (B. C. Grau et al. 2008). Its applications often need ways to associate additional information with ontologies, entities, and axioms. To this end, *OWL2* provides for annotations on ontologies, axioms, and entities.

For example,

One might want to associate human-readable labels with *IRIs* (Internationalized Resource Identifiers) and use them when visualizing an
ontology. To this end, one might use the *rdfs:label* annotation property to associate such labels with ontology *IRIs*.

Various *OWL2* syntaxes, like functional-style syntax, provide a mechanism for embedding comments into ontology documents. The structure of such comments is dependent on the syntax and they are simply discarded during parsing. In contrast, annotations are "first-class citizens" in the structural specification of *OWL2*, and their structure is independent of the basic syntax.

Ontologies, axioms and annotations themselves can be annotated using annotations. Such annotations consist of an annotation property and an annotation value, where the latter can be anonymous individuals, *IRIs*, and literals. Annotation properties can be used to provide an annotation for an ontology, axiom, or an *IRI*.

- **Fuzzy *OWL2*:**

  *OWL2* can be used to represent fuzzy ontologies. Here the fuzzy ontology language is encoded using *OWL2* annotation properties. Fuzzy *OWL2* uses three different concepts such as *fuzzy concepts*, *fuzzy roles* and *individuals*. Fuzzy concepts represent fuzzy sets of individuals. Fuzzy roles represent fuzzy binary relations. Fuzzy *OWL2* uses the syntax of *RDF/XML*. Its main principle is to denote the predicates and nodes of the *RDF* triples through *XML* terms.
Fig. 3.1 The Semantic Web Layers

Fig. 3.2 Example of RDF Graph Model

Class/Concept
Conceptual entity of a domain

Properties
Attributes describing a concept

Relationship
Relationship between class/ concepts and properties

Axioms
Coherency description between concepts /properties / Relations via logical expressions

Fig. 3.3 Ontology Schema
Use of Ontology:

Ontology is used as a form of knowledge representation about the world or some part of it. As ontology specifies the concepts and relationships within a domain, it gives a consistent vocabulary for that domain and the relationships between those concepts.

At present, ontology is key source to many applications such as semantic web services, artificial intelligence, information management and integration systems, scientific knowledge portals and electronic commerce. It provides a shared common understanding of a domain and the means to facilitate knowledge reuse by different applications, software systems and human resources (A. Gómez-Pérez et al. 1998).

SPARQL (Simple Protocol And RDF Query Language):

It is a protocol and query language for semantic web data sources.

Crisp Ontology:

A crisp ontology is an exact (i.e., binary) specification of a conceptualization. In other words, it is an enumeration of the precise concepts and relationships that exist for any information assemblage. In crisp ontology, the domain knowledge (H. Ghorbel et al. 2010) is structured in terms of

- concepts ($C$)
- properties ($P$)
- relations ($R$)
• axioms (Ax)

It is properly defined as a 4 – tuple. That is,

\[ O = (C, P, R, Ax) \]

where,

*C* is a set of concepts specified for the domain.

*P* is a set of concepts properties

*R* is a set of semantic relations specified between the concepts in *C*

*Ax* is a set of axioms and it is a reasoning rule or a real fact.

### 3.1.7 Reasoner:

A reasoner is a prime component for working with *OWL* ontologies. In fact, virtually all querying of OWL ontology should be done using a reasoner. This is because knowledge in ontology might not be explicit and a reasoner is required to infer implicit knowledge so that the correct query results are obtained.

The *OWL API* (Application Programming Interface) contains the *OWLReasoner* interface for accessing *OWL* reasoners. In order to access a reasoner via the *API*, a reasoner implementation is needed.

The following reasoners provide implementations of the *OWL API* *Reasoner* interface:
- Chainsaw
- FaCT++
- JFact
- HermiT
- Pellet
- RacerPro (via OWLLink)
- FuzzyDL

To use any of these reasoners one must download the appropriate libraries and place them in the class path. All of these reasoners implement the OWLReasoner interface and provide implementations of the OWLReasonerFactory interface for instantiating the appropriate OWLReasoner implementation.

The commonly used reasoning techniques are ABox, TBox and RBox.

The TBox contains assertions about concepts such as, subsumption \((\text{Man} \subseteq \text{Person})\) and equivalence \((\text{Man} = \text{MaleHuman})\).

The RBox contains assertions about roles and role hierarchies \((\text{hasSon} \subseteq \text{hasChild})\).

The ABox contains role assertions between individuals \((\text{hasChild} (\text{John}, \text{Mary}))\) and membership assertions \((\text{John} : \text{Man})\).
3.2 Ubiquitous Technology

The advances in wireless communication technologies along with the development of computing technologies help ubiquitous computing to develop ubiquitous computing technologies. The notable acceptable technologies include wearable computers, mobile phones, actuators, wireless communication equipments, Radio Frequency Identifier (RFID) Tags & cards and Personal Digital Assistant (PDA). Following are the associated computing technologies identified for ubiquitous technologies: Situated Computing, Participated Computing, Spatial Computing, Temporal Computing, Intuitive Computing, Cognitive Computing and Interactive Computing. The ubiquitous computing technologies integrate computers in a seamless manner into the physical world. In addition, Sensor technologies such as QR Code (Quick Response Code), GPS (Global Positioning System), Light, Pressure & Humidity Sensors, Magnetic Sensors, Orientation Sensors, Gyroscope, Microphones, Accelerometers, Clocks, Cameras and NFC (Near Field Communication) are also used to acquire information in a ubiquitous environment.

3.3 Smart Classroom

In Smart class room activities are enhanced with the use of pervasive and mobile computing, sensor networks, A.I, robotics, multimedia computing, middleware and agent based software. They support,

- Video and audio capturing in class room.
- Lecture capturing enhanced with the instructors annotations.
3.4 Interactive Classroom

*Interactive class* room promotes collaborative education by proposing a virtual ‘whiteboard’ shared via network among the students through which they can exchange information or virtually attend lessons, without the need to be physically located in classroom’s physical space.

3.5 Student – centric Intelligent Classroom Using AmI

AmI has a significant potential of education by increasing student’s access to information, enriching the learning environment, allowing students’ active learning and collaboration and enhancing their motivation to learn. The intelligent classroom features a sophisticated, unobtrusive, profiling mechanism that facilitates the classroom’s students’ behaviour’s monitoring and assessment, in order to provide user related data to the classroom’s services and applications. This enables inference of potential correlations within various domains. Another feature is incorporation of educational content classification and archiving mechanism. The main feature, collaborative learning, offers both better learning experience and knowledge gain.

3.6 Teachers’ Support Tool for AmI Classroom

Teachers’ support tool in an AmI environment will perform curriculum activities, outweigh monitoring and assistance tasks especially in crowded classrooms. It uses an automated method of observing students’ behaviour and
identifying common problems is needed to enable effective and personalized tutoring. Further, it also uses a performance analysis tool that provides extensive metrics of student progress that the teacher can use to identify topics that require further studying or even adaptation of the teaching methodology. While designing such a tool the data management process should have to be effectively carried out.

3.7 Internet of Things

The IoT concept (Vermesan, O. & Friess, P. 2014) connects physical objects with one another to enable interaction among them and thus relieves the process of connecting end-user devices. IoT systems use cloud computing, low power sensors & data reception electronic devices and distributed data communication frameworks. The basic building blocks IoT systems should normally have embedded electronics, distributed intelligent systems, sensor technologies, automation techniques, cloud computing, applications & service profiles and M2M communication networks. In ICT education, people, process, data and objects are identified as four pillars of Internet-of-Things. Thus low-power electronics, big data analytic techniques and distributed communication technologies make the IoT more effective and demand.

3.7.1 Interactive Smart Board:

An interactive board (IB) is a large interactive display that connects to a computer. A projector projects the computer's desktop onto the board's surface where users control the computer using a pen, finger, stylus, or other device. Figure 3.4 shows an interactive smart board.
Interactive Smart whiteboard use:

- uses running software that is loaded onto the connected PC, such as a web browsers or other software used in the classroom.

- captures and saves notes written on a whiteboard to the connected PC

- captures notes written on a graphics tablet connected to the whiteboard

- uses an Audience Response System so that presenters can poll a classroom audience or conduct quizzes, capturing feedback onto the whiteboard.

3.7.2 Kinect Sensor:

The Microsoft Kinect (Miller, R. 2010) is a novel sensor device mainly used for gesture recognition. It is a horizontal bar connected to a small base with a motorized pivot and is designed to be positioned lengthwise above or below the display. The device consists of an RGB camera, depth sensor and multi-array microphone. It captures the full-body 3D motion, facial recognition and voice recognition abilities. It is developed using a software technology Rare of Microsoft Game Studios owned and designed by Microsoft, and on range camera technology by Israeli developer PrimeSense, which integrates a color camera, a depth sensor and a microphone array. The range of the depth sensor is
2.3-20 ft. which is restricted to 13 ft. by the SDK. The microphone array is comprised of 4 microphones, enabling sound source localization. It captures the position of the object using 3D space. It tracks 48 joints in real-time, at 30 frames per second. Figure 3.5 shows the front view of a Kinect sensor.

3.7.3 RFID Sensor:

Radio Frequency Identification (RFID) is a flexible, wireless, automatic identification technology that transmits information about an object or person, using radio waves. This is a technology that uses radio waves to transfer data from an electronic tag – called an RFID tag or label, which is attached to an object – through a reader for the purpose of identifying and tracking the object. Some RFID tags can be read from several meters away and beyond the line of sight of the reader. An RFID system consists of the following four components shown in figure 3.6:

- RFID Tag / Transponder
- RFID Reader
- RFID Antenna
- PC / Database
Elements of an RFID System

RFID systems have the following two basic elements: They are shown in figure 3.7.

RFID Reader (Interrogator):

RFID Reader is a device that is used to interrogate an RFID tag. The reader sends radio waves to the tag and then the tag sense and detects it and sends back its response containing all the information about the tag. The RFID Reader can read the tag depending on its antenna and the tag's antenna also on how the tag will be placed on the object being detected. RFID readers are usually on, continually transmitting radio energy and waits for any tags to respond to it. RFID readers come in many sizes.

RFID Tags (Transponders):

RFID tags shown in figure 3.8 contain tiny semiconductor chips and miniaturized antennas inside some form of packaging. They can be uniquely identified by the reader/host pair and, when applied or fastened to an object or a person, that object or person can be tracked and identified wirelessly and on the move. RFID tags come in many forms.

For example, some look like paper labels and are applied to boxes and packaging; others are incorporated into the walls of injection molded plastic containers; and still others are built into wristbands and worn by people.
Types of RFID tags

- **Active RFID tags** include on-board power source (miniature batteries) that are used to power the tag, and can transmit signals autonomously.

- **Passive RFID tags** don’t include an on-board power source and have power beamed to them by the reader.

- **Battery Assisted Passive (BAP) or Semi-passive RFID tags** require an external source to wake up but have significant higher forward link capability providing greater range.

- **Smart Tags**
  
  i. **Read only tags:** Information is programmed onto chip during manufacturing, no overwriting, and information constant, least expensive.

  ii. **Write Once Read Many (WORM) tags:** Information added only once along with unique identifier but can be read many times.

  iii. **Read-Write tags:** Open to data manipulation by user’s system without restrictions. It contains a unique identifier but carry an updateable memory for that to be added. It is expensive also.
Fig. 3.4 Interactive Smart Board

Fig. 3.5 Kinect Sensor

Fig. 3.6 Components of RFID System
Fig. 3.7 RFID Readers

Fig. 3.8 RFID Tags

Fig. 3.9 Wireless Speaker
3.7.4 Wireless Speaker:

Wireless speakers or powerful portable speakers shown in figure 3.9 are speakers which obtain audio signals through radio frequency (RF) waves rather than over audio cables. They are used to receive and transmit any voice data from smart phones, tablets and other devices.

3.7.5 Raspberry Pi Board:

Raspberry Pi board shown in figure 3.10 is a miniaturized involved computer of low cost, having ample processing speed and not bigger in size. It comprises of 1 GB of RAM, 1200 MHz quad – core ARM Cortex-A53 processor.

➢ Types of Raspberry Pi (RPi) models:

- The model A+ is the low-cost variant of the Raspberry Pi. It has 256MB RAM, one USB port, 40 GPIO pins and no Ethernet port.
- The model B+ is the final revision of the original Raspberry Pi. It has 512MB RAM (twice as much as the A+), four USB ports, 40 GPIO pins, and an Ethernet port.
- The Pi 2 shares many specs with the Pi 1 B+, but it uses a 900MHz quad-core ARM Cortex-A7 CPU and has 1GB RAM.
- The Pi 3 model B was launched in February 2016; it uses a 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU, has 1GB RAM, integrated 802.11n wireless LAN, and Bluetooth 4.1.
• Pi Zero is half the size of a model A+, with a 1Ghz single-core CPU and 512MB RAM, and mini-HDMI and USB On-The-Go ports.

➢ **Raspberry Pi Models features:**

• Operating Systems: RaspbianRaspBMC, Arch Linux, Rise OS, OpenELECPidora

• Video Output: HDMI Composite RCA

• Supported Resolutions: 640x350 to 1920x1200, including 1080p, PAL & NTSC standards

• Power Source: Micro USB

3.7.6 **Wireless Key Board & Mouse:**

A wireless keyboard and mouse are external computer devices which allow the user to communicate with computers, tablets or laptops using radio frequency (RF), infrared (IR) or Bluetooth technology. They are shown in figure 3.11.

3.7.7 **NFC (Near Field Communication):**

NFC stands for “Near Field Communication” as the name indicates, it enables short range communication between compatible devices. It is another standard for wireless data transitions. It involves at least one transmitting device, and another to receive the signal. The frequency transmission for data through
NFC is 13.56 megahertz, and data can be sent at either 106, 212 or 424 kilobits per second, which is quick enough for a range of data transfers from contact details to exchange pictures and music. NFC is shown in figure 3.12.

➢ **Types of NFC device:**

- Passive NFC devices include tags, and other small transmitters, that can send information to other NFC devices without the need of power source of their own.

- Active NFC devices are able to both sends and receive data, transfer and communicate with each other as well as with passive devices.

### 3.7.8 **U Smart LED light:**

Smart LED light shown in figure 3.13 is a lighting technology that allows lighting using internet-capable LED light bulb designed for energy efficiency. It can be scheduled and controlled remotely. It can be controlled through a mobile app with the integration of Wi-Fi, Bluetooth, and ZigBee.

### 3.7.9 **Renewable Energy Resource:**

This energy is sustainable and wholly inexhaustible. It leads to low power consumption. Figure 3.14 shows solar panel as one such resource.
Fig. 3.10 Rasperri Pi Board

Fig. 3.11 Wireless Keyboard & Mouse

Fig. 3.12 Near Field Communication
Fig. 3.13 U Smart LED light

Fig. 3.14 Renewable Energy Resource (Solar Panel)

Fig. 3.15 Bio-Metric Lock
3.7.10 **Bio-metric Lock:**

Bio-metric is a unique physical or behavioural characteristic which is used to automatically detect individuals. Biometric technologies capture, process and measure these characteristics electronically and compare them against existing records to create a highly accurate identity management capability. Figure 3.15 shows a bio-metric lock.

3.8 **Ontology based User Behaviour Modelling**

There are two ways of modelling users through ontologies. The first approach models the structure of the domain with its elements to represent the characteristics of atomic user. The second approach models the structure of a complex user profile with various dimensions of user’s state. Thus the first approach got its ideas from the overlay user modelling. The second one collected the main ideas from user modelling dimensions.

3.9 **Learning Management System**

A Learning Management System (LMS) is a software application or an online system which is used to administrate, delivery of documents such as electronic educational technology, tracks, generates report about education courses or a training programs.

LMS has a framework structure that handles all features of the learning process. An LMS is the infrastructure that provides educational content and manages it, finds and evaluates the individual learning or training objectives,
tracks the development to meeting those goals, and also presents learning contents for supervising the learning process.

LMS is a Web-based e-learning scenario which connects learning contents and learners together in a homogeneous manner. It is a wide range of system which manages trainings and provides educational records to software for distributing online or blended/hybrid college courses over the Internet with features for online collaborative learning. Colleges, universities, school districts, and schools use LMS’s to deliver online courses and enhance on-campus courses.

The use of centralized and integrated LMS contended with social software tools which supports a social constructivist approach to e-learning by providing students with personal tools and engaging them in social networks. Using social software with the integration of LMS provides a variety of separate tools to the students to self-govern their learning activities and events.

Most LMS’s are web-based software applications used to access to learning content and administrate the learning process. They are also used by many educational institutions to improve and provide online support to classroom teaching to a wide range of learners.

The important dimensions of Learning Management Systems are Student self-governance, training their workflow, providing on-line course materials, on-line assessment, learn from anywhere and at any time, centralized and collaborative learning. Internet-based LMS allow educators to run a learning system partially or fully online, asynchronously or synchronously. Moodle is a
free of Open Source Course or Learning Management System that provides blended learning opportunities as well as platforms for distance learning courses.

3.9.1 **LMS Functionalities:**

Following is the list of LMS functionalities:

- Delivery of Course Contents
- Students Administration and Monitoring
- Training, Scheduling and Tracking Events
- Prospectus and Certification Management
- Skills and Abilities Management
- Analysis of Skill Gap
- Individual Development Plan (IDP)
- Evaluating and subsequent provides the results
- Generating Report
- Management of Training Records
- Course Authorization
- Provision of needed Resources
- Effective Virtual Organizations
3.9.2 **Purpose of LMS:**

The key to understand the variations between LMS and other computer education terms is to understand the systemic nature of LMS. It is the framework structure that handles all characteristics features of the teaching and learning process.

An LMS is the online software application or an infrastructure that delivers and manages instructional content, identifies, assesses individual and organizational learning or training goals and targets. It also tracks the progress towards meeting those goals, by collecting and presenting data for supervising the learning process of the educational institution or an Organization as a whole.

Most LMS are web-based to facilitate access to learning content and administration. They are also used by educational institutions to enhance and support classroom teaching and offering courses to a larger population of learners. LMS’s are used by regulated companies, organizations and industries for acquiescence training.

Recent LMS providers include "performance management systems" to incorporate employee assessments, capability management, skills-gap analysis, progression planning, and multi-rater assessments (i.e., 360 degree reviews). Modern techniques now employ competency-based learning to discover learning gaps and guide course material selection for training.
3.9.3 MOODLE

MOODLE is a new and popular online course management systems (CMS) used by wide range of colleges and universities around the world that allows teachers to interact with students through the way of internet.

The word MOODLE is an acronym for,

Modular Object Oriented Development Learning Environment.

MOODLE is free and open-source software course or learning management system coded in PHP and developed on pedagogical ideologies. It is distributed under the GNU General Public License. MOODLE is used for collaborative learning, blended learning, online distance education, flipped classroom and other e-learning process in schools, universities, colleges, workplaces and other sectors.

It is also featured to create private websites with online courses for educators and trainers to achieve learning targets. MOODLE allows for extending and tailoring learning environments using community sourced Plug-ins.

➢ Features of MOODLE

- Designed to support both teaching and learning

Moodle delivers a powerful set of learner-centric tools and collaborative learning environments that allow both teaching and learning process.
• **Easy to use**

   It is simple interface, which has drag-and-drop features and having well-documented resources along with ongoing usability improvements which make MOODLE easy to learn and use.

• **Free with no licensing fees**

   Moodle is provided freely as Open Source software, by the GNU General Public License. Anyone can adapt, extend or modify Moodle for both commercial and non-commercial projects without any licensing fees and benefit from the cost-efficiencies, flexibility and other advantages of using it.

• **Always up-to-date**

   The Moodle is an open-source approach which means, it is continually reviewed and improved on to suit the current and evolving needs of its users.

• **All-in-one learning platform**

   Moodle provides the most flexible tool-set to support both blended learning and online courses. By Configuring MOODLE with enabling or disabling core features which easily integrate everything needed built-in features for a course. The external collaborative tools are forums, wikis, chats and blogs.
- **Highly flexible and fully customizable**

  Moodle can be modified in any way and personalized according to individual necessities. Its interoperable design allows developers to create new plug-ins and integrate external applications to attain specific additional functionalities.

- **Scalable to any size**

  Moodle can be scaled to support the necessities of both small and large organizations, because of its flexibility and scalability. It has been adapted for the use of education, business, non-profit, government, and community environments.

- **Robust, secure and private**

  Moodle can be easily deployed on a private secure cloud or server for complete control. It is constantly being updated and implemented to safeguarding data security and user privacy, security controls in Moodle development processes and software to protect against it against the unauthorized access, data loss and misuse.

- **Use anytime, anywhere, on any device**

  Moodle is web-based technology, with default mobile-compatible interface and cross-browser compatibility. Contents on the Moodle platform is easily to accessible and reliable across various different web browsers and technological enhanced devices.
**Extensive resources available**

Moodle allows to access extensive documentation and user forums in multiple languages, free content and courses shared by the Moodle users across the world, as well as wide range of plugins contributed by a large global community.

### 3.10 Modeling of Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) (Kennedy J and Eberhart R 1995) is a process of population based search procedure. It is also referred to as stochastic search procedure. It is one of the optimization techniques of Swarm Intelligence. It is the biological behaviour of swarms that forms the main motivation for the development of particle swarm optimization technique. PSO developers borrowed the swarming behaviour of swarms of bees, flocks of birds or schools of fish that have the objective of reaching a point to find food. It uses a population of individuals called *Particles* for solving the optimization problem. The population is known as *Swarm*. PSO has the following five principles with regard to population:

- proximity
- quality
- diverse response
- stability
- adoptability
**Proximity** ensures the ability to show effective memory and time consumption. **Quality** principle practices speed response to quality factors. **Diverse response** will make the population not to deviate from current objectives. The **stability** principle allows no change in the mode behaviour with respect to environment changes. The **adoptability** principle, on the other hand, will change the population’s behaviour mode for the case of genuine computational cost.

The particles in the swarm will be represented by a position vector and velocity vector. PSO generates a population of candidate solution and makes them to move around the search space based on velocity and position of particles. This technique uses fitness functions to optimize its objective function. Each particle will have a personal best position. A global best position vector will also be determined among the swarm. The velocity component and position component need to be changed iteratively to arrive on the optimum results. Thus this computational method will optimize a problem through iterations in order to obtain quality improved candidate solution.

The choice of fitness function depends upon the requirements of a specific optimization problem. The position of \(k^{th}\) particle in the population at \(t^{th}\) iteration of PSO algorithm is mathematically represented by vector \(X_k^{(t)} = (x_{k1}^{(t)}, x_{k2}^{(t)}, ..., x_{kn}^{(t)})\). Similarly, for the same case, the velocity is represented by vector \(V_k^{(t)} = (v_{k1}^{(t)}, v_{k2}^{(t)}, ..., v_{kn}^{(t)})\). The personal best position of the particle \(k\) at a state is represented by vector \(L_k = (l_{k1}, l_{k2}, ..., l_{kn})\). The global best position is represented by \(G_i = (g_1, g_2, ..., g_n)\). Thus the current particle’s position is denoted by \(x_k\) and the
The current velocity of the particle is denoted by \( v_k \). The personal best position of the particle is represented by \( l_k \).

The velocity function is made up of two components, namely *cognition component* (individual experience of a particle) and *social component* (social group behaviour). The cognition component is defined as follows:

\[
V_{kd} = V_{kd} + C_1 \cdot rand() \cdot (L_{kd} - X_{kd}),
\]

The social component is defined as follows:

\[
V_{kd} = V_{kd} + C_2 \cdot rand() \cdot (L_{gd} - X_{kd}),
\]

where,

- \( V_{kd} \) is the velocity vector of the \( k^{th} \) particle in dimension \( d \) of the problem space;
- \( L_{kd} \) is the \( l_{best} \) position vector of the particle in dimension \( d \);
- \( L_{gd} \) is the \( g_{best} \) position vector of the particle in dimension \( d \);
- \( X_{kd} \) is the current position vector of the \( k^{th} \) particle in dimension \( d \);
- \( C_1 \) is the positive acceleration constant to mean personal cognitive learning rate;
- \( C_2 \) is the positive acceleration constant to mean social learning rate; and
- \( rand() \) is a random real number from uniform distribution \([0,1]\).
The above two equations can be combined to get the velocity equation as follows:

\[ V_{kd} = V_{kd} + C_1 \cdot \text{rand}() \cdot (L_{kd} - X_{kd}) + C_2 \cdot \text{rand}() \cdot (L_{gd} - X_{kd}) \]

The position adjustment equation is as follows:

\[ X_{kd} = X_{kd} + V_{kd} \]

The process of PSO is outlined as follows:

To begin with, each particle of a randomly initialized population moves through the search space. The particles then store the good position they have seen. They share that information to each other. The particles make dynamic adjustment of their respective position and velocity according to good positions. Through iterative procedure, they arrive on the global best position.

The following steps brief the PSO method:

Step 1: Start with random initialized population

Step 2: Let each particle to fly through search space

Step 3: Make to remember best solution particle has seen

Step 4: Make to communicate about good positions with each other

Step 5: Make to adjust dynamically own positions and velocities based on good positions

Step 6: Make velocity adjustment through experiences & history behaviours of particles & neighbors.
Step 7: Conduct performance measurement of each particle using fitness function

Step 8: Let particles to fly towards an optimal solution

To understand the modeling of PSO method, first, the following concepts to be made clear in a simple manner:

- **Global Optimum**
- **Unimodel function & Multi-model function**
- **Multi-agent parallel search**
- **Uniform distribution**
- **Sigmoid function**
- **Neighborhood Topologies**
- **Global Best method**
- **Local Best method**

### 3.10.1 Global Optimum:

Optimization is a process of determining the best among the available under certain conditions. It includes tasks of minimization and maximization.

Maximization of a function $k \equiv$ minimization of its additive inverse $-k$.

There exists a different type of optimization like constrained Optimization, unconstrained optimization and dynamic optimization. The constrained optimization methods expect selected decision variables to satisfy the given
constraints. Unconstrained optimization methods do not put any constraints on the values of the selected variables of the problem. Dynamic Optimization methods use objective functions dynamic in nature i.e., they change over time. This changing nature of objective functions will make changes in optima positions.

Optimization problems can be solved by techniques like global and local optimization methods. The global optimization’s objective is to find a global minimum (or lowest) function value in order to determine its corresponding global minimizer in the search space.

The local optimization techniques aims at finding a local minimum in order to determine its corresponding local minimizer in the search space.

### 3.10.2 Unimodel Function & Multimodal Function:

The global minimum of a function is the smallest overall value of its whole range. The function may be unimodal or multi-model. The unimodal will have only one minimum. The Multimodal functions will have multiple local minima.

### 3.10.3 Multi-agent Parallel Search:

Here, the search space will have multiple peaks. It requires multiple agents to start search from various initial locations so that at least one agent should reach global optimum position. This search procedure allows all agents to communicate and exchange information among them to determine the global optimum position.
3.10.4 **Uniform Distribution:**

The uniform distribution will have constant probability of occurrence of all values of a variable. It is also called a *Rectangular Distribution*. If it has equally divided spaced intervals, then it will have an equal number of members of the data in each interval. The distribution is normally defined by its minimum and maximum values.

i.e., $U_m(x,y)$, $x$ being the minimum and $y$ being the maximum. It uses two different functions like *Probability density function* (PDF) and *Cumulative Distribution Function* (CDF).

3.10.5 **Sigmoid Function:**

It is a mathematical function. It has $s$ shape curve. Sigmoid functions use finite limits at infinity and negative infinity. They range from either $-1$ to $1$ or $0$ to $1$. They can be used as activation functions. Sigmoid curves can be used in cumulative distribution functions (for the case of $0$ to $1$). The sigmoid function is defined as follows:

$$S(t) = \frac{1}{1+e^{-t}}$$

This function has the characteristic of monotonically increasing with respect to the change in value of the variable.
3.10.6 Neighborhood Topologies:

These are all structures to describe the social interaction among the members of the searching process in the searching space. The interaction may be of less or more depending upon the size of the neighborhood. As far as convergence is concerned, it is slower for small size and faster for larger size. However, small size neighborhood will ensure quality in solutions.

Hence, it is better for the search process to begin with small neighborhoods and then gradually moves towards larger neighborhoods. Following are the some of the neighborhood topologies reported in the research literature and are shown in figure 3.16.

- **Star or gbest**
- **Ring or lbest**
- **Wheel**
- **Four Clusters**

The Star Topology allows each particle to be connected with each other. It ensures faster convergence. It is also called gbest topology for the fact that it facilitates more familiarity among all particles in the network.

The Ring Topology permits each particle to be connected only with its immediate nearest neighbors. The communication among the particles will be like
a relay process. Hence, the convergence of best result found is slower. It is also known as *lbest* topology.

In *Wheel Topology*, one particle is designated as a focal particle and it is connected to other particles. The communication is carried out only through the focal particle. This node aims to move to the best position through inference in order to communicate the same to the other nodes in the network.

In a *Four Clusters Topology*, there will be four clusters. They are also called *Cliques*. The neighboring clusters are connected with two edges and the opposite clusters are connected with one edge.

### 3.10.7 Global Best Method:

This method makes use of star social network topology in order to collect social information from all particles in the entire swarm. The logic of this method is as follows: each particle in the swarm will have a current position in the search space, a current velocity and a personal best position. The objective function for the *minimization problem* determines the best position with smallest value. The *personal best* position of a particle is a location in search space that will correspond to smallest value computed by the objective function.
Fig. 3.16 Neighborhood Topologies
The *global best* position corresponds to the lowest value of all personal best values determined. The *global best* method is understood through the following steps:

Step 1: Input initial population of particles

Step 2: Evaluate objective function values

Step 3: Initialize acceleration coefficients

Step 4: Initialize velocities of each particle to zero

Step 5: Set iteration number /* Loop starts here*/

Step 6: Compute *personal best* for each particle

Step 7: Compute *global best*

Step 8: Choose two random numbers in the range (0,1) from uniform distribution

Step 9: Compute new velocity for each particle

Step 10: Compute new position of each particle

Step 11: Calculate objective function value for each particle

Step 12: On satisfaction of terminal rule, *goto* step 5, otherwise,

Terminate iteration and display results
3.10.8 Local Best Method:

It uses the ring social topology to exchange social information of environment’s local knowledge within the neighborhood particles.

The following steps make easier to understand the logic of local best method:

Step 1: Input initial population of particles

Step 2: Evaluate objective function values

Step 3: Initialize acceleration coefficients

Step 4: Initialize velocities of each particle to zero

Step 5: Set iteration number

/* Loop starts here*/

Step 6: Compute personal best for each particle

Step 7: Compute local best

Step 8: Choose two random numbers in the range (0,1) from uniform distribution

Step 9: Compute new velocity for each particle

Step 10: Compute new position of each particle

Step 11: Calculate objective function value for each particle

Step 12: On satisfaction of terminal rule, go to step 5, otherwise,

Terminate iteration and display results
3.10.9 Understanding Parameters that Affect the Performance of PSO Algorithm:

Following are the identified parameters that are having impact over the efficiency of the PSO algorithm:

- **Number of Particles**
- **Number of Iterations**
- **Velocity Components**
- **Acceleration Coefficients**
- **Inertia Weight**
- **Velocity Clamping**
- **Velocity constriction**

The *Number of Particles* forms the Swarm size, also called *Population size*. Larger Size of Swarm will lead to more *computational complexity* per iteration. This in turn makes more time consuming also. The *number of iterations* plays the main role in determining reduction of *computational complexity* by neither preferring too large iterations nor preferring too low iterations.

Velocity adjustment of particle is the main intelligent computation that makes fast convergence. The velocity has three different components as follows:

- Inertia Component
- Cognitive Component
- Social Component
The inertia component helps to know immediate past movement direction of the particle and thus prevents unnecessary change of the direction of particles. The cognitive component learns the current performance of the particles with respect to their previous performances. This component helps the particle to make decisions on return to previous satisfied positions.

The social component is used to understand the performance of the particles with respect to their neighbors. It makes each particle to move towards the best position known from the neighborhood. There will be two acceleration coefficients and are associated with two random values. These two coefficients are constants and their main responsibility is to influence the particle velocity’s cognitive and social components. They represent respectively the extent of a particle’s confidence over itself and over its neighbors. The various properties of the constants are summarized in Table 3.1.

The *inertia weight* helps to ensure convergence. It manages a particle’s momentum by adjusting the influence of the previous velocities of the particle. The *velocity clamping* is a process of helping particles to reside within the boundary and to change the step size and search direction during its movement. The *velocity constriction coefficient* ensures convergence behaviour and excludes inertia weight and maximum velocity. It also prevents collapse.
3.10.10 **Strengths of PSO:**

- fast convergence
- uses limited number of parameters
- small impact of parameters to the solutions
- derivative free
- minimum dependency on initial points
- population-based nature
- can solve NP-hard problems in efficient manner
- Avoids minimum local value
- Globalized Search
- Effective to produce compact clusters
- Less Computational Efforts
- Self-adaptive dynamic parameter control
Table 3.1 Properties of the Constants - Summary

<table>
<thead>
<tr>
<th>Properties</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant1=Constant2</td>
<td>all particles continue to fly at their current speed till convergence</td>
</tr>
<tr>
<td>Constant1&gt;0 and Constant2=0</td>
<td>all particles behave independently</td>
</tr>
<tr>
<td>Constant2&gt;0 and Constant1=0</td>
<td>all particles move towards <em>global best</em></td>
</tr>
<tr>
<td>Constant1=Constant2</td>
<td>all particles move towards the average of <em>personal best</em> and <em>global best</em></td>
</tr>
<tr>
<td>Constant1&gt;&gt;Constant2</td>
<td>towards excessive wandering</td>
</tr>
<tr>
<td>Constant1&lt;&lt;Constant2</td>
<td>all particles to run prematurely</td>
</tr>
</tbody>
</table>

Fig. 3.17 Understanding Optimization Concepts through Ontology
Fig. 3.18 Understanding PSO Concepts through Ontology

Fig. 3.19 Understanding PSO Clustering Concepts through Ontology
3.10.11 **Ontology Representation of PSO Concepts:**

This section presents the ontological representation of PSO concepts in order to understand the relationship of various concepts easier. Figure 3.17 describes the overview of optimization process. Figure 3.18 helps to visualize different PSO concepts through protégé ontology tool.

3.10.12 **Ontology Representation of PSO Clustering Concepts:**

This section presents the ontology representation of PSO clustering concepts in order to make the understanding process easier. Figure 3.19 helps to visualize the relationship among the PSO clustering concepts through Protégé ontology tool.

3.11 **Chapter Summary**

This chapter summarized the relevant concepts of this research work. It first presented semantic web related concepts of Ontological Engineering. Then a detailed description of advanced Smart Learning Environment is reported. Finally, the Particle Swarm Optimization concerned methods are well illustrated.