CHAPTER 3
PROPOSED SCHEME

In an interactive environment, there is a need to look at the information sharing amongst various information systems (For E.g. Banking, Military Services and Health care). The specific situation is designed by:

i. Heterogeneities in hardware and software solutions.

ii. Heterogeneities in the structure, purpose and deployment.

Heterogeneities are typical in any other problem domain, where a variety of platforms, systems, databases and technologies etc exist. This system is not necessarily Web-Services based and uses XML as a standard communications language.

GIE System is proposed with a standards-based messaging engine, which is event-driven and provides foundational services for more complex software systems. It is both operating system and programming language independent and provides interoperability between different platforms; for example, between Visual Basic (VB), Java and .NET. It provides communication between disparate systems (healthcare) which can all connect to it regardless of the type of software or hardware used. The Proposed GIE System is also applicable to other domains for e.g.
3.1 Benefits of Standardized Data in Healthcare

Standardized data will not only benefit front line clinical staff but will also boost the performance of all sectors directly and indirectly connected to the health care process. The single most critical aspect in any healthcare system is data. Therefore, capturing data in a precise, standardized, and reproducible manner is critical to exchanging information accurately among widely distributed and differing users. If the data can be captured and stored in a consistent and unambiguous coded form, a consistent and reliable flow of information within and across organizations will soon follow [113]. A structured terminology covering all aspects of clinical care is required to help codify and standardize data.

i. Sharing and integrating information from different applications, medical records, and decision support systems.

ii. Querying information about populations of patients held across conforming organizations.

iii. Determining hospital performance and improvement in care through research analysis.
iv. Helping government health agencies to determine future care plans and quality assurance based on health statistics.

v. Making patients and the general public more aware of their health by access to their health records and general medical information.

It adopts XML to serve as a messaging syntax [114, 115 and 116]. There will be significant influence on healthcare institutes through advanced communications between different healthcare institutes which involves high level of interoperability. The GIE System is to provide foundational services for complex healthcare systems. The paper-based records can no longer meet the clinical necessities because of the complex healthcare systems. Computers have been used in healthcare institutes for decades to assist the integration and manipulation of patients’ data. Interoperability concentrates on the necessity to link up healthcare data. The EHR gives the ability to share patient records which smooth the progress of healthcare experts to advance the reliable and successful healthcare delivery.

This communication will improve accessibility of the patient record, so clinicians who require patient’s demographic or medical information are not bound by the limitations of time or site. Thus, clinicians will have a reliable 24 hour access to relevant patient data that can be available from different locations.
Different titles are laid at the heart of the evolution of EHR such as accessibility, standardization and interoperability. Its goal is to provide health data 24/7 from any healthcare institute. It reduces medical errors and improves quality care. It is based on HL7 standard for the exchange, management and integration of health data to generate EHR which is described as below.

3.2 TECHNIQUES ADOPTED

The HL7 Version 3.0 [117], CDA [118] and XML standards are integrated in the GIE System, which creates an EHR or Discharge Summaries for patients’ [119, 120].

3.3 PROPOSED ALGORITHMS

The following algorithms are proposed for integration and standardization of data in any domain is described.

3.3.1 DBX Algorithm

The DBX algorithm extracts data from the database. It generates XML document and allows an exchange in any domain.

**Input:** {Attributes} // set of attributes

**Output:** XML Document
**Step 1:** Connect to the relational database and execute by selecting a query to get the required data (Attributes).

**Step 2:** Create a new XML document object model (DOM) document tree ‘T’, in which the data will transfer to it.

**Step 3:** The first database element created in the XML document is called the "root" element.

**Step 4:** For each row: add a new element to the XML document, using the table name, then insert it into the XML document as a child of the root element.

**Step 5:** Loop through each column in the current row, and insert the field name, and corresponding value.

**Step 6:** Create a new element for the field and then insert it as a child to the current database row.

**Step 7:** Add the field value as a text node, and then insert it as a child element to the current field node.

**Step 8: Repeat from Step 4:** These loops do not terminate until they have processed every column of every row which has been retrieved from the database.

**Step 9:** Returns the completed XML document as a string.

**Step 10:** Insert the results into the XML document, and display it to the Client.

**Step 11:** End.
3.3.2 XDB Algorithm

The XDB algorithm takes XML document as input and maps its contents to target database attributes.

**Input:** XML Document

**Output:** XML Elements mapped to target database attributes

**Step 1:** Read the EHR Document (XML File) as String.

**Step 2:** The messages are parsed in the tree ‘T’.

**Step 3:** The first element found in the XML document is called the “Root” Node.

**Step 4:** Find child nodes in the XML document of the root node.

**Step 5:** For each row: read and count the no. of elements of XML document, in tree ‘T’ to construct RIM object.

**Step 6:** Loop through each node finds the corresponding attribute name and its value.

**Step 7:** Read map settings from database and place it in a queue.

**Step 8:** Place data in a temporary application table.

**Step 9:** Take ‘n’ no of variables which are equal to database fields count.

**Step 10:** Check for the target field and assign appropriate variable with the value.

**Step 11:** Creating connection to the database to map XML data.
Step 12: Repeat from Step 5: These loops do not terminate until all nodes are processed in the document.

Step 13: End.