Healthcare Data Validation and Conformance Testing Approach Using Rule-based Reasoning

By
Hira Javed
NUST201260767MSEECS60012F

Supervisor
Dr. Khalid Latif
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(March 2015)

Approval

It is certified that the contents and form of the thesis entitled \Healthcare Data Validation and Conformance Testing Approach Using Rule-based Reasoning" submitted by Hira Javed have been found satisfactory for the requirement of the degree.

Advisor: Dr. Khalid Latif

Signature: _____

Date: _____

Committee Member 1: Dr. Hamid Mukhtar

Signature: _____

Date: _____

Committee Member 2: Dr. Sarah Shafiq Khan

Signature: _____

Date: _____

Committee Member 3: Dr. Farooq Ahmad

Signature: _____

Date: _____

Dedication

With affection and gratitude, I would like to dedicate this thesis to my parents and teachers who have been remain a continuous source of inspiration and motivation for me and support me all the way.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by an-other person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Hira Javed

Signature: _____

Acknowledgment

In the name of Allah the bene cent and merciful, on whom we all are dependent for eventual support and guidance.

I would like to express my immense gratitude to my supervisor Dr. Khalid Latif, who has guided and supported me throughout my thesis work and al-low me to work in my own way and polish my skills. I highly appreciate his help in technical writing. His mentorship was predominant in sustaining versatile experience in my long term career goals.

I would also like to thank my committee members, Dr. Hamid Mukhtar, Dr. Sarah Shafiq Khan and Dr. Farooq Ahmad who continuously guide me in my thesis, and provide their valuable suggestions and encouragement.

Finally I would like to thanks my siblings, friends and colleagues who have lent a hand in order to complete my thesis.

Table of Contents

List of Tables

List of Figures

Abstract

HL7 community is profoundly involved in the development of standards in order to exchange, share and retrieve health related information. FHIR is an emerging standard of HL7 that encourages use of JSON as a data serialization approach. JSON is notoriously exible and schema-less approach compared to XML but it is very developer-friendly and concise in content presentation. Validation of healthcare data is crucial task because errors in this data can result in serious consequences that can lead up to increased mortality rate. Currently there does not exist a validator that can validate and conform data as per FHIR. This paper presents an approach to validate healthcare data embodied in JSON documents and to test its conformance with HL7 FHIR standard. We rst developed Description Logic (DL) based schema of the FHIR data model. JSON data is then translated to RDF and we apply rule-based reasoning to validate the data. As veri ed by results,design of the validation algorithm ensures that a validation step is performed in sub sec-onds and overall system remains e cient. Further, the rule based reasoning used to test conformance provides aid in identifying incompatibilities in data which should be xed to bridge the gaps in achieving interoperability.

Chapter 1

Introduction

This chapter gives the basic idea of the concepts involved in this research. It also presents the background and motivation for this study. Moreover, it provides an idea of expected results, and methodology to get and evaluate the results. Finally, it presents the structure of this thesis document.

1.1 Introduction

Healthcare data validation is a challenging and crucial task due to inherently complex nature of the healthcare data. Validation is necessary to ensure that data is in accordance with a content standard and that it can be processed without causing any erroneous implication to the receivers [1]. Conformance testing ensures that validated data is according to a standard template [2]. If data is not conformant with selected standard then it will not be stored for further processing and may be discarded. The validation as well as conformance ensure that correct information about a particular medical entity is exchanged with other hospital or clinic.

One of the ubiquitous challenges in healthcare domain is accurate information at the right time and at the right place. Ful lling this challenge means a seamless connection should exist among diverse systems. This seam-less connection is supposed to support vibrant communities that are going to exchange information so that they can e ectively use the exchanged information. Interoperability allows diverse entities or components to exchange information and then to use that exchanged information accordingly. Interoperability is widely discerned as fundamental necessity to achieve success in healthcare systems as it grants physicians to seamlessly share information with other healthcare providers irrespective of underlying technology stack and architecture. In order to improve healthcare and reduce cost, two or

more clinical entities are needed to exchange patient information. In doing so, two or more clinical entities exchange healthcare data using a standard data model.

Healthcare information is enormously complex and covers a variety of aspects including patient administration, organizational information, labo-ratory and clinical data. There exist diversity and richness of data; di erent models are designed to represent information making healthcare management a crucial task to achieve. Moreover there is a need to transfer accurate pa-tient information on time and in a consistent manner regardless of diverse nature of organizations that are going to exchange such information or one can say that in order to ensure proper care to patients, authenticated in-formation at the right time should be there. A lot of e orts are made, in this regard, by diverse organizations, resulting in several standards that can check the consistency and authenticity of patients information. In case of medical realm, management of same information of a patient among di er-ent healthcare organization is a crucial task. So semantic interoperability should be there to ensure homogenous information about the patient in di-verse hospitals or healthcare organizations. HL7 is an ANSI-accredited SDO (Standards Development Organization) that is involved in development and improvement of standards for interoperability of health information technol-ogy and is committing on development of single standard for single purpose. It de nes a standard format according to which information is transmitted. It facilitates the exchange of clinical data among di erent health systems. It confers standards for exchange, sharing and retrieval of clinical information that provides support in management, provision and evaluation of health re-lated data. HL7 brings forth messaging standards (HL7 v2.x and HL7 v3) as well as content standards and document structure (HL7 Clinical Document Architecture CDA) to ensure interoperability.

Standards play a vital role in data validation. Health Level 7 (HL7), is involved in development of standards for exchange of medical information among heterogenous hospitals or clinics. Fast Healthcare Interoperability Resources (FHIR) is an emerging HL7 content standard [3]. It is a resource based approach. Only resource state is needed to be exchanged instead of whole document as in Clinical Document Architecture (CDA) [4]. These re-sources are concise, brief and provide exibility in terms of extensions to solve issues of variability among heterogenous healthcare systems. Every resource contains human comprehensible part and metadata. It leverages strength of HL7 v3 and hides its underlying complexities and supports REST architec-ture. FHIR is in progress towards formal standardization; its speci cations are modeled in XML and JSON format. We have favored JSON instead of XML as information exchange serialization format in our work because

JSON is schema less approach and is widely accepted by developers community. The need of any system is to interchange data with other systems and JSON is designed speci cally for this purpose. It is language independent data interchange format that is easy to generate and parse and hence making it easy to use and understand. It is worth mentioning that JSON is widely accepted by developers community as a canonical data serialization format.

1.1.1 Validation

Healthcare data validation is necessary and crucial task to achieve because consequence of invalid and inaccurate data may results in poor quality of healthcare related malaise and other associated issues such as mortality. Due to inherent complex nature of healthcare domain and an immense increase in medical data day by day, makes validation of data according to some standard model, a challenging and crucial task to accomplish.

Validation is necessary to ensure that data is in accordance with standard followed and it can be processed without causing any erroneous state to another healthcare organization with whom it is exchanged. Conformance testing ensures that validated data is according to standard used that is FHIR in our scenario. If data is not conformant with FHIR then it will not be stored for further processing and will be discarded. This validation and conformance ensures that correct information about a particular medical entity is going to be exchanged with another hospital or clinic. JSON data is received and then is translated to JSON-LD. After its conversion into RDF rule-based reasoning algorithm is applied on this translated data to validate the data against standard data model (FHIR). This validation ensures consistency, accuracy and compliance with FHIR and can be used further for processing in di erent healthcare organizations or storage. This standard format de nes set of rules according to which information can be processed in a consistent manner

1.1.2 FHIR

FHIR, an acronym of Fast Healthcare Interoperability Resources, is an emerging HL7s message exchange standard. It is a resource based approach, and these resources are concise and provide exibility in terms of extensions to solve issues of variability among diverse healthcare systems. . It also provide ease for data exchange by o ering exibility of resources exchange in terms of single or aggregated resources. Further it allows pro le formation, as a set of resources can make a pro le. It leverages strength of HL7 v3

and hides its underlying semantic complexities and supports REST architec-ture, so is suitable for cloud computing. FHIR provides standard along with exibility to solve issues of variability among diverse healthcare systems by its extensibility mechanism. FHIR is in progress towards formal standard-ization; its speci cations are modeled in XML and JSON format. It ensures interoperability by providing standard and using standard terminologies that are going to be exchanged. Resource is the basic building block of FHIR. All content that is going to exchange is a resource. Every resource contains following properties as:

The same way to de ne and are built from data types that shows common patterns.

Human comprehensible

part Metadata

1.2 Motivation

FHIR is a resource based approach. Resources are brief, concise and are used for data exchange. Instead of whole document, as in CDA, just resources will be exchanged. It is a REST based approach i.e. resources are exchanged using http method, making it suitable for cloud applications FHIR overcome issues of existing interoperability standards and is easy to use approach but, its semantic model is missing and due to that data cannot be exposed or published in RDF. If we have semantic model or linked data, diverse healthcare organizations can share structured data on the Web.

1.3 Objective

The main purpose of this research is to map FHIR resources to semantic structure and then to implement FHIR using linked data principles. This mapping will provide aid in inferring. Further this linked data enables appli-cation developers to expose data using RDF and inference can be supported. Currently no semantic model for FHIR is available. Alignment with RIM and technologies is missing. As FHIR is there to ensure interoperability, so in order to make system FHIR complaint, incoming data is converted into RDF. After achieving RDF within speci ed time, it is processed and val-idated against the schema (ontology of FHIR). After getting validity and consistency, data is processed further. For evaluation, translated RDF is converted back into JSON.

1.4 Problem Statement

Validating FHIR data and then to test its conformance against standard model (FHIR) to ensure consistency of data.

As main aim of FHIR is to achieve seamless interaction or interoperability, so system should be FHIR complaint to work in an interoperable manner. The incoming data that is JSON in the current scenario should be compat-ible with FHIR data model so that it can be used in a useful manner and accurate data at the speci ed time can be made available for the patient. If the incoming data is validated against FHIR schema then it is FHIR com-plaint and can be processed further either in terms of its usage in healthcare organizations or for storage purpose.

1.5 Contribution

Currently there is no validator that can validate FHIR data. FHIR spec-i cations are available in JSON and XML. For JSON, no language schema exists. This paper presents an innovative approach for validation of FHIR compliant healthcare data embodied in JSON serialization format.Semantic model of FHIR's resources is developed in Web Ontology Language (OWL) [5]. In order to validate FHIR JSON against semantic model, JSON is converted into JSON-LD that is one of the RDF serializations. Afterwards rule-based reasoning is applied to validate the data against standard data model, FHIR in our case. The validation ensures consistency and compliance with FHIR and can be used further for processing in di erent healthcare organizations.

FHIR JSON is received and is converted into RDF for its further processing that is its validation against FHIR schema or FHIR ontology. All the attributes of JSON are matched with FHIR ontology and if validation report is true or data is consistent, then incoming data is FHIR compliant and it can be processed or stored according to requirement of the system.

1.6 Evaluation

As the incoming request JSON is translated into RDF and is processed according to demands of system and is validated against FHIR ontology, it is translated back into JSON. If the conversion is exact FHIR JSON, the one that is received, then it ensures that system is working in a correct manner.

1.7 Methodology

As pe proposed methodology, FHIR JSON is received and is passed on to translator. As schema for JSON doesn't exist so RDF based semantic structure of FHIR is developed and FHIR JSON is translated into JSON-LD. This JSON-LD is validated by consulting rules of ontology, constraints of primitive types and by applying reasoner. The response from system is in the form of valid data if FHIR JSON is received and invalid data if FHIR JSON is not received.

1.8 Expected Results

As per proposed methodology, if FHIR JSON is received then validator should give response in valid and consistent data and if other data is provided that is not FHIR data then validator should give response in terms of invalid and inconsistent data.

JSON-LD is tested by fetching rules from ontology, primitive data types, and applying reasoner on that data. After this nal result is provided. This overall translation process should be in sub-seconds so that it does not act as hurdle in validation of data.

1.9 Structure

Rest of the thesis is structured as follows:

Chapter 2: Background Information and Literature Review explains information about FHIR, existing Interoperability standards and ontologies.

Chapter 3: System Architecture shows the overall model of the system and ow of information.

Chapter 4: Ontology explains about semantic model formation and modelling challenges.

Chapter 5: Validation explains how the incoming data is validated against FHIR ontology.

Chapter 6: Evaluation explains the consistency of the system by getting back the same data after translation as that was received.

Chapter 7: provides results and conclusion.

Chapter 2

Background and Related Work

2.1 Data and Schema Validation

Data validation is used to check correctness and meaningfulness of data. It is meant to provide explicit guarantees for robustness, certainty and consistency for di erent kinds of input provided for user in an automated system or in an application.Continuous change of data by constraints related to dif-ferent users makes data validation a crucial task. To achieve quality of data, its consistency, completeness and correctness should be ensured.

Data validation measures accuracy of captured information and ensures that all data values are accurate and correct in an application. It imposes restriction on values in terms of numbers, time, and text etc. for preventing invalid data to be recorded and stored for future use . Further it ensures that data is valid for their contemplated data types and that data will remain valid around applications working.

If data veri cation is not performed prior to deployment of work ow then it may result in incorrect execution of process, inconsistency in data or suspension of the process [6]. According to Shanks et al [7], schema validation is critical in high quality system development. After constructing a conceptual model, there is a need to validate it with the requirements of stakeholders. If not validated properly, defects in the model might propagate to subsequent system design and implementation activities. If these defects are not discovered until late in the development process, they are often costly to correct.

Decker et al. [8] performed schema validation tasks and check satis ability of schema. In validation they checked whether a schema of database is suitable for intended requirements and needs. The authors de ne a distinguished view predicate for each speci c task of validation. An attempt for

task execution can be made by attempting to satisfy the request for insert-ing predicate corresponding to that task. Failed attempt to insert satis able shows schema unsatis ability.

2.2 Conformance Testing

ISO/IEC specify that conformance testing is attainment of a service, product and a process [9]. Conformance clause specify that all the require-ments should be satis ed in order to claim conformance. It veri es implemen-tation in order to determine deviations from de ned speci cations [10]. For conformance testing, there is no appropriate interface-based testing service. Software Diagnostics and Conformance Testing Division of National Insti-tute of Standards and Technology, develops and establishes testing methods and testing tools in order to improve quality of software and to check its conformance for a standard. Most part of their e orts are concerned with XML based messages [11]. Australian Healthcare Messaging Laboratory [12] contains a conformance testing service that is used to test a single message relevant to healthcare standard.During testing di erent areas of message are tested. Mostly this testing is concerned with structure and format of mes-sage. Actual content is validated through speci c databases or lookups. This testing assures that speci c business rules are adhered to.

Conformance testing shows whether a product or service is in compliance with the standard followed. It is characterize as a testing to nd out whether application conscientiously ful lls all the requirements for a given standard. It is concerned with evaluation and implementations external behavior and assessment of its adherence to speci c standard. It is performed to ensure that all requirements of a speci c standard are implemented in a correct manner. It is a type of functional testing where functionality of a program is tested from speci cations. Series of test cases are performed to nd out that the developed system is working according to de ned requirements or not. This testing is very important especially if data sources are heterogenous.

Prerequisite of conformance testing is clear speci cations [13]. Several standard based systems have their foundation to achieve goal of data valida-tion. However following of standards is not enough, conformance with those standards is required and is essential to ensure validity of data. Gebase et al. [14] followed conformance testing strategies for generally used healthcare data exchange messaging standard.They examined two strategies. In their rst approach they used an upper tester that tests the interface and lower tester that also acts as peer application and conduct testing. In their sec-ond approach they used actors that are small modules and run on separate

threads. They support subclass of functionality de ned by a standard. These actors interact with application being tested.

2.3 JSON vs XML Validation

Javascript Object Notation (JSON) is a serialization format in the form of attribute-value pairs for structuring data [15]. In order to interchange data, a light weight serialization format is required. JSON ful lls this requirement of being light weight and further it is language independent so is easy to use and understand and also provides ease in integration.

Di erent approaches are in use for JSON validation. Few online JOSN validators are available and JSON libraries are also available for use and integration in an application. JSON Lint [16] an open source project is a reformatter and an inline validator results in valid or invalid JOSN and if JOSN isinvalid then it will show type of error in JOSN. JSON lint pro [17] is also a JSON validator that provides ability to di erentiate among two datasets of JOSN [18].Further JSON formattor and Validator [19] is used for debugging of JSON documents.

2.4 Ontology Based Schema Validation

Logic reasoner is a software application that is used to infer consequences from set of axioms. These inference rules are speci ed by means of ontology language. Ontologies play a vital role in semantics by o ering accurate terms to de ne web resources.Reasoning over de nitions of web resources is essential to automate process of accessibility. Semantic web is an e ort that has been introduced by W3C so that explicit description of web resources can be provided. Semantic web came up with set of standards for exchanging machine understandable information. Among these standards RDF provides speci cations of data model and is XML-based serialization syntax.

OWL, a semantic web standard, enables de nition of domain ontologies and their modelling through object-oriented approach. Wang et al. [20] came up with context ontology and on the basis of that context ontology, they use logical reasoning to check consistency of information. They have imple-mented logic based reasoning schemes that reasons over low level explicit context to derive high-level implicit context . According to Bicer et al. [21], the most di cult task is to exchange information among heterogenous health-care systems. They have worked on developing a tool, OWLmt developed within ARTEMIS project. Exchanged messages are annotated with OWL

and then this message is mediated through OWLmt. This tool reasons over the instances of source ontology and generate instances for target ontology as per de ned mapping patterns.

Distinct range of inferences have been investigated for heterogenous DLs with varying expressivity. Expressivity can be determined by extent with which it allows concepts and roles description. Generally the increase in expressiveness came at cost of increase of complexity in terms of reasoning processes [22].Inspite of high complexity, optimized DL reasoning system was enforced based on tableau process. Most distinctly FACT [23] and RACER [24] are used. These systems shows that high worst case complexity would hardly be confronted. These implementations perform surprisingly well on practical applications. Another research is dedicated to light weight DLs, having limited expressivity but have high computational properties for certain reasoning tasks. Reasoning on large ontologies by using these DLs can be performed e ectively. Reasoning can e ectively be performed on large ontologies using these DLs. DLs is employed in several domains such as bio-medical, databases and other applications [25]. Success of DL is based on adoption of OWL (a DL based language) as a standard language of ontology for semantic [26].

In order to represent RDF several serialization formats are used. These formats include Turtle [27], N-Triples [28], N-Quads [29], JSON-LD, N3 [30] and RDF/XML. We have used JSON-LD. It is JSON based serialization format. JSON-LD is acronym for JavaScript Object Notation for Linked Data . It is a way of transferring linked data using JSON. Use of JSON-LD requires little e orts from developers to shift from JSON to JSON-LD [31]. JSON-LD allows data serialization similar to JSON [32]. It is recommended by W3C and is being developed by JSON.

It is de ned around "context" that provides further mapping of JSON and RDF. Context links object properties of JSON with concepts of ontology. While mapping syntax of JSON-LD to RDF, JSON-LD suppress values to a speci ed type or tag it with a language. Context can be placed in same le or can be placed in a separate le from where it can be referenced via HTTP link header.

Chapter 3

Proposed Framework

For validation of FHIR JSON against schema of FHIR data model, we have proposed an architecture of validator that validates FHIR JSON data. The validator takes FHIR JSON as an input and pass it to translator that translates FHIR JSON into JSON-LD. The JSON-LD based data is passed on to validator that validates structure, primitive types and cardinality restrictions of FHIR JOSN by using semantic structure of FHIR, FHIR primitive types in XSD and a resoner that reasons over the data. The validator then gives response in the form of valid and consistent data as per FHIR standard constructs. But if data is not according to FHIR data model then it will result in invalid and inconsistent data. The validation process is depicted in Figure 3.1. To demonstrate this process, an example of FHIR JSON is shown in Figure 3.2 and its translated JSON-LD is shown in Figure 3.3. Components of the system are described in detail in subsequent sections.

3.1 FHIR Schema in OWL

Ontology is an explicit and formal speci cation of shared conceptualization that shows how people perceive or think about things that are restricted to particular area. It not only enumerates factual domains but also provides aid in inference through axioms. Main purpose of ontology is to get meaning that is readable by machine so that accurate and automated reasoning can be performed. OWL describes the semantics of concepts / classes and their properties in documents. As ontologies are de ned in logic-based manner, so consistent, accurate and meaningful distinction can be performed among classes, properties and their relations. Due to ontologies shared viewpoint declaration is possible, providing support for di erent systems to communicate with each other. In order to capture the semantics for inference, rules

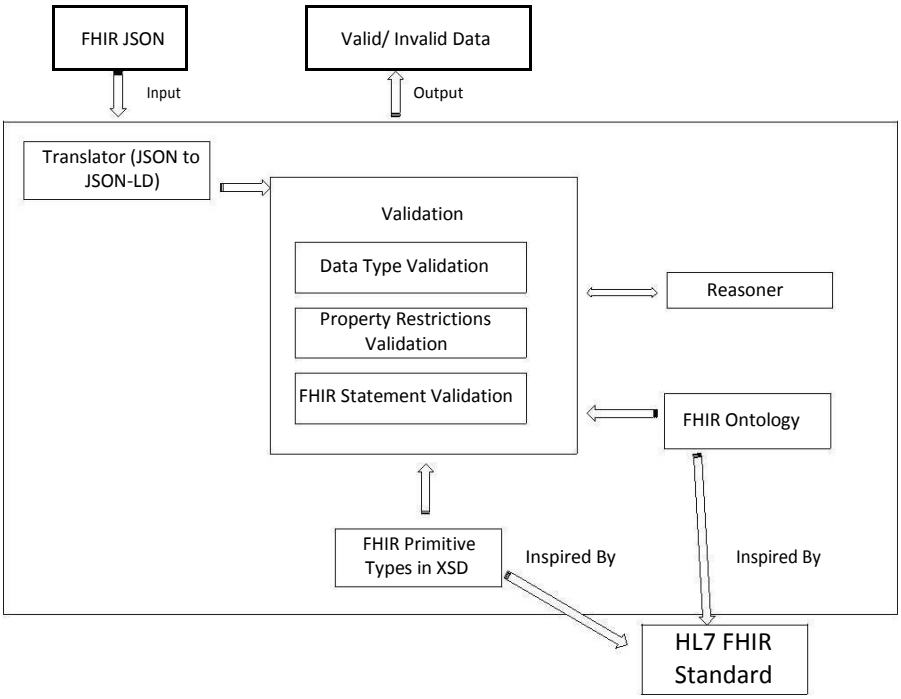

Figure 3.1: Validation Process

and constraints are needed in addition to factual knowledge. These rules can be used to generate new facts from existing knowledge, and to validate the consistency of knowledge. FHIRs semantic structure (ontology) is developed.

3.2 FHIR Resource

A FHIR resource of Practitioner is shown in Figure 3.4. Quali cation is its dependent resource and it can not exist with out parent resource i.e. Practitioner in current scenario.If we consider above mentioned resource of Practitioner then "dateTime" is primitive data type and Identi er is complex data type. Details of data types (primitive and complex are explained later).

```
{
  name :
Organization ,
publisher : FHIR
Project ,
  status : draft , date :
  2013-11-26 ,
}
```

Figure 3.2: FHIR JSON

3.3 FHIR Data Types

FHIR contains two kinds of data types. These include primitive and complex types. Primitive or base data types are prede ned types of data. There are prede ned set of values that can be assigned to primitive data type. Examples of primitive data types are shown in Table 3.1.

In ontology it is modelled as follows:

```
<xs:simpleTypename="oid">
<xs:restrictionbase ="xs:anyURI">
<xs:patternvalue="urn:oid:
        (0|[1-9][0-9]*) (\(0|[1-9][0-
        9]*))*"/>
<xs:minLengthvalue ="1">
</xs:restriction>
</xs:simpleType>
```

This is an example of string data type string. xs:restriction shows that values in FHIR string will be same as xsd string. xs:minLength shows that its value should be of at least one character or digit. Several complex types are also used in FHIR. These complex types are shown in Figure 3.5. All

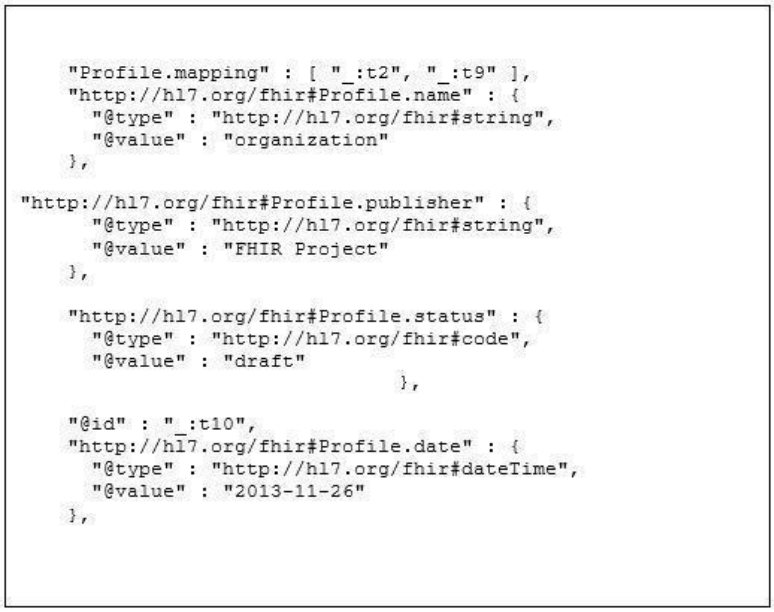

Figure 3.3: JSON-LD

complex types are derived from Element. Quantity can be measured in terms of Age, Distance, Duration, Count and Money.

Quantity, that is a complex type, has its own properties such as value, comparator. Possible values of comparator should be of code type that is a primitive data type. In ontology model, all complex types are implemented as subclasses of Element.

3.4 Ontology Hierarchy

All the resources are implemented as sub-classes of "Resource" and all the dependent resources are implemented as sub-classes of "DependentResource" followed by name of Resource. For example in Figure 3.4 dependent resource "Quali cation" is mapped as "Practitioner.quali cation".

This ontology is developed by following a modular approach. For example if we consider party registration, it represents functions necessary to manage, search, and access provider registry, independent of the underlying technol-ogy stack. Party registration repository represents various resources includ-ing provider demographics, organization demographics, provider groups and hospital organization (e.g. cardiology service group).Each ontology module in top level of hierarchy is imported in the lower level ontology modules as in Figure 3.7. Complex data types in FHIR are mapped as a separate ontology

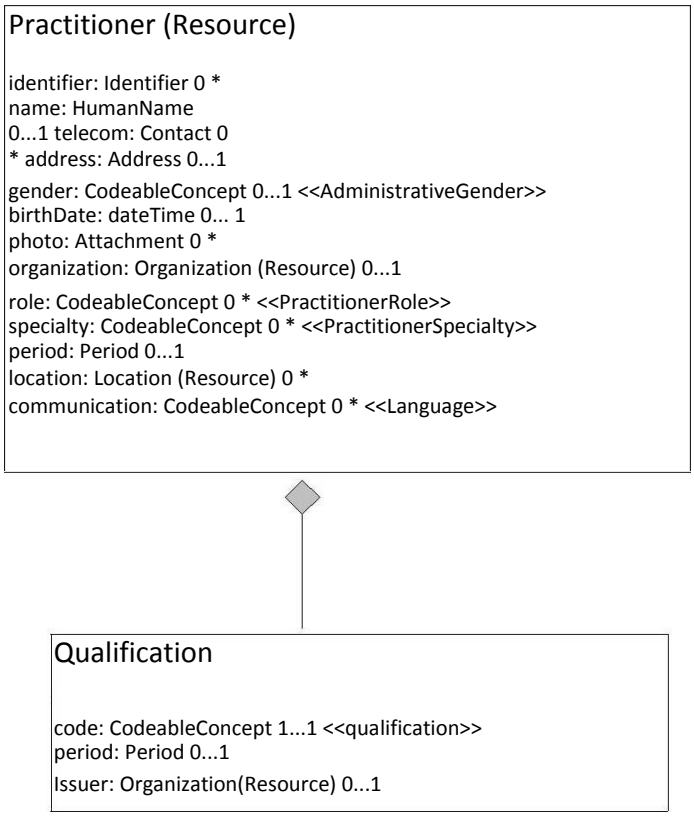

Figure 3.4: Practitioner Resource in FHIR

and this module is used in all the other modules in ontology hierarchy. Every module of ontology contains set of FHIR resources. These modules along-with their resources is shown in Table 3.2

3.5 Cardinality Restrictions

If we consider Figure 3.4 then in this gure, property identi er contains cardinality of 0 to many or 0 to *. name is having cardinality 0 to 1 and contained property code has cardinality of 1 to 1. Restrictions are modelled in the following manner in ontology

Primitive Type	Example Value
Integer	745
String	HL7 v2
dateTime	2002-09-24+06:00
Boolean	True/False
decimal	10.0
instant	2013-04-03T15:30:10+01:00
date	2012-05-24
base64Binary	dXNlcm5hbWU6cGFzc3dvcmQ=
uri	ldap://[2001:db8::7]/c=GB?objectClass?one
Code	draft
Oid	urn:oid:2.16.840.1.113883
uuid	urn:uuid:a5afddf4-e880-459b-876e-e4591b0acc11
id	12gh

Table 3.1: Primitive Data Types

1. 0 to *:
   FHIR Example:

   Identifier: Identifier 0 to * fhir:Practitioner.identifier rdf:type owl:ObjectProperty; rdfs:domain fhir:Practitioner ;

   rdfs:range fhir:Identifier.

2. 0 to 1: FHIR Example:
   name:HumanName 0 to 1

   fhir:Practitioner.name rdf:type owl:FunctionalProperty, owl:ObjectProperty; rdfs:domain fhir:Practitioner ;
   rdfs:range fhir:HumanName.

3. 1 to 1

   FHIR Example:
   Code:CodeableConcept 1 to 1

   fhir:Practitioner.Qualification.code rdf:type owl:FunctionalProperty, owl:ObjectProperty ;

   rdfs:domain fhir:Practitioner.Qualification;
   rdfs:range [rdf:type owl:Class ;
   owl:intersectionOf (fhir:CodeableConcept
   [rdf:type owl:Restriction;

   owl:onProperty fhir:Practitioner.Qualification;
   owl:maxCardinality "1"xsd:nonNegativeInteger])].

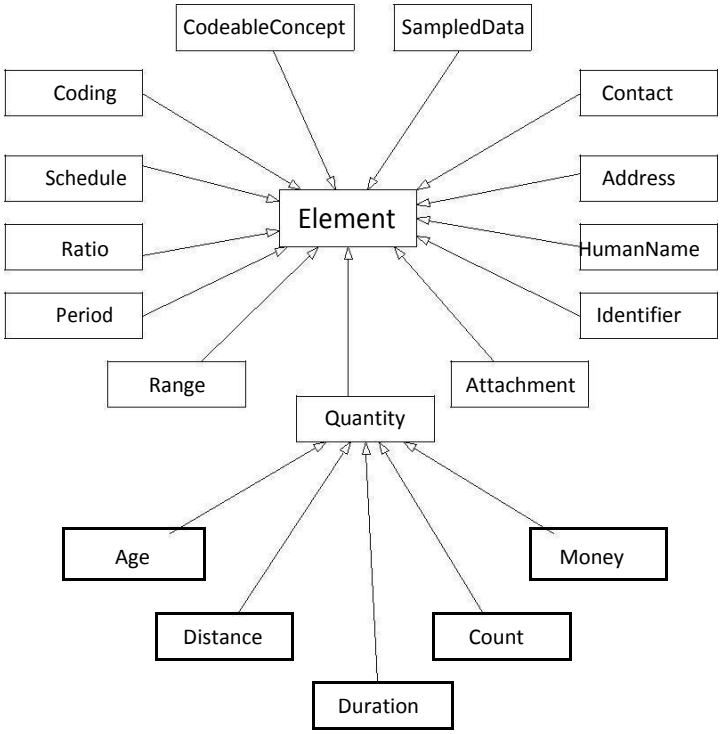

Figure 3.5: FHIR Complex Data Types

3.6 OWL-DL

We have developed our ontology using OWL-DL (Description Logic). OWL-DL is suitable in this scenario because it provides aid in inferring. It is used for reasoning in any application domain. It model concepts, relationships, and individuals and its fundamental concept of modelling is axiom that is a logical statement that relates concepts with their respective roles.

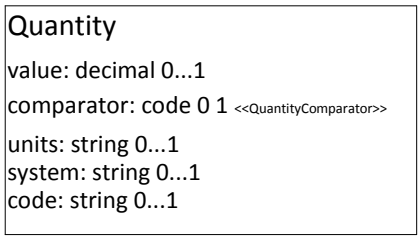

Figure 3.6: Complex Data Type Example

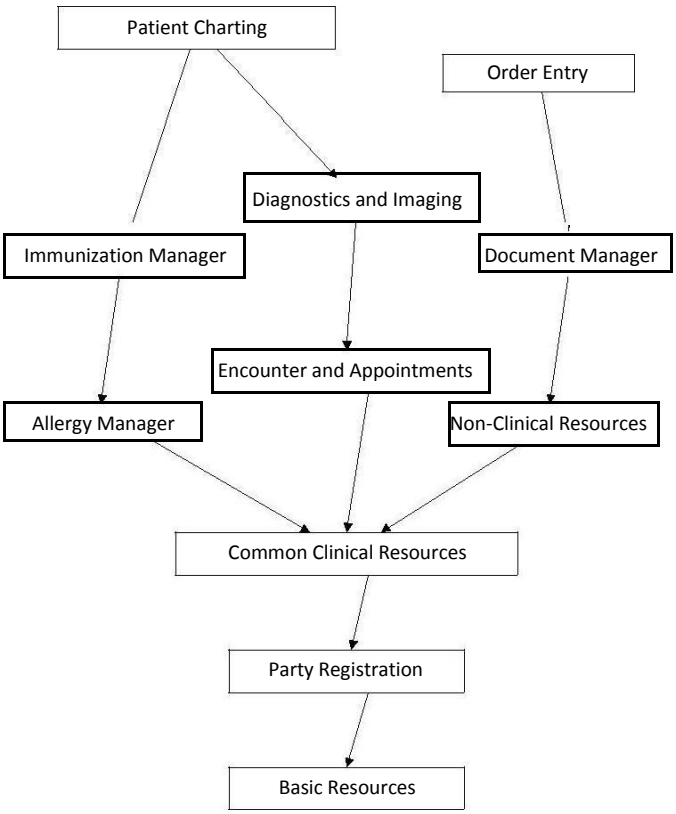

Figure 3.7: Modules of FHIR Ontology

Ontology Module	FHIR Resources
Patient Charting	CarePlan, Condition, Procedure, Referral, MedicationPrescription, MedicationAdministration, MedicationStatement, MedicationDispense
Order Entry	Order, OrderResponse, Supply
Diagnostics and Imaging	DiagnosticReport, DiagnosticOrder, DeviceObservationreport, ImmagingStudy
Immunization Manager	Immunization, Immunization Recommendation
Document Manager	Composition, DocumentReference, Document Manifest, Message Header
Encounter and Appointments	Encounter, Appointment
Allergy Manager	AllergyIntolerance, AdverseReaction, Alert
Non-Clinical Resources	Provenance, Operation Outcome, Query, Pro le, Conformance, Security
Common Clinical Resources	Group, Device, Specimen, Substance, Other, List, Observation, Media, Medication
Party Registration	Patient, RelatedPerson, Practitioner, FamilyHistory
Basic Resources	Location, Organization, ValueSet, ConceptMap

Table 3.2: FHIR Resources in Modules of Ontology

Chapter 4

Validation and Conformance Testing

4.1 Validataion

A method that is used for nding out that a developed system ful lls all the requirements and speci cations that is its intended aim is validation. It is meant to provide explicit guarantees for robustness, certainty and consistency for di erent kinds of input provided for user in an automated system or in an application. Certain rules for data validation can be used by following any methodology for validating data.

Data validation imposes restriction on values in terms of numbers, time, and text etc. so, is preventing invalid data to be recorded and stored for future use and ensures that all data values are accurate and correct in an application. Data validation ensures that data is valid for their contemplated data types and that data will remain valid around applications working

Data validation is necessary to perform for those parts of application that requires input of data from users. Human error is more incredible despite of the fact that such users are intended users.

4.2 Importance of Validation

Data Validation is measure to guarantee accuracy of captured information. If validation is not properly carried out then error rate of 2% to 8% can be there. Data validation ensures that information of a particular pa-tient that is spread across multiple records can distort mortality risk. Any mistake while adding data in any system may result in severe issues and their correctness is time consuming and di cult to track back. IF record of

two patient shaving same name and birthdate are shu ed and one patient is having diabetes and other one is having allergies then outcome may harm to both of the patients. Such common errors should be reduced in order to ensure high quality outcomes and patients safety.

As per AHIMA [?] for validation, one should identify that how data is captured and should ensure it as per some standardized nomenclature or value set. For this purpose, data that is an accurate and exact representation of patients outcome can be considered and used as evidenced based practice and it can be helpful in improving health. Resultant of this all is in improved patient care and safety. Data should be stored for backup purpose in case of emergency situations. This recovery prevents loss of records and further pro-vide aid to healthcare providers in recovering of data and providing proper care to patients in case of natural disaster.

HIPPA rules of security requires healthcare providers to maintain or transfer information electronically so that appropriate technical and physical safe-guards can be provided in order to ensure con dentiality of healthcare in-formation and can protect this information against hazard to its integrity, disclosure or unauthorized use.

The e ciency of any system is dependent on quality of input data and validation of that data is the only measure to nd out that the gathered information is correct or not. In order to perform validation of data, some standard should be followed against which data can be validated. Data is validated by matching it with all the constructs of standard model used. If data is as per that standard then it is valid data and if it is not as per that standard then it is not a valid data as per that standard. After validating data, conclusions are drawn and data is used for further processing and storage.

4.3 Example of Validation of Healthcare Data

According to Importance of NHSN Dialysis Event Data Validation [?] , many discrepencies are resolved by validating data of patients of dialysis. According to this Dialysis Event (DE), data should be accurate, consistent and as per DE de ned constructs and de nitions. As a concequence of inaccurate data, mortality issues and healthcare related malaise can be there.

To cope up with such scenarios DE assess supervision methods so that de - ciencies can be detected and further they provide training education to sta about commonly generated errors and issues so that consistent and accurate data as per DE can be ensured.

We have performed data validation of healthcare data that validates data as per FHIR standard constructs and give response in the form of valid or

invalid data.

4.4 Overview of Validation in Current Sce-nario

After conversion of JSON into RDF it is validated against FHIR schema. If JSON is conformant to FHIR ontology then it will be processed further. Rule based reasoning is performed here that validates the data according to the rules that are de ned in ontology. Validation is performed from di erent perspectives that are described in subsequent sections.

4.5 Data Type Validation

Data type validation veri es the use of correct primitive data values.For example in a slot designated to store only Boolean values, the incoming data should be either true or false. If the data type is integer, its values will be of numeric such as 123. This type of validation is mostly handled at user interface level. Data type validation veri es individual characters that are provided from user are consistent with the expected characters of data types that are de ned in programing language or in schema, according to which data will be validated. Data type validation is performed by us on incoming data that is in RDF. This validator validates whether values in the data are conformant to FHIR data types or not. These data types include both primitive and complex data types. Testing is performed by giving value that does not exist in the range which results in invalid data. Examples of primitive types validation are shown in Table 4.1.

Data Type	Valid Data	Invalid Data
dateTime	2002-09-24+06:00	20020924
decimal	10.0	123
instant	2013-04-03T15:30:10+01:00	2013-04-03T15:30
date	2012-05-24	20142405
oid	urn:oid:2.16.840.1.113883	urn:oid:33.333.4.2.546

Table 4.1: Valid and Invalid Data Examples

4.6 Cardinality Constraints

Cardinality value validation ensures that values will remain between max-imum and minimum bounds of data limit. This type of validation is associ-ated with investigating that whether the data ful lls cardinality constraints or not or its values are according to that cardinalities that are de ned in schema against which data is going to be validated. If a property is marked as functional or has cardinality=1, it should have only one. If more than one values are present then incoming data is violating cardinality constraints of schema. Examples of primitive types validation are shown in Table 4.2.

Cardinality Type	Property Example	Explanation
0-1	name	There can be one value for name
1-1	code	Exactly one value should be there for code
1-*	content	one or more values should be there
0-*	telecom	More than one value can be there

Table 4.2: Cardinality Validation

We have applied reasoner that checks di erent types of validity scenarios according to inference rules that are de ned in FHIR ontology or semantic structure. These inference rules ensures correct data as per FHIR constructs (data types and cardinality constraints). Some of these rules are de ned below:

1. Exactly one RelatedPerson relates to one patient. (patient:Patient(1 to 1))
   9x9y(RelatedPerson(x)) Patient(y)^(x=y)).
2. Every RelatedPerson has a name.
   name:HumanName (0 to 1)
   8x (RelatedPerson(x)^ Name(x))) Name(x).
3. telecom can have zero or more values.
   telecom:Contact (0 to *)
   8x(Practitioner(x))telecom(x).

4.7 Rule Based Reasoning

If we consider FHIR resource of RelatedPerson as shown in Figure cite4.1 then it can be validated by collectively following all the rules within the resource.

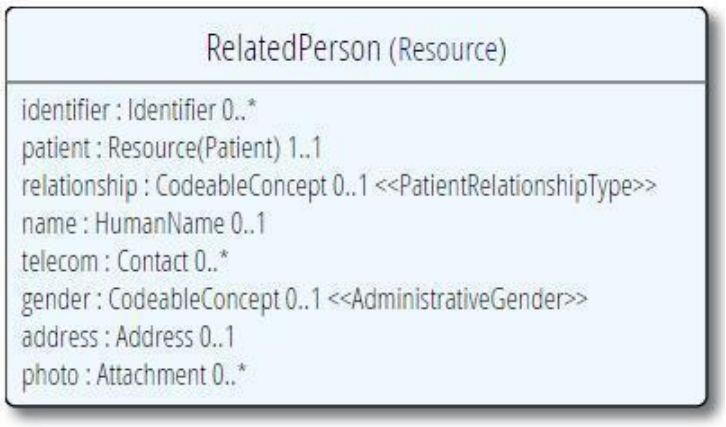

Figure 4.1: FHIR Resource Related Person

In Figure 4.2 all the rules in terms of description logic are de ned. For a resource to be true its all rules should be true. And operator exist within each rule and it is acting as a connector which shows that for a resource to be True, all the rules should be true. If all restrictions and values are ful lled resource is valid else it will be declared as invalid.

$$\forall x(\text{RelatedPerson}(x)) \Rightarrow \text{identifier}(x) \wedge$$
$$\exists x\, \exists y(\text{RelatedPerson}(x) \Rightarrow \text{Patient}(y) \wedge (x=y)) \wedge$$
$$\forall x\,(\text{RelatedPerson}(x) \wedge \text{relationship}(x)) \Rightarrow \text{relationship}(x) \wedge$$
$$\forall x\,(\text{RelatedPerson}(x) \wedge \text{Name}(x)) \Rightarrow \text{Name}(x) \wedge$$
$$\forall x(\text{RelatedPerson}(x)) \Rightarrow \text{telecom}(x) \wedge$$
$$\forall x\,(\text{RelatedPerson}(x) \wedge \text{gender}(x)) \Rightarrow \text{gender}(x) \wedge$$
$$\forall x\,(\text{RelatedPerson}(x) \wedge \text{address}(x)) \Rightarrow \text{address}(x) \wedge$$
$$\forall x(\text{RelatedPerson}(x)) \Rightarrow \text{photo}(x) \Rightarrow \text{Valid}$$

Figure 4.2: Rules for RelatedPerson

4.8 FHIR Structure Validation

FHIR statement validation checks that all the attributes of data are as per FHIR standard. If there exist any property that is not part of FHIR data model then it will be declared as invalid. Conformance testing is performed

by us in order to nd out whether system validates the correct data as per FHIR standard. If it is FHIR compliant i.e. the data ful lls cardinality restrictions criteria or no extra feature apart from FHIR data model is present and data is according to FHIR data types then it is used for further processing otherwise it is discarded and an error message is generated.

4.9 Validation Data Flow Diagram

Data ow diagram of validation is shown in Figure ??. It shows ow from start where JOSN is received and till end that shows input JOSN is valid or not.

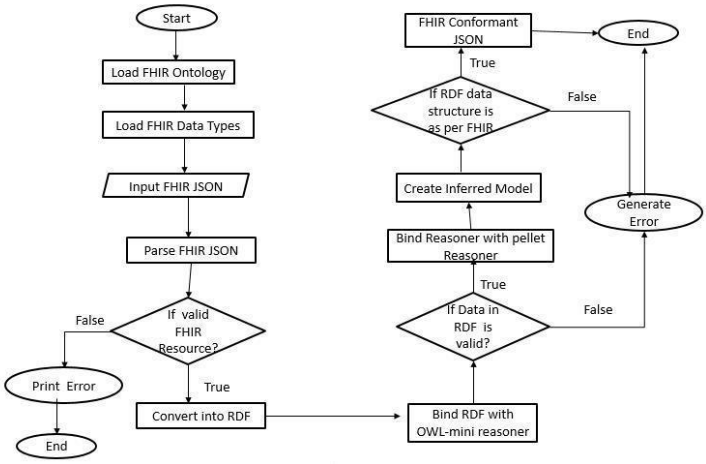

Figure 4.3: Flow Diagram

Chapter 5

Evaluation

Evaluation of the system is performed by checking whether the overall data is FHIR conformant by validating it against FHIRs semantic structure and by nding out that data translated from JSON to JSON-LD is correct.

5.1 Data Set and Test Environment

Validation and conformance is performed on 266 examples of FHIR documents, out of which 221 are correct and are available on FHIR website. These examples vary in complexity from low to medium and high. The complexity is calculated on the basis of number of attribute-value pairs. Examples hav-ing more number of attribute-value pairs are considered to be more complex. Table 5.1 shows complexity criteria for examples of FHIR resources.

Attribute-Value Pairs	Complexity	Number of Examples
Up to 40	Low	97
41 to 59	Medium	75
60 and above	High	49

Table 5.1: Complexity vs Translation Time

Conformance and validation for 45 examples is also performed by introducing errors in valid FHIR documents. These errors include following:

Addition of an extra property (one that does not exist in the FHIR speci cations). This type of data is discarded and reported by valida-tor.

Data with incompatible cardinality is introduced. For example in Practitioners resource (see Figure 3.4), a property "name" has maximum

cardinality of 1 but data with more than one values for "name" is introduced.

Error in primitive data type is introduced. For example in Practitioner resource (see Figure 3.4) a property "birthdate" will accept value of 'dateTime' format but a 'String' value is introduced instead.

The evaluation is performed on a regular desktop with following speci cations: Core i5, 2.53GHz, 4 GB memory, 64 bit Windows OS, Hard Disk 300 GB.

5.2 Evaluation Criteria

In order to evaluate the system, we have proposed following criteria.

All valid FHIR documents should be declared as FHIR conformant.

Only valid FHIR documents should be declared as FHIR conformant.

If there is any inconsistency in FHIR document then it should not be considered as valid and FHIR conformant. The system is hit with several users simultaneously and response time and throughput is calculated.

5.3 Results

FHIR JSON is passed on to validator where OWL reasoner is used to infer over JSON data by comparing it with FHIR data types, FHIR cardinality restrictions and FHIR ontology. For any resource, that is going to be translated so that it can be validated as per FHIR schema, conversion time is calculated by executing it 10 times and then average time is calculated. The conversion from JSON to JSON-LD took time in milli second (ms). Table 5.2 shows translation time.

With the increase in number of attribute-value pairs, time for translation increases. This trend is shown in Figure 5.1 where x-axis shows time in milli seconds and y-axis shows number of attribute-value pairs.

It results in valid and consistent JSON if its as per FHIR constructs. If it is not as per FHIR data model then validator responds with an error message.Validation time is shown in Table 5.3. This validation time shows that with increase in number of attribute value pairs, validation time increases.

E ciency of the system was measured by calculating response time and throughput using J-Meter. Out of 221 examples, 49 FHIR resources of high

Resource Type	Attribute-Value Pairs	Translation Time (ms)
RelatedPerson	43	13
RelatedPerson	60	17
Patient	73	19
93(Patient)	93	21
96 (Practitioner)	96	21
Practitioner	115	23

Table 5.2: Translation Time (JSON to JSON-LD)

FHIR Resource	Complexity Level	Validation Time (sec)
Patient	Medium	126.8
Organization-Pro le	High	135
Location	High	133
Device	Medium	125.1
Organization	Medium	131.7

Table 5.3: Validation Time

complexity were selected and overall response time was calculated. Initially system was hit by 10 users and response time was calculated. Number of users are kept on increasing till 40,000. Then 49 examples of medium complexity are considered and then 49 examples of low complexity and response time is calculated. Figure 5.2 shows response time for three type of resources that includes patient, organization and locations resource.

With increase in number of users, response time increases. Throughput of several requests is calculated that shows how e ciently system carry its functionality when it is accessed by multiple users in a concurrent manner. Initially requests of high complexity are considered and their throughput is calculated. Initially 10 requests are taken into account but the number of requests keep on increasing until it reaches 40,000. Then throughput is calculated for medium and low complexity. Figure 5.3 is showing throughput graph along with increasing number of users.

Throughput reaches to maximum limit for Organization resource till 69 seconds.

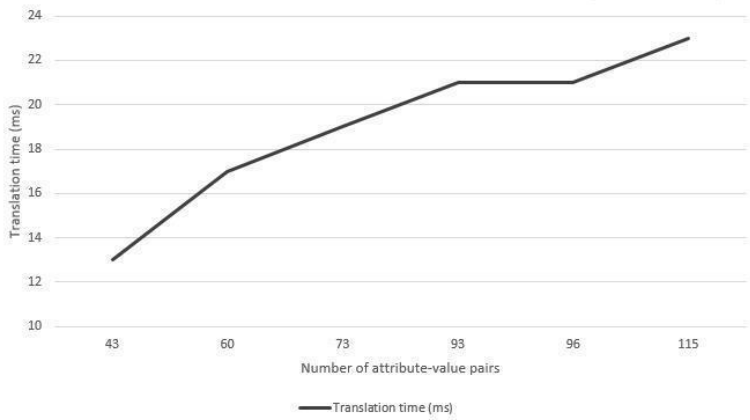

Figure 5.1: Relationship of attribute-value pairs with complexity

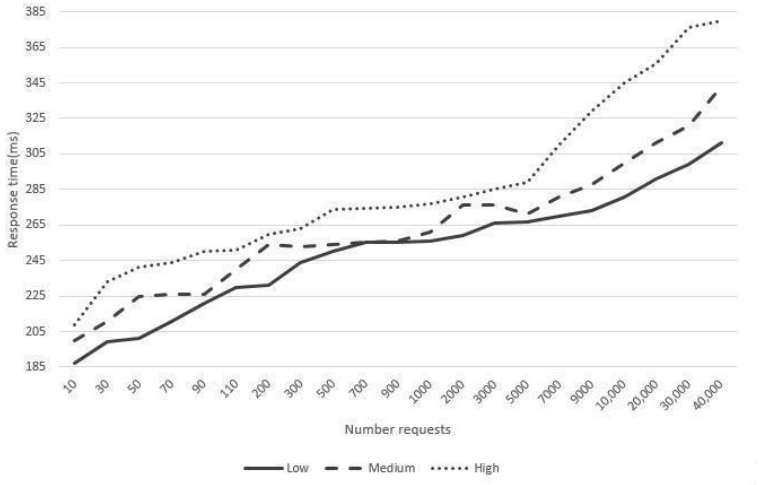

Figure 5.2: Response Time

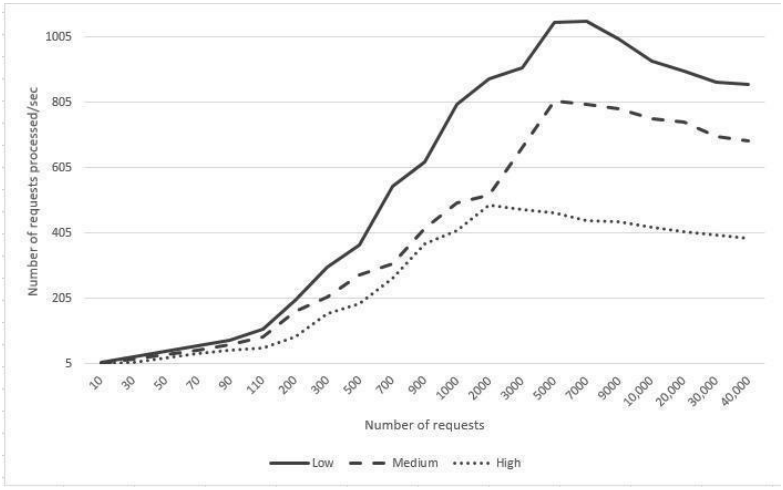

Figure 5.3: Throughput

Chapter 6

Conclusion

Data validation is a crucial task in healthcare domain because of inherent complexity of this domain and data availability in bulk. Data is validated against a standard model that is FHIR, an emerging HL7 healthcare message exchange standard, in this scenario. In this paper we have used JSON that is schema-less serialization format.

We initially translate incoming JSON to JSON-LD, to validate it against some standard language constructs. JSON-LD s then validated against FHIR schema by rule based reasoning, primitive data types and FHIR ontology and nd out whether it is FHIR conformant or not. Validated and conformant data is used for further processing and storage purpose. Correctness of gen-erated JSON-LD by this translator is ensured by converting it back to FHIR JSON. Translation of JSON to JSON-LD remains in sub-seconds and hence this translation is not imposing any signi cant delays in validation process.

E ciency is calculated by calculating throughput and response time.Initially system is hit by 10 users and response time is calculated for 10 number of users. After this we keep on increasing the number of users and response time is calculated for all of these. If we consider organizations resource, then it is tested initially for 10 users then 30 and so on up to for 100 users and throughput is calculated. In the same manner next resource is taken and response time is calculated. Resources of medium complexity are considered. System is accessed by several users and throughput is calculated. Just like for response time, initially 10 users are considered and throughput is calculated for them. After that number of users keep on increasing and throughput is calculated.

If we consider Patients resource then initially for translation of patients resource to JSON-LD, throughput is calculated for 10 users. After that num-ber of users keep on increasing till the number reaches to 100 and throughput is calculated. This validation and conformance tool will conform healthcare

data as per HL7s standard data model that is FHIR in our scenario. As FHIR is an emerging standard and is continuously updating , if any exten-sions will be performed in future then it will be adopted in this validator. This validator will validate and test conformance as per updated standard. If we consider Questionnaire resource then it is updating with time. As per this updated resources, semantic model needs to be modify and its conformance will be tested using validator.

References

[1] Him's role in data capture, validation, and maintenance, 2011. [Online; accessed Aug 2011].

[2] What is compliance testing, 2013.

[3] FHIR standard, 2014. [Online; accessed 30-September-2014].

[4] CDA release 2.

[5] Web ontology language.

[6] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, and Cameron Foulger. Data ow and validation in work ow modelling. In Proceedings of the 15th Australasian database conference-Volume 27, pages 207{214. Aus-tralian Computer Society, Inc., 2004.

[7] Graeme Shanks, Elizabeth Tansley, and Ron Weber. Using ontology to validate conceptual models. Communications of the ACM, 46(10):85{89, 2003.

[8] Decker Hendrik, Teniente Ernest, and Urpi Toni. How to tackle schema validation by view updating. In Advances in Database Tech-nologyEDBT'96, pages 535{549. Springer, 1996.

[9] Terence P Rout. Iso/iec 15504evolution to an international standard. Software Process: Improvement and Practice, 8(1):27{40, 2003.

[10] Lynne Rosenthal, Mark Skall, and Lisa Carnahan. White paper: Con-formance testing and certi cation framework. 2001.

[11] Tanja Toroi, Juha Mykk•anen, and Anne Eerola. Conformance testing of open interfaces in healthcare applications-case context management. In Interoperability of Enterprise Software and Applications, pages 433{444. Springer, 2006.

[12] Messaging testing process, 2005.

[13] Gerrit Jan Tretmans. A formal approach to conformance testing. 1992.

[14] Len Gebase, Robert Snelick, and Mark Skall. Conformance testing and interoperability: A case study in healthcare data exchange. In Software Engineering Research and Practice, pages 143{151, 2008.

[15] The application/json media type for javascript object notation (json), 2006.

[16] JSON lint.

[17] Json lint pro.

[18] Json validator: A comparison of tools and techniques, 2013. [Online; accessed 12-May-2014].

[19] JSON formattor and validator, 2014. [Online; accessed 01-October-2014].

[20] Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using owl. In Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pages 18{22. Ieee, 2004.

[21] Veli Bicer, Gokce B Laleci, Asuman Dogac, and Yildiray Kabak. Artemis message exchange framework: semantic interoperability of exchanged messages in the healthcare domain. ACM Sigmod Record, 34(3):71{76, 2005.

[22] Anni-Yasmin Turhan. Description logic reasoning for semantic web on-tologies. In Proceedings of the International Conference on Web Intelli-gence, Mining and Semantics, page 6. ACM, 2011.

[23] Ian Horrocks. Using an expressive description logic: Fact or ction? KR, 98:636{645, 1998.

[24] Volker Haarslev and Ralf M•uller. Racer system description. In Automated Reasoning, pages 701{705. Springer, 2001.

[25] Thomas Springer and Anni-Yasmin Turhan. Employing description log-ics in ambient intelligence for modeling and reasoning about complex situations. Journal of Ambient Intelligence and Smart Environments, 1(3):235{259, 2009.

[26] Ian Horrocks, Peter F Patel-Schneider, and Frank Van Harmelen. From shiq and rdf to owl: The making of a web ontology language. Web semantics: science, services and agents on the World Wide Web, 1(1):7{ 26, 2003.

[27] RDF 1.1 turtle, 2014. [Online; accessed 25-Feburary-2014].

[28] Module: Rdf::ntriples, 2015. [Online; accessed 10-January-2015].

[29] RDF 1.1 n-quads, 2014. [Online; accessed 25-Feburary-2014].

[30] Notation3 (n3): A readable rdf syntax, 2011. [Online; accessed 28-March-2011].

[31] M Sporny, D Longley, G Kellogg, M Lanthaler, and M Birbeck. Json-ld syntax 1.0, a context-based json serialization for linking data. working draft, w3c, july 2012.

[32] Markus Lanthaler and Christian G•utl. On using json-ld to create evolv-able restful services. In Proceedings of the Third International Workshop on RESTful Design, pages 25{32. ACM, 2012.