Dynamic web forms for linked data management

# Ch# 1. Introduction and Motivation

## 1.1 Dynamic Web Form Generation

**Highlight the motivation, background to the problem etc.**

Now a days, web form is extensively used for data collection, analysis and reporting. It provides users an interface on a web page to add or modify the relational data stored in any databases on server side. Traditionally a web form is developed manually with the combination of different components such as text fields, combo boxes, radio buttons, check boxes etc.

Linked Data is a method of exposing and connecting related data on the web from different sources, using RDF (Resource Description Framework) for data modeling. Linked data adds value to the data, since it becomes more meaningful and useful but at the same time it becomes more tedious and costly to publish and maintain.

The web application designed according to RDF schema needs lots of modifications at multiple layers to incorporate any kind of change in schema. So either data remain unpublished or not properly maintained due to changes required in the structure of data over time. Currently there doesn’t exist a framework which read RDF schema and dynamically generate a web form to add or modify the data in a triple store.

## 1.2 Problem Statement and Research Questions

There is a need of such a framework that not only generates a web form for RDF schema but also handles the user data after form submission by converting the response into triples and stores it into a triple store using SPARQL. It will remove the tedious manual work which makes publishing of structured data quick and economical. More linked data on web will make the web more useful.

Main objective of research is to develop an ontology driven framework for linked data collection and its view generation.

This research will help in speeding up the semantic web application development and also reduce the effort required for its maintenance.

## 1.3 Thesis Contribution and Outline

**Briefly mention your research contributions and summary of each subsequent chapter**

The thesis is divided into two chapters.

This is the first chapter which introduces linked data, highlights the motivation and background to the problem i.e. dynamic web form generation.

The second chapter is background and literature review that discusses web form elements, form layouts, and lists other research efforts in related field of this thesis.

# Ch# 2. Background and Literature Review

## 2.1 Web Form Elements and Events

**Detail of form element, their meta-data, and associate events**

A Formstores the form data and provides methods for reading and updating current values by form elements. It also keeps a list of listeners receiving data update events.

A *Form Element* represents a schema <element> of simple type or an <attribute>. It is a labeled single form field displaying a current value and allowing user to alter it. A Form Elementhas the following properties:

* Data type: a set of possible values assigned to the form field.
* Value: a current value of this field.
* Label: a short text to be displayed aside this field.
* Control: a user interface component that displays and allows changing the current value.
* Possible restrictions: some rules narrowing the set of possible values assigned to this form field.
* Possible read only flag: if enabled user cannot alter the field value.
* Possible required flag: if enabled user is required to enter a value.

The possible restrictions for a *Form Element* are: an enumeration of possible values, the minimum length of value, exact length of value, maximum length of value, minimum inclusive value, maximum inclusive value, minimum exclusive value and maximum exclusive value.

*Events* are Handlers that are placed on Web *Form Element* objects by calling the name of the javascript event handler, such as onclick or onchange, with a set of arguments. The format of the arguments passed to onclick (or any similar method) is a string, a hash reference, or a reference to an array of multiple hash references.

The *Events* associated with a *Form Element* are: onclick, onchange, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onfocus, onblur, onkeypress, onkeydown, onkeyup,and onselect.

## 2.2 Form Layouts

**Discuss different layout choices and display patterns (UI) in web forms**

Layout is another aspect of a web form – do the form fields appear as a table, side by side, below each other or in columns, their alignment etc.

## 2.3 Research Efforts

**Discuss related projects and research papers**

### 2.3.1 A model-driven approach to building modern Semantic Web-Based User Interfaces(Chavarriaga and Macías, 2009)

This approach is based on a Visual Domain-Specific Language. It describes a mechanism intended to design and generate web user interfaces through the use of SemanticWeb and the Web 2.0 technologies. It also describes a mechanism for the transformation process in the automatic generation of web user interfaces.

### 2.3.2 Haystack: A Platform for Authoring End User Semantic Web Applications(Quan and others, )

It Enables developers to create Semantic Web applications more easily.It is Domain Specific Language, designed for working with RDF data. It apply RDF to improve developer experience.

### 2.3.3 A Profile Approach to Using UML Models for Rich Form Generation(Cerny and Song, 2010)

This approach proposed and implemented a system which eliminates the need of a front end developer and also the related task at business layer.They have extended their previous work(a tool that viewed form generation from JPA entity), as UML profile(here UML models are capable of holding data validation information and form generation).This tool provides a configurable translation from entities to view forms.

### 2.3.4 Model-Driven Rich User Interface Generation from Ontologies for Data-Intensive Web Applications (Canadas and others,)

Web UIs from OWL2 domain ontologies was presented. UI for a data-intensive application can be induced from the domain ontology classes, properties and axioms. To obtain an enhanced result it is a presentation model. It captures presentation features related to the UI(UI development is platform specific task, JavaEE and JSF technologies for Web application development). Two frameworks of rich UI components were considered, although the approach can be extended to other frameworks.

### 2.3.5 Exhibit: Lightweight Structured Data Publishing (F. Huynh and others, )

Exhibit enables web site authors to create dynamic exhibits of their collections without resorting to complex database and server-side technologies. The collections can be searched and browsed using faceted browsing. Assorted views are provided including tiles, maps, etc.

### 2.3.6 Semi-Automatic Generation of GUIs for RDF Browsing (Pazienza and others, 2010)

### 2.3.7 Defining The Semantics Of Rule–Based Web Applications Through Model–Driven Development (Canadas and others, 2011)

### 2.3.8 User Interfaces Supporting Casual Data-Centric Interactions on the Web ( David F. Huynh, 2007)

### 2.3.9 Conceptual Modeling and Code Generation for Rich Internet Applications (Bozzon and Comai, 2006)

### 2.3.10 Ontology Driven Dynamic Web Interface Generation (Luna and Schwabe, 2009)

### 2.3.11 Applying Ontology and Reasoning to the Semantic Web for Recommendation (Junayeed and Abdullah , 2009/2010)

## 2.4 Anything else that comes to your mind