Journal of Theoretical and Applied Information Technology
30th November 2017. Vol.95. No 22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SAGG: A NOVEL LINKED DATA VISUALIZATION APPROACH

[1]NOEMI SCARPATO AND [1]GIANFRANCO ALESSIO

[1]San Raffaele Roma Open University, Dept. of Human sciences and Promotion of the quality of life, Rome, Italy

E-mail: [1]noemi.scarpato@unisanraffaele.gov.it [2] gianfranco.alessio@unisanraffaele.gov.it

ABSTRACT

In this paper, we describe the Semi-automatic GUI Generator (SAGG), a knowledge-based visualization system able to create GUI in a semi-automatically way. Since the linked has been introduced their expressiveness made possible to provide users with a lot of useful information, visualize this information is a crucial issue in the realization of the semantic web principles. The objective of our approach is the induction of tailored GUI able to show the considered linked data in the better way. The key idea behind our approach is the exploitation of existing web pages to deduct visualization patterns for linked data.

Our intent is to provide common users with a semi-automatic GUI generator, this system is able to visualize linked data without the necessity for the above-mentioned users to know semantic technologies for data visualization.

Following, we present an introduction to linked data visualization systems, our solution to linked data visualization issue: the SAGG system, its architecture, the implemented algorithms to realize it and its user interaction mechanisms.

Keywords: *Linked Data Visualization, Semantic Web, Information Visualization, Knowledge Visualization, OWL, SPARQL.*

1. INTRODUCTION

Visualization of linked data is a key issue of Semantic Web. While linked data are machine-understandable, in its raw representation they are not comfortably understandable to humans. As the traditional web provides users with appealing GUIs, the Semantic Web should be able to provide common users, that are interested to consume linked data, with user-friendly GUIs.

The existing knowledge-based visualization approaches present some troubles and open challenges. One of the main challenges in this kind of approaches is the rapid prototyping of GUI capable to satisfy the common user information needs.

The SAGG system aims to provide common users with the ability to interact with the web of data without knowing the representation languages (e.g. RDF, SKOS) and the visualization languages (e.g. FRESNEL).

Our intent is to provide common users with a semi-automatic GUI generator, this system will be able to visualize linked data, without the necessity for the above-mentioned users, to know semantic technologies for data visualization.

In this paper, we provide a semi-automatic GUI generator able to visualize linked data by asking users only graphic- sample web pages and the set of data to show.

To realize this we perform the analysis of the sample pages and the search over data and finally we produce a configuration file able to described the created GUI.

The final output of our system is a GUI similar to the sample page in the graphic structure and filled with users data.

Nowadays linked data has been introduced in different areas such as legal [1][2][3] medical [4][5][6] , cultural heritage [7], [8] , Humanitarian Assistance and Disaster Relief [9][10][11] and Autonomous and Connected Vehicle [12].

In all of these areas, users aim to visualize data in easily and rapidly way, we present SAGG as solution of this problem

In this paper, firstly we analyze state of art of linked data visualization approaches. Then we provide a critical analysis of these approaches and finally, we propose a novel solution to solved open challenges: SAGG a novel approach to linked data visualization through the generation of GUIs in a semi-automatic way [13], [14].

We also provide the implementation of this approach: the SAGG system. To the best of our knowledge, our approach is new in knowledge-based visualization research field. SAGG is a knowledge-based visualization system that exploits some examples submitted from users to automate the visualization process.

In particular we will describe in details the components of SAGG system which we have created in order to implement the approach described in [13]:

- SAGG VIPS algorithm [14]: a new approach to Web Page visual segmentation (VIPS) algorithms.
- SAGG Semantic Search Algorithm [14]: a novel Semantic Search algorithm.
- SAGG Structuring & Formatting Fresnel-based approach: a Configuration File composer module able to create Groups, Lenses and Formats according to the Fresnel standard language.

2. LINKED DATA VISUALIZATION APPROACHES: STATE-OF-ART

The Semantic Web [15], also defined as “Web of data”, implies that data in the Web should not only be machine-readable but also machine-understandable.

Starting from this assumption, it is necessary to introduce the concept of metadata, that can be defined as "data that describe data": this kind of information allows communication between computers and humans. The metadata can make the meaning of data on the Web explicit so as to provide computers with enough information to handle such data.

The key idea of this vision is to make the Web machine-understandable, different from traditional Web where data are only machine-readable.

To define metadata, it is needed to refer to a shared model of the world, then the definition of ontologies has been introduced to provide this model in the Semantic Web. The ontologies establish a common conceptual description and a joint terminology between members of communities of interest (human or autonomous software agents). While Ontology term is borrowed from Philosophy, several definition of the concept of ontology have been provided in the past.

In this paper we take the following definition for ontology concept, this definition is usually adopted in computer science: “an ontology is an explicit, formal specification of a conceptualization needed to organize knowledge in specific domains and applications.” [16].

Making data machine-understandable opens new scenarios on the Web, where humans and machine can communicate without misunderstandings.

This scenario presents several challenges in many fields of research.

One of the most active areas of the Semantic Web is the development of a variety of tools (e.g. browser, search engine, semantic annotation tools etc...) for the authoring, extraction, visualization, and inference of metadata.

For what concerning the Semantic Web, visualizing the structure of a formally defined information resource domain is important not only for designers and developers, but also for the business and individual users of the first generation of the Web.

Visual representation of metadata in their raw form, is difficult to understand for common users, then in the last years a new field of research has been investigated to implement the information visualization on the Semantic Web.

In the Semantic Web, visualization is a crucial aspect to provide users with GUIs that show metadata in a visually comprehensible way [17].

For above-mentioned reasons the objective of this paper is to achieve this challenge.

To do that in this paper, we analyze the state-of-art of visualization research field and we propose a novel solution to carry out a system able to create GUIs in a semi-automatic way: SAGG system.

Recently many attempts to realize a linked data visualization system has been provided in literature: [18],[19],[20],[21],[22],[23],[24].

All of this approach try to provide a smart way to visualize linked data, following we analyze the existing different approaches to realize a linked data visualization system.

There are two main research areas useful to improve linked data visualization:

- Information Visualization
- Knowledge Visualization

Information Visualization can be defined as:

“ ...a process of transforming information into a visual form enabling the viewer to observe, browse, make sense and understand the information. It typically employs computers to process the information and computer screens to view it using methods of interactive graphics, imaging and visual design. It relies on the visual system to perceive and process the information.” [25]

Information visualization (InfoVis from now on) has been recently devoted to exploit metadata to provide proper representation for data in the Semantic Web. Once the metadata are available, it is relatively straightforward to use them in order to visualize them.

The knowledge-based visualization (KVis from now on) research field instead, is involved in implementation of systems that exploit domain knowledge to define proper patterns of visualization. In this sense, RDF visualization can surely benefit from the results and findings in the InfoVis field.

The idea behind both of these research areas is that management and comprehension of information and knowledge may be enhanced using a proper pattern of visualization.

However, most existing knowledge-based visualization applications work only on specific domains/tasks, reflecting the difficulties in generalizing and reapplying visualization approaches to new problems or domains.

In the Semantic Web, the task of visualization of metadata consists of visualizing the linked data in a visually comprehensible form. The objective of this task is to make linked data human-understandable. To accomplish the task of linked data visualization it is due to specify which information contained in an RDF graph and how this information should be presented.

There are two major approaches to Semantic Web data visualization: adopting and applying existing InfoVis solutions or developing completely new techniques specifically tailored for the knowledge representation paradigms of the Semantic Web .

The information that is originated from RDF data can be interpreted as a subgraph. This subgraph, can be extracted by using queries: selecting and composing these queries (for which a standard is available from early 2008, in the form of the SPARQL query language) requires a combination of domain/technical expertise to be applied.

Another key point of Semantic Web Data Visualizations is the definition of the graphic elements that are associated to domain information. These can be selected to expose a representation on their own.

The Semantic extensions to RDF, such as RDFS/OWL or other standards such as SKOS may play a pivotal role in representing this information, through " ad hoc" ontologies of visualization, these ontologies refer to graphical aspects such as templates decorating UI widgets, geometrical aspects (size, width, depth …), or other aspects strictly related to the source data (the order in which certain collections of resource are displayed, or the way they are clustered …).

It is very important to point out that the pairs of type:

<RDF-resource, representation> (1)

are relevant "per se", because they can be exposed, collected and reused by generic linked data browsers and viewers according to the same paradigm which is proposed by the Semantic Web for resource shareability and reusability.

2.1. Visualization Process

The visualization of data can be defined as a process that aims to transforming information into a visual form.

This can be achieved on the one hand through the selection of proper sub-graph in an RDF graph, that should match user needs and on the other hand through the definition of how this sub-graph could be presented.

In [26] the authors state that this visualization process can be divided in three steps:

- Selection: aims to identify sub-graph.
- Structuring: aims to define proper visualization structures.
- Formatting : aims to define formats related to above mentioned structures to styling visualization pattern (e.g. colors, fonts, background).

Each of this steps can be completed by the application of several techniques.

The selection step can be implemented by the following approaches:

- Manual filtering: to select the subgraph, users need to select the resources manually.
- Incremental navigation: to select subgraph, users navigate through linked data.
- Querying the metadata repository: to select subgraph, users need to create a query.

As well as on the Web, users need submit query to retrieve information, in the Semantic Web users should be able to submit a query to select relevant data.

The semantic search is the field of research that defines methods to search linked data. The semantic search approaches can be improves through the integration of techniques of Web page visual analysis.

Structuring & Formatting steps, instead, deals with the definition of visualization structures and theirs styles. In [27], the authors discuss the design of the Annotation Profile Model, which consists of a data-capturing (the Graph Pattern Model) and a presentation part (the Form Template model).

Furthermore in [28] the authors defines an RDF display vocabulary: Fresnel.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Fresnel allow to define RDF visualization information that can be linked to an RDF resources to define representation information of data.

Fresnel defines two basic concepts:

- Lenses: define which properties of one or more RDF resources to display and their order of presentation.
- Formats: determine how to render the resources, their properties and values.

By using Fresnel it is possible to realize the Structuring & Formatting steps of the above mentioned visual process.

The key idea below these approaches is the definition of a evident separation between the data and their representation.

These approaches, also, enable developers to define a chain of operations that allow the realization of knowledge-based visualization systems, that are able to create a user-friendly GUI to browse linked data.

2.2. Knowledge-Based Visualization Systems

The key idea of Knowledge-based visualization systems is to exploit linked data to visualize them. Define proper user interfaces is one of the most challenges in the Semantic Web, in particular, one of the main difficulties for the researchers, is to make linked open data available in an understandably way in order to support their exploitation. Linked data have by their nature an high coupling value, then we need to show complex and very large graphs of resources.

Furthermore, retrieval and organizations of linked data in the Semantic Web interfaces are two additional challenges in the visualization of Semantic Web data.

Recently several approaches have been developed to create representation forms for visualizing linked data. In [29] the authors describe many different approaches to visualize linked data, they show how the research field of linked data visualization is a key issue of semantic web realization.

In [30] authors underline how the growing amount of (big) linked data make visualization a critical point on the exploitation of linked data.

As mentioned above the visualization process can be divided into three steps: selection, structuring, and formatting. While the selection deals with the identification of relevant sub-graph in a big linked-data graph, the structuring and formatting steps deal with the organization and styling of these data.

In literature has been defined many different linked-data visualization approaches able to implement the structuring and formatting steps into the Knowledge-Based Visualization systems.

These approaches can be divided in:

- Graph-based approach: provides a graph view representation [31] [32] [18].
- Faceted browsing approach: provides forms view filled with formatted RDF representation of resources [33].
- Domain-specific approach: provides a graphical representation tailored on a specific domain .

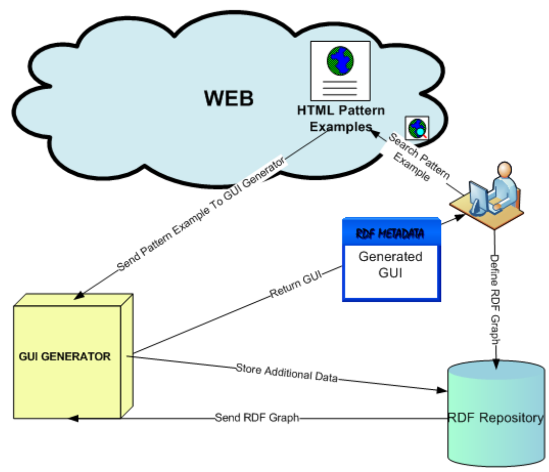

Figure 1: SAGG Scenario

- Widgets-based approach: provides a rich graphical representation of resources by exploiting graphical metaphor.

As example: tables, lists and trees realized by using XHTML, XUL and SVG scripts and the related stylesheets [34] [35].

Each of the above-mentioned approaches presents some issue that should be addressed and solved.

- The graph-based approach presents some problems in displaying very large graphs.
- The faceted browsing approach lacks the possibility to show clearly the relationships between data.
- The domain-specific approach is able to show only linked-data related to a specific domain.
- The widgets-based approach is a good solution but the existing implementations require the definition of the specific configuration files. Most of the attempts to implement this kind of tools foresee the creation of this configuration files from users.

Following we propose an attempt to solve this problem through the realization of a Knowledge-based visualization system: the SAGG system.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SAGG is able to generate a widgets-based GUI, performing the automatic creation of the configuration files.

3. SAGG

As mentioned above, the linked-data consumers should be able to visualize data and interact with them without know implementation details of the knowledge-based browsers. In particular, users should be able to visualize linked-data in a rich GUI able to represent the linked-data complex information and the relations between them.

The goal of SAGG is to go a step forward in this direction, by introducing a further level of abstraction which is provided by interdisciplinary work of domain experts. The SAGG provides the above-mentioned users with rapidly deployed mock-ups of required interfaces, possibly already working at a basic level of detail (thus needed some fine-tuning, which are requested to the users).

Our approach is to design a mechanism and a chain of processes (in order to identify a realizable architecture) that automatically generates the queries for extracting the desired subgraph, starting from available examples, and also defines the right representation for the selected resources.

The approach we are proposing has high potentials and aims to solve one of the critical issues of linked-data consumption, the visualization.

The key idea behind SAGG ideation is the exploitation of existing user interfaces to infer one or the best GUI for considered linked-data.

Given a set of linked-data, there is a high probability that the Web contains many different pages that employ graphics widgets for showing the same knowledge items.

For example, for representing statistical data it is possible to find a web page that contains similar data showed in a table, our idea is to use this table to show our data.

SAGG system, by adopting Fresnel[28] vocabulary, is able to collect the visualization information of the web page in a list of pairs:

$$\langle \text{Lenses, Formats} \rangle \quad (2)$$

and to use them to configure the user's interface to generate the linked-data GUI.

We decided to exploit existing web pages as visualization example for two main reasons:

- inducing the regularities for extracting the best graphical structure for the UI.
- inducing the rules to automatically fill graphical structures with linked data.

The second point is extremely importance: the representation examples on the Web do not need to have been produced by the same dataset own by the user. Instead, the same example just needs to contain data representing information originated from the same domain (or, at least, that share a sensible overlap with it). Then, this information can be searched over the real dataset owned by the user, hoping that a good percentage of the data will be recognized for it and that the system would thus be able to induct the queries needed to extract analogous data from the dataset. The key points of our approach are:

- Configuration Files are lively created when the user starts a learning process over data observed from a browsed example.
- The patterns of representation are determined by the users;
- The query identifying the interesting subgraphs is learnt by an automatic process;
- The information about the pattern of representation are stored and placed at user's disposal for future sessions of navigation (or to be exported for other interested users/developers).

The SAGG scenario is composed of several elements (components and relevant objects) whom interaction is shown in **Error! Reference source not found.**.

SAGG components are:

- A Semantic Repository containing linked data;
- A GUI Generator that performs the automatic generation of user interface.
- SAGG relevant objects are:
- Some HTML pages the content of which is being selected by the users;
- The automatically generated UI.

The linked data Repository contains many triples describing the domain of interest (the model) as well as the data provided by the user and/or retrieved during the processing of inputs. Also, data are modeled for representation purposes and describe the created pairs:

$$\langle \text{triple , representation} \rangle \quad (3)$$

through Fresnel Lens and Formats.

In the initial phase, if the linked data repository contains already enough data, it can immediately be used as a seed to learn new UIs from available examples retrieved from the Web, with no supervision. If the repository is empty and many similar examples are available from Web pages, it

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

can be automatically populated by semantically annotating even very few pages (as reported in [36]).

To realize our approach a small number of annotated examples are needed. Exploiting these pages it is possible to apply the wrapper induction techniques.

In summary, SAGG receives as an input: HTML pages selected by the user (or parts of them) and the available linked data. Then SAGG analyzes the input and automatically generates a GUI mock-up. Finally, SAGG, exploiting the generated SPARQL queries, is able to extract correct values from the linked data repository and to fill the generated GUI with them.

4. THE DATA MODEL

The data model behind SAGG algorithm foresees a set of elements which are identified and analyzed in standard documents.

In the SAGG's approach, the DOM standard structure is adopted to make assumptions regarding the content of the page. The input of SAGG is called *"PatternExample"*. A *"PatternExample"* is an HTML page, or part of this, that users identify as relevant for their domain. A *"PatternExample"* can be segmented into a set of recognizable Graphic Objects called GOs:

$$PatternExample = \{GO_0, GO_1, \ldots GO_n\} \quad (4)$$

A GO is a part of a "*PatternExample*" that exposes relevant and mostly self-contained, information. SAGG assumes that the content of a GO is an independent information unit i.e. information that can be analyzed separately from the rest of the page with respect to the user data. As from the adopted DOM formalism, the GO can be structured through a tree with a root element (e.g. TABLE, LIST ...) identifying its nature as a list of trees. A GO is composed by a list of atomic information units called GE.

$$GO=\{GE0, GE1, \ldots GEn\} \quad (5)$$

In SAGG's approach, it is assumed that a pair of GEs contained in the same GO may be bound through a relationship. The following grammar has been defined to identify the possible relations between two GEs.

$$(GE, +, x) \quad (6)$$

The two types of relations of previous grammar are:

- \+ being sibling
- x dependency

The sibling relation is a generic relation that is established between pairs of GE which are part of the same GO and are considered as peers with respect to the content analysis process. Note that the choice of the name "sibling" is appropriate with respect to the semantic analysis and assessing of relationships among GEs.

On a syntactical perspective, this should not bring confusion with the fact that "sibling" elements may be nested inside each other or belong to different branches of the DOM tree and thus not be siblings in the hierarchical organization of elements.

The second relation defined in SAGG grammar, called dependency, can be established among pairs of elements where one of them acts as a pivot for the other (the relation has thus a direction).

Usually, a pivot establishes a dependency with several other elements (e.g. a column header in a table is a pivot for the content of the cells in its same column).

GEx	GEy	Relation	Rank
0	0	0.2	0.2
0.3	0	0.2	0.5
0	0.3	0.2	0.5
0.3	0.3	0.2	0.8
0	0	0.4	0.4
0.3	0	0.4	0.7
0	0.3	0.4	0.7
0.3	0.3	0.4	1

Figure 2: Triples ranking values

This type of relation implies a stronger binding with respect to the sibling relation (which relies on simple keyword-based research in the graphs) in this case the algorithm can apply more constraints on the research. For each GO a list of triples can be defined:

<GEx , relation, GEy> (7)

These triples are ranked according to the following method.

First of all, we assign a weight to the sibling relation (0.2) and to the dependency relation (0.4).

Then to determinate the weight for each GE involved in the relation, we have implemented an algorithm to check if some GEs can be mapped with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

some to nodes of the dataset. If this condition is verified, a weight equal to 0.3 is assigned to it.

The final rank of a triple is the sum of the weight of its relation and weights of each of the two GEs. In the *Figure 2: Triples ranking values*, we show all possible combinations of weights for a triple.

5. USER INTERACTION

As mentioned above the users of SAGG system are common users who want to navigate in the Semantic Web. Typically they know their own domain and are able to identify the data they need. Usually, users aim to access only a part of these data in an appealing graphical form which gives them the possibility to consume the information.
The combination of the data identified and their representation provides users with the possibility to automatically create web pages for the semantic web.

In spite of its partial automatization, the SAGG's approach designates a centric role for users, as they provide the semantically annotated examples and, where appropriate, they validate the data retrieved from Web pages. The interaction between the users and the system consists first of all in the definition of the set of linked data that the users want to represent. Then the users browse the Web and search pages that contain data similar to the selected set of linked data.

Subsequently, the user selects in the retrieved pages a graphical pattern and asks the system to extract a UI widget and to populate it (by using an ad hoc query) with data from the available linked data dataset.

So, SAGG starts the chain of processes and proposes a widget. For this widget, SAGG provides the possibility to modifying and then saving the Fresnel format.

The second step is to edit the query (e.g. adds more restrictions or simply changes some of its characteristics) and save this query into the corresponding Fresnel Lens.

This sequence of steps can be reiterated several times to refine the UI and change the associated query accordingly.

6. ARCHITECTURE

The architecture of the SAGG system is based on the visualization process because it provides all of three steps of the above mentioned process: selection, structuring and formatting.

The SAGG architecture is organized as a flow that involves a chain of different modules, each of them providing an output that is exploited by one or more of the following modules.

The SAGG architecture is composed of four modules, in addition, there is a data manager to access the user's semantic repository. The first three modules of the architecture are involved in achieving each of the three steps of the visualization process. The fourth module realizes the automatic composition of the final GUI.

The SAGG architecture adopts the reusability and shareability of the information that are worthwhile in all Semantic Web applications.

The first module of the SAGG architecture is devoted to the pattern generation and carries out, along with the Query Generator, the selection step of the visualization process.

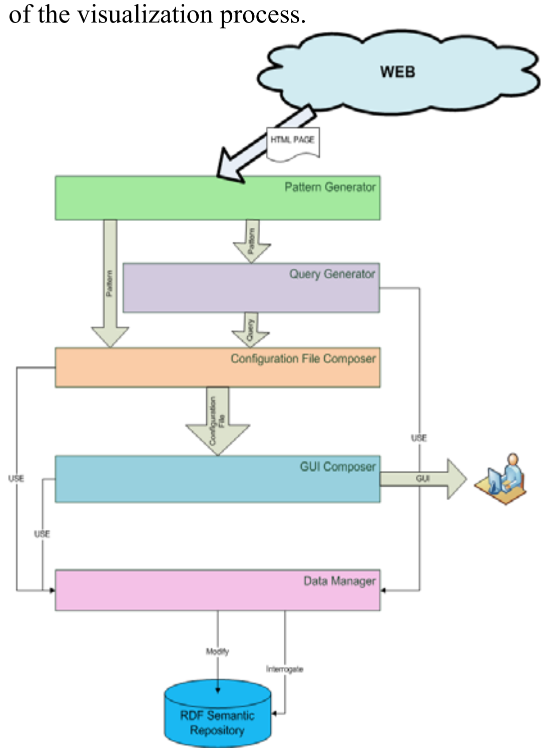

Figure 3:SAGG Architecture

This module implements a Web page segmentation algorithm to identify graphics objects (GO) (see Figure 4) and their graphic elements(GE). Furthermore, the same module generates a list of triples as:

<GEi, relation, GEy> (8)

these triples contain the relations between all pairs of GEs that are part of the same GO. A key aspect of this processing is that the structure of the GO is exploited to extract useful information, in particular,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the position of GEs in a recognized GO is analyzed to establish relationships between GEs and to assess the nature of these relationships.

Algorithm GO identification Algorithm

for $i = 0$ $node_i$ i **do**
Previous rules to identify the GOs are applied;
for $y = 0$ GO_y y **do**
if in a GO there are elements that are GO themselves **then**
the GO is added separately to the list of GOs ;
Further generic GO is created for the remaining part of the root element.
end if
end for
end for

Figure 4: GO Identification Algorithm

The second module: Query Generator, implements a semantic search algorithm that takes as input a set of GOs with the associated list of triples generated by the Pattern Generator module.

The Query Generator module analyses the matching graph patterns to induct the query which best approximates the data contained in the input. This process is performed for each of the GOs. The output of this module consists of a list of GOs and related queries able to populate them.

In the first two modules, we implement two algorithms to identify patterns of representation and queries able to fill them, implementing the selection step of the visualization process.

The following modules instead are in charge of generating the user interface.

In the SAGG system, we adopt the Fresnel vocabulary[28] to create the user interface. The Configuration File Composer module uses both the patterns and queries provided by the previous modules to create a Configuration File modeled according to the Fresnel vocabulary. The Pattern Generator identifies the patterns used to generate the Formats. The Query Generator provides the queries to generate Lenses.

In the SAGG's approach, the Configuration File composer implements both the structuring and the formatting steps of the Visualization Process through the definition of both Lenses and Formats.

The GUI Composer uses the Configuration File to automatically generate a GUI capable of showing the user's dataset organized according to the required pattern of visualization.

Moreover, the produced Configuration Files are stored into a knowledge base and can be shared with other users.

This module has been designed to be capable to parse a Fresnel file and to create a form using the information available. The generated forms can be exported in different Formats as HTML, JSP, XUL, form java etc.

The last module of the SAGG's architecture is the Data Manager; it provides an interface to access the

Algorithm SSA

for $i = 0$ GO_i i **do**
we analyze the list of triples <GEi , relation, GEy >
for $i = 0$ $triple_i$ i **do**
The keywords are extracted from textual content of both GEs by using the Chaos parser
A list of triples like: <$keyword_{i,k}$, relation, $keyword_{y,k}$>related on <GE_i , relation, GE_y>is created.
If possible the keywords related to the GE are mapped into some nodes in the knowledge base
if keyword can be mapped in a node **then**
This keyword is marked as a seed (this means that this keyword has a weight equal to 0,3)
end if
We assign a weight value to the <$keyword_{i,k}$ relation, $keyword_{y,k}$ >triple, this value is calculated as the sum of the weight values of both keywords in the triple plus the weight value assigned to the relation (this value depends on the kind of relation see paragraph 5.3 for further details)
end for
The resulting list is arranged according to the weights
end for

Figure 5: SSA Algorithm

user dataset and it is realized through the OWLArt APIs.

These APIs offer an abstraction layer over different RDF triple store technologies so that applications can use them and easily switch among different storage technologies without adapting source code to their APIs. Considering previous motivations, in the SAGG system, the OWLArt API has been integrated into the Data Manager module.

7. SAGG WEB PAGE VISUAL ANALYSIS ALGORITHM

In this section we describe the rule-based VIPS algorithm. We design this algorithm to realize the Pattern Generator module.

The VIPS algorithm take as an input various format of web documents, for example, an HTML Document.

The first task of the algorithm is the identification of the above-mentioned GOs and of their graphic elements GEs in the sample web page. The algorithm assumes that in case of an entire web page there are more GOs in this page, in the case of a part of the page, instead, it assumes that input contains only a GO.

The contributions provided by SAGG's VIPS algorithm are the introduction of a further level of segmentation in VIPS approaches and the introduction of an approach that states that the segmentation it isn't only useful to identify different topics within a Web page but can provide further information about the nature of each of these topics in the page.

In the next section, we propose the SSA algorithm that is a semantic search algorithm that exploits the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SAGG's VIPS algorithm output to improve its search process.

8. SAGG SEMANTIC SEARCH ALGORITHM

As mentioned above the retrieval of linked data is a challenging task. In SAGG's approach, we ask users to provide an example page in order to extract both a representation pattern and a query able to populate the generated GUIs.

The SSA algorithm is the semantic search algorithm we have implemented to accomplish the challenge of extract these queries.

In Figure 5 is shown the flow that brings to the creation of these queries.

The SSA algorithm, also , provides the analysis of the generic relations for mapping them into specific relationships into the knowledge base.

We define a threshold to determine which are the triples to be analyzed. In the current implementation of the Query Generator module, this threshold is fixed to 0,4 (this means that only the triples which contain at least one seed are analyzed) . The threshold can be tuned to arrange the performances of the system (e.g. if the threshold is smaller than 0,4 more triples are analyzed, this can lead to the retrieval of further correspondences with the knowledge base but also can worsen the performances generating noisy information).

In Figure 6 is shown the steps performed from SSA Algorithm to analyze the recognized triples.

At the best of our knowledge, the SSA algorithm differs from other semantic search approaches [37] because they require an input query or at least a list of keywords. SSA instead exploits the output of SAGG's VIPS algorithm to automatically generate the query.

This approach makes the composition of candidates lists of triples easier in comparison with other approaches.

Furthermore, the SSA algorithm implements an algorithm of query expansion guided by the definition of relationships identified by the analysis provided by the SAGG's VIPS Algorithm. The SSA algorithm tries to map not only the keywords but the entire triple and this allows to identify both resources and relations.

Algorithm SSA 2

```
For each triple it is checked if the rank value is greater than threshold.
If it is:
if rank == 1 then
  ( this means that the relation are dependency and the keywords are both a seed )
  An "InstanceOf "relation between seeds is searched into the knowledge base.
  if the triple is into the knowledge base then
    A query like: "SELECT ?x WHERE ?x a keyword_{y,k} "
    is added to the queries list associated to GE_i
    A value keyword_{y,k} is added to the value list associated to GE_y
    The same weight of the analyzed triple (1 in this case) is associated to each of
    this value. ( The values are also added because they could be used to fill the GE
    in case of not valid query are identified. )
  else if This triple isn't into the Knowledge base then
    It is searched into the knowledge base a "subClassOf "relation between seeds.
    if This triple is into the knowledge base then
      A query like: "SELECT ?x WHERE ?x rdfs:subClassOf keyword_{y,k} "is
      added to the queries list associated to GE_i
      A value keyword_{y,k} is added to the value list associated to GE_y
      The value 1 is associated to each of them.
    else
      The knowledge base is explored until a triple is found.
      if This triple is into the knowledge base then
        The related query and value, are added to proper list.
      end if
    end if
  end if
end if
```

Figure 6: SSA Algorithm 2

9. SYSTEM EVALUATION

As mentioned above SAGG approach is new in the knowledge visualization research field, then it is difficult to compare SAGG with other visualization approaches that require the submission of configuration file to visualize RDF information. To test our system we have submitted to SAGG system the *foaf ontology* and the personal web page of Tim Berners Lee, then we have compared the configuration file created by SAGG with the foaf example provided by Fresnel web site and with the configuration files: fresnel-foaf-example.n3 and foaf-example-fresnel-simple-core-and-lens-group.n3 provided in the Simile repository.

With respect to the downloaded configuration files, the SAGG configuration file adopts a "fresnel:sparqlSelector" in the identified lenses. Moreover the SAGG system has been able to recognize a simple lens for foaf:Person class.

Comparing this lens with the lenses related to foaf:Person defined in others configuration files, we can notice that there are not "fresnel:showProperties" values.

In conclusion, we can say that, while it defines easier lenses, SAGG can be used to automate the creation of the configuration file.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

10. OPEN RESEARCH ISSUES AND FUTURE WORKS

This paper deals with a complex problem that concern the linked data visualization. The proposed solution, that consists of the SAGG system seems to be useful and actionable. However, a step forward has to be done introducing, the combining of several Configuration Files to generate more complex GUIs, possibly specifying interrelationships (i.e. semantic constraints) between them.

While this could simply be seen as a further refinement process resulting in more complex Fresnel configurations, we would stress the importance for the user of being able to specify compositional patterns for reusable atomic Fresnel units, in a sort of Semantic Mash-up.

This would open up the way for reusable and shareable libraries of active UIs (i.e., bringing with them the information on how to populate them starting from the available data), which should be easily searched (according to different perspectives: what they show and how they do it etc), accessed, imported (into heterogeneous Semantic UI developing environment) and composed according to user/developer needs, in the spirit of the Semantic Web vision.

Another important future improvement is to explore the possibility to expand our model to include the formalization of relations between GOs with the aim to extend the VIPS algorithm (SAGG VIPS) and consequently provide a more informative input to the semantic search algorithm (SSA).

The exploration of this kind of relations is based on the idea, as mentioned above, that not only the significance of the position of the GEs in a GO, but also for the GOs their position in the example pages can be useful to understand their meaning, mostly in the cases where a GO is nested in another GO.

Moreover, the SAGG's Vips algorithm can be improved by the definition of further strategies that will strengthen the identification of the dependence relations between GEs in different kind of GOs (not only table but also paragraph etc..).

11. CONCLUSION

In this paper, we presented (in its latest release) SAGG: an innovative approach to generating visualization forms able to show linked data in an appealing manner.

To the best of our knowledge, this approach represents an innovative contribution in the knowledge-based visualization research field.

We described the data model behind our system (SAGG) and the architecture which implements it.

Each module of the architecture deals with one different aspect of the visualization process.

Our system provides the identification of the proper pattern of representation and the generation of the structured queries starting from web pages examples input.

The SAGG also implements the generation of Configuration File according to Fresnel W3C specification and finally it carries out the generation of customized GUIs to show linked data.

Moreover, in the SAGG approach, we have implemented two innovative algorithms (the SAGG's VIPS algorithm and the SSA algorithm).

Differently from other approaches in literature, our method can interact with different domains as the SAGG asks users to indicate their own knowledge base and it generates the GUI exploiting the example Web page provided from them.

The SAGG approach can be implemented and integrated in very different scenarios as an extension for: Semantic Enhanced Web Browsers, RDF Browsers, Ontology Editors and Annotation Tools.

Beside to the visualization feature, our approach can be exploited to facilitate the knowledge augmentation. Indeed as mentioned in the rest of the paper SAGG identifies not only the candidate queries but also the values that could be used to fill the GEs. If these are not in the knowledge base they can be added to the generated GUI in any case and can be proposed to the users and if the values are validated, they are stored in the knowledge base.

At the other hand, at the present stage of development, SAGG is able to create only simple GUI and the identified queries can select only a (relevant) part of the data.

SAGG system assumes that users know at least the domain of data and are able to select proper examples, if not the system could not be able to generate proper GUI.

In conclusion, we can affirm that the SAGG system has reached the goal of providing a RDF user-friendly browser with a very appealing graphical interface.

Furthermore, the SAGG system is compliant with the reusability and sharing principle of the Semantic Web vision because it provides a standard Configuration File that can be reused and shared among users.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] M. Bianchi, M. Draoli, G. Gambosi, M. T. Pazienza, N. Scarpato, and A. Stellato, "ICT tools for the discovery of semantic relations in legal documents," in *CEUR Workshop Proceedings*, 2009, vol. 582.

[2] M. T. Pazienza, N. Scarpato, and A. Stellato, "STIA*: Experience of semantic annotation in Jurisprudence domain," in *Frontiers in Artificial Intelligence and Applications*, 2009, vol. 205, pp. 156–161.

[3] G. Boella, L. Di Caro, L. Humphreys, L. Robaldo, P. Rossi, and L. van der Torre, "Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law," *Artif. Intell. Law*, vol. 24, no. 3, pp. 245–283, Sep. 2016.

[4] P. Ferroni *et al.*, "Risk Assessment for Venous Thromboembolism in Chemotherapy-Treated Ambulatory Cancer Patients," *Med. Decis. Mak.*, vol. 37, no. 2, pp. 234–242, 2016.

[5] P. Ferroni, F. M. Zanzotto, N. Scarpato, S. Riondino, F. Guadagni, and M. Roselli, "Validation of a machine learning approach for venous thromboembolism risk prediction in oncology," *Dis. Markers*, vol. 2017, pp. 1–7, 2017.

[6] F. Guadagni *et al.*, "RISK: A random optimization interactive system based on kernel learning for predicting breast cancer disease progression," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10208 LNCS, pp. 189–196.

[7] A. R. D. Accardi and S. Chiarenza, "Musei digitali dell ' architettura immaginata : un approccio integrato per la definizione di percorsi di conoscenza del patrimonio culturale Digital museums of the imagined architecture : an integrated approach," *DISEGNARECON*, vol. 9, 2016.

[8] M. Pennacchiotti and F. M. Zanzotto, "Natural Language Processing Across Time: An Empirical Investigation on Italian," Springer, Berlin, Heidelberg, 2008, pp. 371–382.

[9] F. Fallucchi, M. Tarquini, and E. W. De Luca, "Supporting Humanitarian Logistics with Intelligent Applications for Disaster Management," *INTELLI 2016*, p. 64, 2016.

[10] F. Fallucchi, M. Tarquini, and E. W. De Luca, "Knowledge Management for the Support of Logistics During Humanitarian Assistance and Disaster Relief (HADR)," P. Díaz, N. Bellamine Ben Saoud, J. Dugdale, and C. Hanachi, Eds. Cham: Springer International Publishing, 2016, pp. 226–233.

[11] D. Zhang, L. Zhou, and J. F. Nunamaker Jr, "A Knowledge Management Framework for the Support of Decision Making in Humanitarian Assistance/Disaster Relief," *Knowl. Inf. Syst.*, vol. 4, no. 3, pp. 370–385, Jul. 2002.

[12] A. Pieroni, N. Scarpato, and M. Brilli, "Industry 4 . 0 Revolution in Autonomous and Connected Vehicle A non-conventional approach to manage Big Data," *J. Theor. Appl. Inf.*, vol. 96, no. 1, 2018.

[13] M. T. Pazienza, N. Scarpato, and A. Stellato, "Semi-automatic generation of GUIs for RDF browsing," in *Proceedings of the International Conference on Information Visualisation*, 2010, pp. 267–272.

[14] M. T. Pazienza, N. Scarpato, A. Stellato, and a R. T. Group, "Application of a Semantic Search Algorithm to Semi-Automatic GUI Generation University Telematica San Raffaele," *Int. Conf. Lang. Resour. Eval. (LREC 2012)*, pp. 3631–3638, 2012.

[15] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities," *Sci. Am.*, vol. 284, no. 5, pp. 34–43, 2001.

[16] T. R. Gruber, "A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications," *Knowl. Creat. Diffus. Util.*, vol. 5, no. April, pp. 199–220, 1993.

[17] V. Geroimenko and C. Chen, *Visualizing the Semantic Web*. Springer, 2006.

[18] F. Ghorbel, F. Hamdi, N. Ellouze, E. Métais, and F. Gargouri, "Visualizing Large-scale Linked Data with Memo Graph," *Procedia Comput. Sci.*, vol. 112, pp. 854–863, 2017.

[19] T. Ermilov, D. Moussallem, R. Usbeck, and A.-C. N. Ngomo, "GENESIS – A Generic RDF Data Access Interface," *Proc. Int. Conf. Web Intell. - WI '17*, pp. 125–131, 2017.

[20] H. S. Nguyen, H. P. Pham, T. H. Duong, T. P. T. Nguyen, and H. M. T. Le, "Personalized facets for faceted search using wikipedia disambiguation and social network," in *Advances in Intelligent Systems and Computing*, 2016, vol. 453, pp. 229–241.

[21] E. Kharlamov, L. Giacomelli, E. Sherkhonov, B. C. Grau, E. V Kostylev, and I. Horrocks, "SemFacet: Making Hard Faceted Search Easier," *Proc. 26th ACM Int. Conf. Inf. Knowl. Manag.*, pp. 2475–2478, 2017.

[22] M. Arenas, B. Cuenca Grau, E. Kharlamov, S. Marciuska, and D. Zheleznyakov, "Faceted Search over Ontology-Enhanced RDF Data," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, 2014, pp. 939–948.

[23] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "Visualizing ontologies with VOWL," in *Semantic Web*, 2016, vol. 7, no. 4, pp. 399–419.

[24] M. Weise, S. Lohmann, and F. Haag, "LD-VOWL: Extracting and visualizing schema information for linked data," *CEUR Workshop Proc.*, vol. 1704, pp. 120–127, 2016.

[25] "Infovis: Home." [Online]. Available: https://www.infovis.org/. [Accessed: 20-Oct-2017].

[26] L. Rutledge, J. Van Ossenbruggen, L. Hardman, and P. O. Box, "Making RDF Presentable : Integrated Global and Local Semantic Web Browsing," *Simile*, pp. 199–206, 2005.

[27] M. Palmér and M. Nilsson, "Annotation profiles : Configuring forms to edit RDF," *Proc. 2007 Int. Conf. Dublin Core Metadata Appl. Appl. profiles theory Pract.*, pp. 10–21, 2007.

[28] E. Pietriga, C. Bizer, D. Karger, and R. Lee, "Fresnel: A Browser-Independent Presentation Vocabulary for RDF," in *ISWC 2006 - The Semantic Web*, vol. 4273, no. 4273, Springer, Berlin, Heidelberg, 2006, pp. 158–171.

[29] A. S. Dadzie and M. Rowe, "Approaches to visualising Linked Data: A survey," *Semantic Web*, vol. 2, no. 2. IOS Press, pp. 89–124, 2011.

[30] N. Bikakis and N. T. U. Athens, "Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art - Semantic Scholar," 2016.

[31] M. T. Pazienza, N. Scarpato, A. Stellato, and A. Turbati, "Semantic Turkey: A browser-integrated environment for knowledge acquisition and management," *Semant. Web*, vol. 3, no. 3, 2012.

[32] M. T. Pazienza, N. Scarpato, A. Stellato, and A. Turbati, "Din din! The (Semantic) Turkey is served!," in *CEUR Workshop Proceedings*, 2008, vol. 426.

[33] C. Becker and C. Bizer, "DBpedia mobile-a location-aware semantic web client," *Proc. Semant. Web Chall.*, pp. 1–8, 2008.

[34] T. Berners-lee *et al.*, "Tabulator: Exploring and Analyzing linked data on the Semantic Web," *Swui*, vol. 2006, no. i, p. 16, 2006.

[35] A. Soylu, M. Giese, E. Jimenez-Ruiz, G. Vega-Gorgojo, and I. Horrocks, "Experiencing OptiqueVQS: a multi-paradigm and ontology-based visual query system for end users," *Univers. Access Inf. Soc.*, vol. 15, no. 1, pp. 129–152, Mar. 2016.

[36] S. Soderland, C. Cardie, and R. Mooney, "Learning Information Extraction Rules for Semi-structured and Free Text."

[37] G. Madhu, A. Govardhan, and T. K. V. Rajinikanth, "Intelligent Semantic Web Search Engines: A Brief Survey," *Int. J. Web Semant. Technol.*, vol. 2, no. 1, pp. 34–42, 2011.