# StYLiD: Social Information Sharing with Free Creation of Structured Linked Data

Aman Shakya
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, Japan 101-8430
shakya_aman@nii.ac.jp

Hideaki Takeda
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku,
Tokyo, Japan 101-8430
takeda@nii.ac.jp

Vilas Wuwongse
Asian Institute of Technology
Klong Luang, Pathumthani,
Thailand 12120
vw@cs.ait.ac.th

## ABSTRACT

Information sharing can be effective with structured data. The Semantic Web is mainly aimed at structuring information by creating widely accepted ontologies. However, users have different preferences and evolving requirements. It is not practical to attempt perfect schema definitions with strict constraints. Creating structured formats should be a collaborative and evolutionary process. Social software motivates wide participation by providing easy interface. We propose a system called StYLiD for sharing a wide variety of structured information. Users freely define their own structured concepts. The system consolidates different versions defined by different users. The attributes of the different concept versions are aligned semi-automatically into a single unified view. Popular concepts gradually emerge from the concept cloud and stabilize. Concept definitions are flexible. An attribute value can take a literal or a resource URI and the suggestive range does not constrain the contributors. StYLiD generates unique dereferenceable URIs so that data items can form a linked data web. Structured data is embedded in machine readable form using RDFa. Search and browsing features are provided to utilize the structured data and consolidated concepts.

## Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

## General Terms

Design

## Keywords

Structured data, information sharing, social semantic web, concept consolidation, collaboration, linked data, RDFa

## 1. INTRODUCTION

Information sharing on the Web has become a basic need in communities. We want to share a wide variety of information. It would be desirable to have some system which can facilitate the modeling and sharing of such heterogeneous pieces of information. Structuring data helps handling different types of data systematically. There are many advantages of having structured data[10, 2].

- It becomes easy to define the semantics of data and make it machine understandable so that processing can be automated.
- Information sharing becomes more effective when data is structured following common conventions.
- Search and browsing becomes more effective with structured data.
- Structured information can be easily mixed. It becomes easy to integrate information from various sources.
- We can have interoperability between different systems by forming standard formats. Even multiple structure definitions for similar data may be mapped to each other.

Thus, structured data becomes open and shared for all rather than being closed in proprietary systems. With the growing significance of structured data, the Web is rapidly moving towards a *Structured Web* which can be a transitional step towards the Semantic Web and can be fully realized with current technologies[10, 2].

Efforts for the Semantic Web have been mainly being directed towards creating standard formats in the form of ontologies. However, currently there are not many ontologies to cover the wide variety of information we may want to share[19, 22]. Even if ontologies do exist, it may be difficult to search an appropriate one for our purpose. Further, understanding and using such ontologies is not an easy task for non-technical users. Like the Web, the Semantic Web should let anybody to share information about anything. There is a long tail of information domains for which different individuals have information to share[8]. There are separate well-established solutions for dealing with the head of few popular information types. However, for the long tail, availability of software is rare and developing individual solutions every time is infeasible. Moreover, a uniform solution would be desirable for interoperability and integration.

Creating new ontologies and information systems is not easy. Data modeling is a difficult task. It should be flexible to accommodate requirements and exceptions that surface in the future. Users may need different data and varying levels of details depending upon the purpose. Moreover, people


*SWKM'2008: Workshop on Social Web and Knowledge Management @ WWW 2008,* April 22, 2008, Beijing, China.
.

have different views and should be allowed to maintain their preferences. It is not practical to impose a single standard or strict constraints.

Thus, creating ontologies or common formats should be a widely collaborative process[19]. A small team of ontology engineers cannot take into account the wide range of data and requirements of all users. However, to have large scale collaboration and to motivate general users, information sharing systems should be easy to use and understand. Ontologies can be a by-product of the usual information sharing activities in the community.

On the other hand, social software has proven to be successful in drawing huge user participation and contribution. Tagging is successful because it is very simple and anyone can contribute easily. Systems like tagging and social bookmarking do not impose any hard constraints for sharing data. However, these systems do not provide much semantic structure to information. Though some social software systems do provide structured data, they are closed systems with less interoperability and integration with other systems.

Recently, the combination of social software with Semantic Web technology towards a Social Semantic Web has been gaining significant attention[6, 1]. However, we need more tolerant mechanisms and ways to round up inconsistencies and inaccuracies that result from the informal approach of the social web[17]. We will still have a non-standard web with multiple formats. In the web, heterogeneous or overlapping conceptualizations are bound to appear[1]. However, the problem of mapping representations is not difficult, as long as the information is structured[10]. The initial step for the Semantic Web is to generate lots of data and we should facilitate easy contribution and provide incentives. Rationalization of data can be done later[8].

We propose a system called StYLiD (an acronym for Structure Your own Linked Data) which gives users the freedom to define the structure of their own data. It is easier to define one's own quick data model than to search for suitable ontology or schema and understand it. We propose to let the users input information freely without imposing any constraints, just like tagging. Computations can be done later to consolidate similar concepts, deal with inconsistencies and align multiple definitions. Concepts can gradually converge to stability by usage in the same way as folksonomies. The quality and stability of data is maintained when many eyeballs are watching and people can vote contents. This has been demonstrated well by social sites like Wikipedia[1] and Digg[2]. Furthermore, StYLiD is an open system that can link to external data and allows others to link in for building a linked data web[3].

We discuss some use case scenarios in Section 2. We describe the StYLiD platform in detail in Section 3. Section 4 gives some details about implementation. We discuss some related works in Section 5. Finally, we conclude in Section 6 and state some ongoing and future work.

## 2. USE CASE SCENARIO

Fig. 1 is a use case diagram which briefly shows what the user would be able to do with the system. Some details are given below.

[1]http://www.wikipedia.org/
[2]http://digg.com/

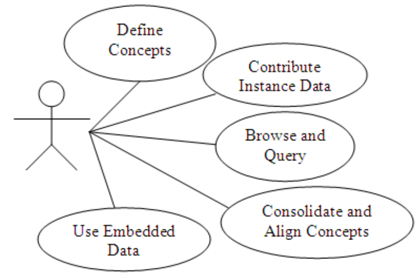


**Figure 1: Use case diagram**

### 2.1 Defining Own Structured Concepts

Suppose a user wants to share some structured information. However, he cannot find a suitable schema or any system for handling such data. He may freely register an account on StYLiD and define his own structured concept on the fly. He simply enters the concept name and a list of attributes. If a similar concept already exists in the system, he may choose to use the concept directly and enter instances or modify the existing concept to create his own version. He may modify his own concepts later and add more attributes whenever needed.

### 2.2 Flexible and Relaxed Data Entry

The user may easily start sharing data using his own structure definition. Any other registered user can also use his concept and contribute instance data. While entering data the system helps the user by suggesting range of values for the attributes. The user may easily pick instances from this range. However, any suitable value may be entered even though it is not in the suggested range. The user may easily type in literal values for attributes. If the user knows a resource URI for the value, it may be entered to link to that resource. The corresponding resource may also be entered later and the original entry edited to specify the URI link.

### 2.3 Browsing and Querying Structured Data

All the user defined concepts are visualized as a concept cloud where popular concepts are seen bigger. The user can browse different types of data with the concept cloud. When the user hovers over any concept, the attributes and description of the concept are shown so that the concept and its structure can be instantly understood (see Fig.5). This is useful to see how well defined the concept is and whether it is appropriate for him. He may wish to view only the concepts defined by him or any particular user as a concept cloud. He also maintains a personal concept collection of useful concepts, also viewed as a concept cloud. Instances of a concept can be viewed in a record view or a table view. The user may switch between these views. The user can navigate through linked data entries. The data entries may also link to external resources. The user may search data instances using a simple web-based interface by specifying the concept name and a set of attribute name, value pairs as criteria. Advanced users may directly query the system using a SPARQL query interface.

### 2.4 Consolidating and Aligning Concepts

Different versions of a concept defined by different users are consolidated by the system and shown as a single virtual concept. The different versions are grouped together in the concept cloud. The individual concepts in a group can be identified by visible labels for the creator name and version number. By clicking on a consolidated concept, the user would be able to see all the instances of all versions. He may want to see all the instances of a concept regardless of the creator or version. He may also want to see all the instances of a concept defined by a particular user regardless of the version. He may want to see only the instances of a particular version defined by a particular user. The consolidated concept cloud offers the desired granularity.

When the concept is a single distinct concept, the table view is straightforward, each attribute displayed as a column. However, when it is a consolidated concept, the corresponding attributes of the individual constituent concepts have to be aligned first. The system automatically suggests alignments in a form-based interface. The user may update this and add mappings not suggested by the system. Then all the data can be viewed in a unified uniform table view. The user may also rename the attributes of the integrated view and hide unwanted columns if needed to get a customized view.

## 2.5 Utilizing Machine Readable Embedded Data

The system embeds machine understandable RDFa[3] data in the HTML posts. An RDFa aware browser would be able detect such contents and offer suitable operations for the user. Many RDFa tools and plug-ins are becoming available[4] and we may expect more powerful tools to be available in the future. The use of RDFa has also been demonstrated by recent works on semantic clipboard[15] which would allow users to copy structured data into useful desktop applications. The user may copy and paste the embedded structured data elsewhere on the Web or distribute using social media.

# 3. THE STYLID PLATFORM

The StYLiD platform realizes the use cases described above. It enables the users to define their own concepts on the fly and share structured data. The main contributions of the system are as follows. Details are provided in the subsections to follow.

- **Sharing structured data with user-defined concepts.** Users may define their own concepts with attributes, freely and easily, and share structured data using them. Different users are allowed to have different versions of the same concept. Users can share, reuse and refine such concept definitions.

- **Consolidation of user-defined concepts.** Multiple versions of concepts defined by different users are consolidated and corresponding attributes are aligned to produce a unified consolidated view. Popular concepts emerge out from the cloud of concepts.

- **Flexible definitions and relaxed data entry.** Users are allowed to input information freely, according to their needs and preferences, instead of attempting perfect schema definitions and imposing strict constraints.

- **Open system for creating linked data.** The system allows open access to its data using open standards. It can link both internal and external data to support a linked data web.

[3]http://www.w3.org/TR/xhtml-rdfa-primer/
[4]http://rdfa.info/2007/02/12/call-for-proposals-rdfa-utils-services/

StYLiD is still a prototype and development is going on. A demo installation is available online[5]. Currently we are populating some sample data in the academic domain with different versions of concepts like faculty, courses, seminars, etc. Heterogeneity is common in such data because academic institutes have different systems and formats. Most of the data is being populated with the help of scrapers created using the free online service, Dapper[6]. We intend to continue using StYLiD in this domain with real users. However, the system can be installed and used for any other domain or general purpose.

## 3.1 Sharing Structured Data with User-Defined Concepts

The main interface of StYLiD is shown in Fig.2. The users of the system may freely define their own concepts by specifying the concept name, some description (optional) and a set of attributes. Each attribute is defined by the attribute name, description (optional) and a set of concepts as the suggested value range (optional) as shown in Fig.3. Any user may enter instance data for the concepts using the interface shown in Fig.4. An attribute of a concept can take a single value or multiple-values. Each of the values may be a literal or a resource (identified by its URI). If the value is a resource URI, a human readable label may be entered along with the URI.

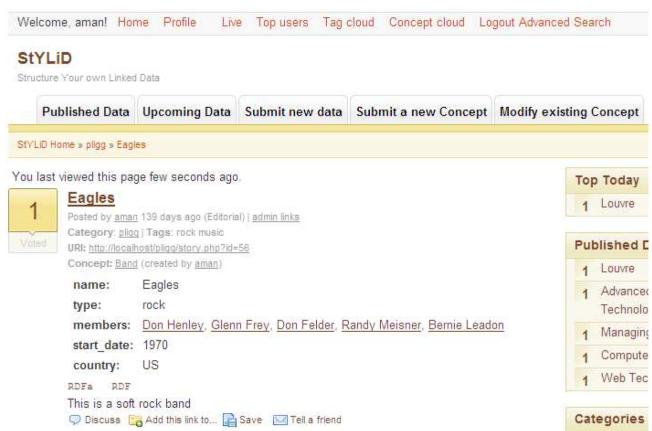


**Figure 2: StYLiD interface**

The system allows different users to define their own concepts having the same name. Moreover, users do not need to define concepts from scratch. The user can modify an existing concept to make own version. However, users are not allowed to tamper with others' concept definitions. The system makes a copy of the concept and allows the user to make modifications on it. It keeps record of the source from which the modified concept was derived using the *dc:source* property. Users can update their own concept definitions keeping the existing instances consistent. Attributes can be

[5]http://dutar.ex.nii.ac.jp/stylid/
[6]http://www.dapper.net/

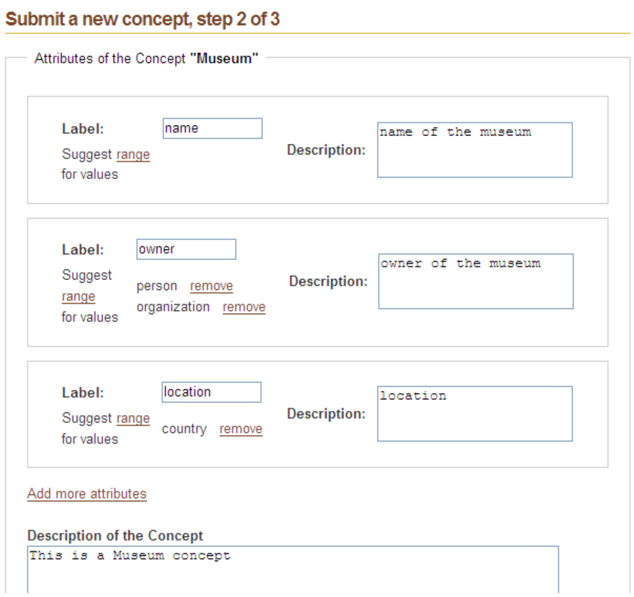


**Figure 3: Interface to create a new concept**

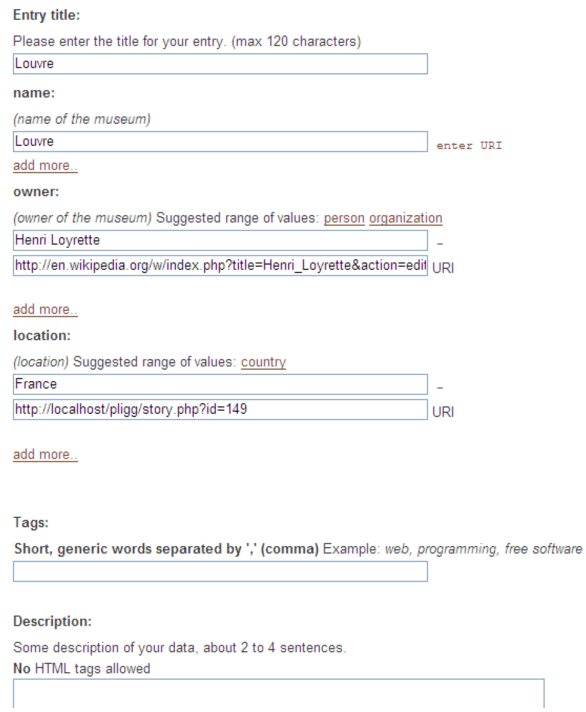


**Figure 4: Interface to enter instance data**

added. However, if we need to rename or delete attributes of the concept a new version of the concept should be defined to keep the existing data intact. Thus, the same user can also have different versions of his/her concept with the same name.

**Structured Data Formats.** The system embeds machine readable structured data in HTML using RDFa format. It also outputs the data in RDF format separately. Thus, the system produces formal machine understandable contents though the user interface is quite simple and informal like a tagging system.

**A Personal Structured Data Space.** The system offers every user a personal structured data space. It provides a *Concept Collection* for each user, as seen in Fig.5. Concepts created or adapted by the user are automatically added to this collection. Besides these, users can also add any other useful concepts to their collection. The users need not be overwhelmed by the huge cloud of concepts defined by the large number of users. Moreover, the concept collection is also helpful to mark the concepts that the user has been using out of numerous concepts and different versions. The concepts actually created by the user are also shown in a separate tab.

## 3.2 Flexible Definitions and Relaxed Entry

Creating perfect concept definitions with strict constraints is not easy and practical. It is difficult to think of all attributes and all possible value ranges at the time of concept definition. It may also be difficult to say whether an attribute value would be a literal or a resource and whether the attribute would have a single value or multiple values. While defining a concept A, if an attribute must take a resource of type concept B, we must first ensure that concept B has already been defined. If concept B has an attribute which takes resource values of type concept C, then concept C must be defined first, and so on.

Similarly, at the time of instance data entry, it may be difficult for the user to enter perfect data as mandated by a schema. All attribute values may not be known. Proper resource URIs for attribute values may not exist or the user may not be able to find it at the time. Moreover, exceptions may always exist no matter how well the schema has been designed and unpredicted new data instances may appear.

The system tries to avoid these difficulties in data modeling and data entry by allowing flexible and relaxed definitions. The concept definition may be incrementally updated later and new attributes may be added. New versions of the concept may be defined by different users or even the same user. The range of values defined for attributes, as seen in Fig.3 and 4, is only suggestive and do not impose strict constraints. Rather the system assists the user to fill data using the suggested range. The suggestive range may be updated later by including more concepts or narrowing down to refine the range. The system accepts literal values though resource values may be desirable for an attribute. Instances may be updated later to change a literal into a resource value by adding the URI. The users may input single or multiple values for any attribute as appropriate. With such relaxed data entry interface, of course, we may get some imperfect, incomplete or heterogeneous data. However, users generally enter appropriate or sensible data for their purpose. This has been evidenced by systems like tagging and wiki which accumulate large volume of good data in spite of having completely relaxed interface.

## 3.3 Consolidation of User Defined Concepts

Concepts defined by different users with the same name are grouped together by the system. This forms a single virtual concept which consolidates all the grouped concepts. This consolidated concept can be used to retrieve all the instances though different users have different definitions for the concept name.

If $C_1, C_2, ...C_n$ are the concepts defined by users *1, 2,... n* with concept name "C", the consolidated concept is given by $C = C_1 \cup C_2 \cup ... \cup C_n$

Further, different versions of the same concept defined by a single user are also grouped together. Thus, we can obtain all the instances of a concept defined by a user irrespective of the version.

If $C_{i1}, C_{i2}, ...C_{im}$ are the versions *1, 2,... m* of concept "C" defined by the user $i$, then the consolidated concept for the user is given by $C_i = C_{i1} \cup C_{i2} \cup ... \cup C_{im}$

### 3.3.1 Consolidated Concept Cloud

All the concepts contributed by different users are visualized together as a *Concept Cloud*, similar to a tag cloud. Better concept definitions will satisfy more users and will have more instances. Popularity of concepts is visually highlighted by increasing size. Popular concepts will receive more attention and motivate more use in turn. Thus, stable definitions will gradually emerge out from the vast cloud of concepts as more instance data are contributed. Clicking on any concept shows all instances of the concept.

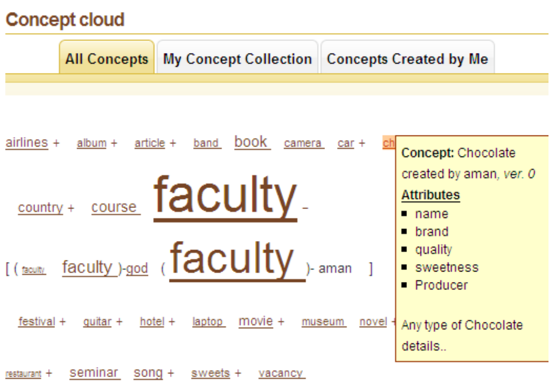


**Figure 5: Concept cloud**

A consolidated concept formed by grouping different versions can be expanded into a *sub-cloud*. The sub-cloud shows all the versions of the concept defined by different users, labeled with the user name. Further, in the sub-cloud, if multiple versions are defined by the same user, they are sub-grouped together. In the Fig.5, the "Faculty" concept has been expanded to show two versions by the user "god" and one version by "aman". The sizes of all the different versions in the sub-cloud add up to form the size of the consolidated concept. Clicking on the consolidated concept shows all instances of all the versions of the concept. Similarly, we can also see all instances of the multiple versions of a concept defined by a single user by clicking on the user name.

### 3.3.2 Semi-Automatic Concept Alignment and Unification

Different concepts in a consolidated group are aligned to produce a uniform and integrated view. When the instances of a consolidated group of concepts are viewed as a table, as shown in Fig.7, the system automatically suggests alignments between the attributes of the concepts, as shown in Fig.6. Matching attributes are automatically selected in the form-based interface. Currently, the mapping is simply based on the Levenshtein edit distance[7] between the attribute labels. So slight variations on spelling and morphology are easily handled.

It is not possible to make the alignment fully automatic and accurate. Moreover, alignments may vary for different users and for different purposes. So it is desirable to have the user in loop though the system greatly simplifies the work by providing automatic suggestions. The user can complete the process by adding matching attributes that the system could not detect or modify the suggested mappings. Thus, we propose to use both machine intelligence and human intelligence for the alignment process.

**A Unified View.** Each set of aligned attributes can be considered as a single consolidated attribute for the consolidated concept. The system automatically fills a name for each consolidated attribute, as shown in Fig.6, though the user may rename it as desired. The user may even remove attributes from the unified view, if not required. Thus, the user can create a unified view of the consolidated concepts, customized according to his need, and view heterogeneous instance data in a uniform table.

Align Concepts

| Album created by god (ver. 0) | Album created by god (ver. 1) | Album created by aman (ver. 0) | Combined attribute name |
|---|---|---|---|
| name | name | name | name |
| release_year | release_year | released | release_year |
| num_of_tracks | num_of_tracks | | num_of_tracks |
| price | price | CD_price | CD_price |
| | Artist | artist | artist |
| | | genre | genre |

Add more attributes

**Figure 6: Aligning the attributes of multiple concepts**

Search results for faculty concept

Record View

| | name | title | location | phone | email | research_directorates |
|---|---|---|---|---|---|---|
| Scott Aaronson | Scott Aaronson | Faculty | 32-G638 | | aaronson@csail.mit.edu | Theory |
| Hal Abelson | Hal Abelson | Faculty | 32-386H | 253-5856 | hal@mit.edu | AI |
| Ted Adelson | Ted Adelson | Professor of Vision Sciences | 32-D424/ 46-4115 | 258-9501 | adelson@csail.mit.edu | AI |
| Anant Agarwal | Anant Agarwal | Faculty | 32-G782 | 253-1448 | agarwal@csail.mit.edu | Systems |
| Saman Amarasinghe | Saman Amarasinghe | Faculty | 32-G778 | 253-8879 | saman@csail.mit.edu | Systems |
| Arvind | Arvind | Faculty | 32-G866 | 253-6090 | arvind@csail.mit.edu | Systems |
| Hari Balakrishnan | Hari Balakrishnan | Professor | 32-G940 | 253-8713 | hari@csail.mit.edu | Systems |
| Regina Barzilay | Regina Barzilay | Assistant Professor | 32-G468 | 258-5706 | regina@csail.mit.edu | AI |

**Figure 7: Table view**

## 3.4 Open System for Creating Linked Data

The system helps in creating a linked web of data with the use of URIs. It generates unique dereferenceable URIs for each concept, attribute and instance. Each concept is uniquely identified by the concept name, its creator and the

[7] http://en.wikipedia.org/wiki/Levenshtein_distance

version number (if the same user has defined different versions of the concept).

An example URI for a concept "Car", version 2, defined by the user with ID 1 would be like
*http://www.stylid.org/stylid/concept_detail.php?concept_name=**Car_ver2_1#car***

Similarly, consolidated virtual concepts are also assigned URIs so that they can be uniquely referenced. An attribute is uniquely identified by the concept and the attribute name.

For example the URI for the price attribute of the car concept would be
*http://www.stylid.org/stylid/concept_detail.php?concept_name=**Car_ver2_1#price***

An instance is uniquely identified by the system generated ID for the instance. The URI of an instance is different from the URL of the post showing it. A concept URI dereferences to a page describing the details. An instance URI dereferences to the post showing its details. The details page contains both human readable and machine readable data.

Data instances can be linked to each other by entering resource URIs as attribute values(see Fig.4). The linked data is manifested as simple hyperlinked entries for the user (see Fig.2). However, the linking of URIs helps in the creation of a linked data web, not just hyperlinked pages. The system can link to URIs from any system on the Web. On the other hand, it allows others to link in to its data by providing unique dereferenceable URIs.

StYLiD is an open system that does not lock data into itself. Besides allowing others to link in, the system facilitates the reuse of structured data. Structured information snippets in embedded formats like RDFa may be posted elsewhere or distributed via social media. The system provides an advanced search interface, as shown in Fig.8, which can be used to retrieve instances of a concept specifying attribute, value pairs as criteria. The system also provides a SPARQL query interface for open external access.

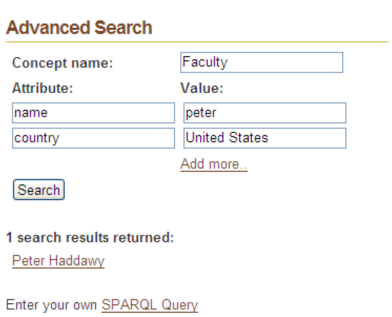


**Figure 8: Advanced search interface**

## 4. IMPLEMENTATION

Fig.9 shows the system architecture of StYLiD. It is built upon a social software platform for harnessing user contributions. The social software provides all the basic features such as content management, assessing popularity of contents, user management, social networking and communication among users. The concept management component enables the users to define their own structured concepts. The component handles the different versions of concepts defined by different users. The structured data management component gathers the instance data contributions from users. The concept management component also handles URI management by assigning each of the concepts and instances a unique dereferenceable URI. The system links structured data items using the URIs. The concept consolidation component consolidates multiple versions of a concept defined by several users. It maps the different versions by aligning attributes and provides a unified interface for the consolidated concept. The structured data embedding component embeds structured data in HTML output using RDFa. RDFa is W3C supported and a comparison with other embedded formats[8] indicates that it is a reasonable choice. The system produces snippets with embedded structured data which can be posted elsewhere. All the concepts and structured data contributed by users are stored in the collaborative data store coupled with the social software. The structured concepts and data are stored as RDF triples in a MySQL database. The system provides some services to exploit the structured data like structured browsing, search and query and allows RDFa driven features discussed in the use case scenario (Section 2.5).

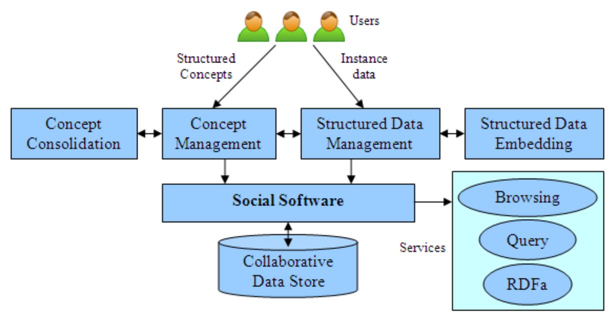


**Figure 9: System architecture**

StYLiD was built upon Pligg[9], a popular Web 2.0 content management system. This open source social software has a long list of useful features and strong community support and, furthermore, provides extensibility. It uses PHP and MySQL. We used the RDF API for PHP (RAP) as a Semantic Web framework to manage structured data.

## 5. RELATED WORK

There have been several recent works on collaborative creation and sharing of structured data on the web. Freebase[10] is one of the most prominent works. Similar to Google Base[11], it allows users to freely define their own structured types and input instance data. However, Freebase keeps the structured types defined by different users separate. It does not consolidate or relate similar concepts. Even concepts having the same name are not shown in a combined way. User defined types and domains are kept within the user's personal space and not easily promoted to the standard types and domain collection. So it is difficult to leverage the structured concepts defined by the large number of users. Moreover, it is difficult for casual users to create their own

[8]http://bnode.org/blog/2007/02/12/comparison-of-microformats-erdf-and-rdfa
[9]http://www.pligg.com/
[10]http://www.freebase.com/
[11]http://base.google.com/

types in Freebase because of the strict constraint requirements. All the attributes must have strict types and the range should be within the types already defined in the system. The attribute and range definitions cannot be altered later if some instances of the concept already exist. Further, it may also be difficult to enter instance data in Freebase because of strict schema constraints. If an attribute takes as value a resource of some type, the resource must be entered first. Although Freebase has made a lot of instance data available by scraping data from vast sources like Wikipedia and MusicBrainz, a non-existing instance must be modeled and entered by the user. Freebase interlinks instance data to each other as attribute values. However, it cannot link to external resources at the data level and it is difficult for other systems to link to Freebase data resources.

The myOntology[19] project proposes to use the infrastructure and culture of Wikis to enable collaborative and community-driven ontology building. It intends to enable general users with little expertise in ontology engineering to contribute. It is mainly targeted at building horizontal lightweight ontologies by tapping the wisdom of the community. However, myOntology is not aimed at collaboratively creating structured concepts and sharing structured data in the community based on that. Freebase and myOntology are both based on Wiki technology. Semantic Wikis, like Semantic MediaWiki[12], IkeWiki[16] and many others[12], further enhance Wikis to make the collaborative knowledge contributed by users more explicit and formal. The relations between resource pages are encoded by semantically annotating navigational links using simple syntax. However, semantic Wikis usually deal with instance data resources but do not consider forming generic schemas for structuring data. Wikis are excellent platforms for creating shared resources collaboratively. However, each concept or resource can only have a single prominent version which everyone is assumed to settle with. In practice, people may have different perceptions about the same concept. Further, users have different information sharing requirements and may need to model the same concept in different ways. StYLiD offers the flexibility and allows users to maintain their own preferences. Takeda et al.[21] had modeled heterogeneous system of ontologies by introducing *aspects*. A *combination aspect* integrates various aspects and a *category aspect* is a collection of aspects about the same thing but with different conceptualizations. They proposed muti-agent communication by translating messages across different aspects.

There had been various works on semantic blogging[4, 13, 11, 18] which exploit the easy publication paradigm of blogs and enhance blog items with semantic structure. Structured blogging[13] also embeds machine readable information in blog entries. Structured tagging techniques, like the Flickr machine tags[14], geo-tagging, triple-tags[15] or dc-tagging[16] try to inject structured information in existing social tagging platforms. However, all these systems deal with very limited types of metadata and the schemas do not evolve.

[12]http://ontoworld.org/wiki/Semantic_Wiki_State_Of_The_Art
[13]http://structuredblogging.org/
[14]http://www.flickr.com/groups/api/discuss/72157594497877875/
[15]http://geobloggers.com/archives/2006/01/11/advanced-tagging-and-tripletags/
[16]http://efoundations.typepad.com/efoundations/2006/10/dctagged.html

Works have been done on deriving ontologies from folksonomies[22, 20]. The basic ideas include grouping similar tags, forming emergent concepts from them, making the semantics more explicit, utilizing external knowledge resources and finding semantic relations. Similar techniques can also be applied on the community-grown concept cloud in StYLiD to have emergent ontologies. Folksonomies serve collaborative organization of objects. Works like MoaT (Meaning of a Tag)[17] try to make the semantics of tags explicit. However, the data objects are still left unstructured. With StYLiD users collaboratively contribute the structure too.

Revyu[7] is a reviewing and rating site where people can review and rate anything. The system generates dereferenceable URIs for *things, reviews, people* and *tags*. Data items can easily be linked with other items using URIs. Revyu produces RDF output and provides a SPARQL endpoint for query. It also exposes reviews using hReview microformat embedded in XHTML. However, most concepts are modeled simply as things. The detailed structure of the information is not modeled and different things are not differentiated.

Exhibit[8] is a lightweight framework which attempts to empower the ordinary users to publish structured information on the Web for effective browsing, visualization and mash-ups. However, authoring such pages would be cumbersome to the users. Potluck[9] is a data mash-up tool for casual users which can align, mix and clean structured data from Exhibit-powered pages. Fields can be merged by simple drag-and-drop, so that different data sources can be uniformly sorted, filtered and visualized. Merged fields are implemented as query unions. We also use a similar technique. Currently, Potluck can only handle Exhibit-powered pages and not dynamic pages and other semantic formats. The schema alignment is manual. We propose to have some automation in schema alignment instead of leaving the entire work to the users. There is a large body of research about schema matching[14] and ontology alignment[5] which can benefit us.

# 6. CONCLUSIONS AND FUTURE WORK

We proposed StYLiD as a single platform for sharing a wide variety of structured data. Users can freely define their own concepts. Relaxing constraints would encourage more user contribution to better meet their requirements. The task of consolidating, aligning and unifying user defined concepts can be handled by the system without bothering the users much. Although several definitions of a concept may exist, the system can provide a single consolidated view so that even heterogeneous structured data can be handled uniformly. It also facilitates the emergence of popular and stable generalized definitions. Keeping the system open and adopting URI conventions support the creation of a linked data web. Thus, even with the informal base of social software we may produce formal machine understandable structured data which can be shared, interlinked and integrated.

In the future, sophisticated schema mapping techniques[14, 5] may be incorporated to better align concept attributes automatically. On the other hand, we are working on maintaining the alignments completed by users collaboratively to utilize human intelligence too rather than relying on sophisticated computations every time. We may also allow users to save aligned unified views customized for their purpose

[17]http://www.moat-project.org/

in their own private space. Better query interfaces could be developed to query and sort instances of consolidated concepts using the combined attributes of such unified views. We may compute relations between concepts based on their structure definitions and instance data. Ideas from works on deriving ontologies from folksonomies[22, 20] may be used. Similar concepts with different names can be clustered together. Synonymous or morphological variants of concept names may be consolidated. On the other hand, ambiguous concept names may be sub-grouped by intended meaning. We can organize concepts into hierarchical domains. Scrapers may be associated to concepts for gathering abundant data from current web pages. Visual scraper creation tools may be provided so that users can easily create and share the scrapers too. We can facilitate users to contribute plugins for handling different types of structured data embedded in the pages. Other useful features, like mash-ups may be introduced to benefit from the structured data. The structured data in StYLiD may also be exposed through an API or extended RSS.

## 7. REFERENCES


[1] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. In *Proceedings of the 16th International World Wide Web Conference (WWW2007)*, Banff, Alberta, Canada, May 2007.

[2] M. K. Bergman. What is the structured web? AI3 Blog, July 2007. http://www.mkbergman.com/?p=390

[3] T. Berners-Lee. Linked data. World wide web design issues, July 2006. http://www.w3.org/DesignIssues/LinkedData.html

[4] S. Cayzer. Semantic blogging and decentralized knowledge management. *Communications of the ACM*, 47(12):48–52, December 2004.

[5] J. Euzenat, T. Le Bach, J. Barasa, et al. State of the art on ontology alignment. *Knowledge Web Deliverable D2.2.3*, 2004.

[6] T. Gruber. Collective knowledge systems: Where the social web meets the semantic web. *Journal of Web Semantics*, 2007.

[7] T. Heath and E. Motta. Revyu.com: A reviewing and rating site for the web of data. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 895–902. Springer, 2007.

[8] D. Huynh, D. Karger, and R. Miller. Exhibit: lightweight structured data publishing. In *Proceedings of the 16th international conference on World Wide Web*, pages 737–746. ACM Press New York, NY, USA, 2007.

[9] D. F. Huynh, R. C. Miller, and D. R. Karger. Potluck: Data mash-up tool for casual users. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *ISWC/ASWC*, volume 4825 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 2007.

[10] A. Iskold. The structured web - a primer. Read Write Web, October 2007. http://www.readwriteweb.com/archives /structured_web_primer.php

[11] D. R. Karger and D. Quan. What would it mean to blog on the semantic web? *Journal of Web Semantics*, 3(2):147–157, 2005.

[12] M. Krötzsch, D. Vrandecic, and M. Völkel. Semantic MediaWiki. In *Proceedings of the 5th International Semantic Web Conference (ISWC06)*, pages 935–942. Springer.

[13] K. Moller, U. U. Bojars, and J. G. Breslin. Using semantics to enhance the blogging experience. In *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 679–696. Springer Berlin / Heidelberg, 2006.

[14] E. Rahm and P. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal The International Journal on Very Large Data Bases*, 10(4):334–350, 2001.

[15] G. Reif, M. Morger, and H. C. Gall. Semantic clipboard - semantically enriched data exchange between desktop applications. In *Semantic Desktop and Social Semantic Collaboration Workshopat the 5th International Semantic Web Conference ISWC06*, Athens, Geogria, USA, November 2006.

[16] S. Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 388–396. IEEE Computer Society Washington, DC, USA, 2006.

[17] S. Schaffert. Semantic social software: Semantically enabled social software or socially enabled semantic web? In *Proceedings of the SEMANTICS 2006 conference*, pages 99–112, Vienna, Austria, November 2006. OCG.

[18] A. Shakya, H. Takeda, V. Wuwongse, and I. Ohmukai. Sociobiblog: A decentralized platform for sharing bibliographic information. In J. a. B. Pedro Isaìas, Miguel Baptista Nunes, editor, *Proceedings of the IADIS International Conference WWW/Internet 2007*, volume 1, pages 371–380, Vila Real, Portugal, October 2007. International Association for Development of the Information Society, IADIS Press.

[19] K. Siorpaes and M. Hepp. myOntology: The marriage of ontology engineering and collective intelligence. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 127–138, 2007.

[20] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In E. Franconi, M. Kifer, and W. May, editors, *Proceedings of the European Semantic Web Conference (ESWC2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.

[21] H. Takeda, K. Iino, and T. Nishida. Agent organization and communication with multiple ontologies. *International Journal of Cooperative Information Systems*, 4(4):321–337, 1995.

[22] C. Van Damme, M. Hepp, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 57–70, 2007.