I.J. Modern Education and Computer Science, **2016, 2, 22-29**
Published Online February 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2016.02.03

Rule Based Communication Protocol between Social Networks using Semantic Web Rule Language (SWRL)

Enesi F. Aminu
Federal University of Technology/ Department of Computer Science, Minna, +234, Nigeria
E-mail: enesifa@futminna.edu.ng

Olaide N. Oyelade
Ahmadu Bello University/Department of Mathematics, Zaria, +234, Nigeria
E-mail: olaide_oyelade@yahoo.com

Ibrahim S. Shehu
Federal University of Technology/ Department of Computer Science, Minna, +234, Nigeria
E-mail: Ibrahim.shehu@futminna.edu.ng

Abstract—Social network sites have become *de factor* in fostering human relationships and business prospects. Several social networks abound with little interoperability functionality that enables exchange of profiles of users. Though, proprietary Application Programming Interfaces (APIs) are provided as endpoints for applications in retrieval of user profile. Moreover, semantic web Friend of a Friend (FOAF) is now been used as a medium for realizing semantic social networks to be able to share user's profile across sites. And since the goal of semantic web is to provide autonomous data centric system coupled on ontology and reasoning, we propose a novel communication protocol named iProc, and usable by autonomous agents that relies on the distributive nature of social network data in coalescing a virtually centralized social network and as well providing means to enlarge user's connectivity to other users across different sites. This paper presents the architecture for a proposed iProc. Furthermore, an implementation of the FOAF files to be used was carried out and discussed.

Index Terms—Semantic Web, Semantic Social Network, Ontology, FOAF, Ontology Web Language (OWL).

I. Introduction

Social networks have become part of human daily activities that are using web technologies to interact with their friends, share pictures, involve in editing wiki pages, reading and commenting on blogs and more through different media such as Flicker or YouTube. Presently, means of storing these data in their distinct formats are no more posing any serious problems and the advantage of analysis overshadow the overhead cost of creating user profiling ubiquitous.[1] This whole paradigm shift is often more technically described as Web 2.0 [2]. However, a new trend from the research space during the last decades refer to as the semantic web emerged [3], meant to provide models for interoperable data between applications and can be of great interest for communities from the Social Web. By relying on standard models to represent data as well as shared semantics between applications, it presents a better platform to integrate and query data from different systems, as well as creating links between them. The online communities can better understand through the Semantic Web technologies, by providing general means to link and represent information from various distributed systems and heterogeneous data sets [4].

Generally, combination of semantic web and social web can be considered in two different perspectives: on the first perspective, some efforts is directed on using semantic web technologies to model social data, with models for instance Friend Of A Friend (FOAF) [5] and Semantically-Interlinked Online Communities (SIOC) [6], social web data can be represented using shared and common models, and then it becomes more interoperable and portable between applications. While on the other perspective is taken the advantages of the wisdom of the throng from Web 2.0-based services which is an ideal prospect for creating a large amount of Semantic Web data.

As earlier noted, representing social networking information using Semantic Web technologies, such as FOAF [4] and probably the most well-known ontology, it provides a model to represent people (with a foaf: Person) class, their properties and attributes (ranging from foaf: name to foaf: school Homepage) as well as a foaf:knows relationship which is used to represent social networking aspects. This last relationship is semantically weak, and to overcome this lack of precise semantics, the RELATIONSHIPvocabularyhttp://vocab.org/relationship/ - provides a set of subproperties such as rel: colleague Of or rel: life Partner Of to describe more precise relationships between people. Moreover, since ontologies can be extended in a distributed manner, anyone can create his or her own property, for example, wrote A Chapter With could be used to identify people in a social network as being co-authors. To this end, the proposed novel communication protocol refers to as iPROC, that relies on the distributive nature of social network data in coalescing a virtually centralized social network where a user's connectivity is enlarged across different social network.

II. Related Works

According to the work of [7] titled analyzing the role of semantic web in social networking sites, the paper is primarily concerned with the semantic web architecture, tools, technology, and its basic aim to aggregate the electronic data sets that collects about the social networks and its roles within the architecture of semantic web considering the progressive increment in the sites of social networks. However, the paper did not discuss the platform of harmonizing or interacting with a user's data of a social network across other different social networks.

In the work of [8] they proposed an architecture for an open, distributed social network, which is built solely on Semantic Web standards to address the problem of online social networking where users do not have full control over their data and are bound to specific usage terms of the social network operator and suffer from a lock-in effect due to the lack of interoperability and standards compliance between social networks.. The architecture combines vocabularies and protocols such as WebID, FOAF, Semantic Pingback and PubSubHubbub into a coherent distributed semantic social network, which as claim capable to provide all crucial functionalities known from centralized social networks. However, we propose a novel communication protocol and demonstrate its workability through accessing FOAF of some social networks.

Considering the work of [9], they proposes a semantic context-based access control model (denoted as SCBAC) to be used in mobile web services platform by coalescing semantic web technologies with context-based access control mechanism. The researchers proposed context ontology to characterized contextual information and utilize it in the inference engine so as to take care of context information in the model. Also, the work aimed at specifies access control policies as rules over ontologies representing the concepts introduced in the model, and uses semantic web rule language (SWRL) to form policy rule and infer those rules by JESS inference engine.

Also, attention was equally given to the article of [10] on knowledge systems and retrieval techniques where knowledge is described as rules of objects. The aim of the article is to make available a platform for further research work on improved knowledge retrieval techniques for knowledge systems by integrating intelligence. However, the researchers admitted that ontology and semantic web technologies have immensely added a great value to information sharing and easier accessibility of data. Although they noted deficiency associated with the technologies.

Finally, in the research work of [11] that centered on social communication (for example, messengers, face book, etc), they described the social network as hi-tech society that has little or no semantic contents. Therefore, they devised a semantic framework that can eradicate the shortcomings of the present semantic nature such as the chances of data interference between preferred and prowler users. Whereas, in our own case; we are proposing a rule based communication protocol for engaging social networks using SWRL.

III. The Architecture Iproc and it Working Techniques

An architectural overview of the proposed communication protocol working environment is presented in this section. A three-layered system is illustrated in Figure 1, with all strata abstracting real life components. The system consists of two basic events/activities: friending and multiple account aggregation. This makes the proposed iProc a twofold communication protocols. Each of the protocol is tunneled through different semantic web based agent-like node that modulates the underlying rules. It translates and transforms their different internal working structure into a unified collector. iProc-F and iProc-C are the two resulting protocols from iProc and they shall be further discussed in subsequent sections. The entire system overview is dotted with edges and nodes through which signals are being sent across from one layer to another. A simple flow of signal within the system starts from what is being referred to as a master agent embed on user's mobile device. Depending on the event being invoked by the master agent, one slave agent will be involved. For instance, when friending event is invoked by the master agent, a friend request message must be sent to slave agent. All the nodes in the system are numbered to indicate the flow of a chosen event.

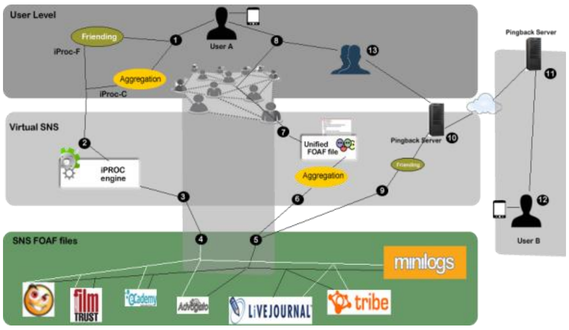

Fig.1. System Overview

Formally, the iProc system may be model $iProc_{sys} = P = < U, N, V, KB, S>$ where U is a set of varying SNS account holders or user $\in \{u_0,\ldots u_n\}$, N denotes a set of nodes (this nodes vary in their nature and functions) $\in \{n_0,\ldots n_n \mid \forall n_i$ {Engine|Foaf_file}}, V denoting vertices or channels for signal passage V $\in \{v_0,\ldots v_n\}$, KB represents several knowledge bases from which our foaf files are being sourced from while S includes the Semantic Pingback server and other relevant servers required in the system.

Data Source

The semantic web supports the aggregation of data that are distributed across several semantic web data repositories. Social networks form a distributed data sources which most often do not share information among themselves. This seems to lock in account holders from having a cross platform data access and sharing privilege. We present a centralized data access and sharing system that virtualizes and synergize user's profile thereby enabling a user with multiple accounts on different SNS to perform some in-house tasks (such as friending and viewing of personal network association) and untie them from soloed services. In the SNS foaf file layer of Figure 1, some social networks are identified as sources of data for the system. Social networks such as Rossia.org, LiveJournal, InsaneJournal, GreatestJournal, eCademy, Advogato, Tribe and Minilog.com, provides their users profile information as FOAF files. Note that FOAF supports the linking of data, information and people across the web [12]. This leads to a machine readable data source and also creates an avenue for agents to crawl over them so as to source specific information to a centralized system. Though not all FOAF files from these SNSs provide a full list of their members, however FilmTrust, Ecademy, and Advogato do make this provision in their FOAF files. Furthermore, another viable data sources that can be plug into this system as illustrated in Figure 1 includes Semantic data wiki and Semantic web-based retrieval engine like Sindice and Swoogle. OntoWiki is a good example of Semantic data wiki that also serves as a Linked Data publishing engine.

Friending

Friending here means the process of finding a friend and accepting friending requests on social networks. In this work, we propose a framework for connecting social network users from different SNS. The framework uses WebID, the Semantic Pingback technology, a Pingback server, user devices (majorly mobile device), and some categorized FOAF files. WebID is a distributed identification and users connecting approach in social network, and it provides users with authentication and access contrivances [13]. Identification through WebID is done through URI, this makes the WebID protocol request one more additional HTTP connection aside the one used in making original request. This approach helps users manage their profile from another point. On the other hand, Pingback technology was oriented by the blogosphere which provide support interlinking on the web and enriching end users with the advantages of the Linked Data Web [14]. Semantic Pingback in distributed semantic social network (DSSN) is to engender the connectivity, for the first time, of different users within that same network.

The WebID file is used in this system for identification of users. Users profiles are encoded in the foaf file contained in the WebID. The friending process goes thus: User A indicates in his WebID that it desires a friendship request with User B. this indication is typified using the foafq: knows construct. Then User B's WebID is relayed with this request using the pinging structure on their respective servers. When alerted of User A's friend request, User B may consider accepting this request by effecting it in their WebID that he also know – foaf:knows – User A. correspondingly, a pingback message is sent to User A informing him of User B's acceptance of the friending request earlier sent out. The inter-network communication described here is being engineered by the Semantic web concept which allows distributed information to be linked together or exchange communication pattern that is winged upon intelligent agents.

However, in our case we propose DSSN that is virtualized on the existing social networks available in the third layer labeled SNS FOAF files. The argument here is each user maintains preferred host for their WebID, and then publish all their multiple social network accounts in this WebID. Then whenever they initiates a friending request, prospective recipient of such request will be drawn from varying social networks and may be viewed from their clustered form according to the name of their social network. Once a friend is fingered from any social network as a recipient of such friending request, a ping message is sent to the server hosting the WebID of the to-be-friend. And once the to-be-friend friend receives the request and accepts it by publishing it in his WebID, the serving hosting the WebID of this to-be-friend then sends a pinback message to the WebID of former friend who initiated the friending task. This allows social network users to move out of the siloed or locking experience in un-interacting social networks. We imagine that a cross platform friend network might be viewable by each user. We meant to say that say User A has an account with Ecademy and has 100 associates or friends while User B has an account with LiveJournal and maintains 50 associates or friends. We assume that in reverse, User A also has account with LiveJournal and maintains a list of 20 friends while User B has 40 friends on Ecademy. Then User can from one screen view on his mobile device, navigate through all the friends maintained by User B across all the social network platform User B is registered. This also holds for User B with respect to User A's friends. We note that their WebID will serve as good miniaturized repository to store information which could help retrieve this view.

Mathematically, a model to show the friending process described above is denoted by f(x). That is:

$$f(x) = t\left(R_f D_v\right) + s\left(\sum (KB), W_{id}\right) + F \tag{1}$$

The R_f and D_v in t stands for the time making friending request and displaying the prospective friends. While the search cost on all knowledge bases (KBs) that are involved and the WebIDs of the participating users of this friending task, is also added to the $f(x)$. Finally, the acceptance and consummation of the friendship is denoted by F.

Multiple User Profile Aggregation/Federation

While we seek to discourage a centralized social network that tends to close-in users, this work does not propose another form of federation of profiles. However, DSSN nodes a canvassed for, and from which users can view a virtualized or one time aggregation of cross platform (multiple social networks) profiles.

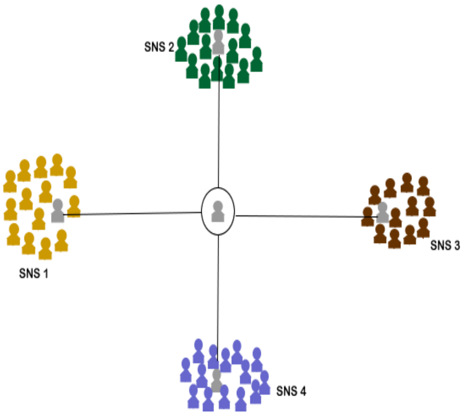

Fig.2. Profile Federating Effects

In Fig.2, a profile clustering or federation effect is being illustrated. Here again, the clustering effect is virtualized within the second layer of Figure 1. A user's multi SNS accounts information are held in their WebIDs and from which login details into those respective SNSs can be retrieved. So, when User A logs into his DSSN node controlled app, his profile login details are retrieved from the WebID and a remote access to each of his profiles on all the SNSs he has account with. While user profile retrieval is being made from each of these SNS, the corresponding friends or associates this user owns are also retrieved along. Figure 2 shows a specific user who has established connection with four accounts he has with four different SNSs, and the network associates of each of those accounts on their respective SNS are brought into view.

Mathematically, a model to show the multi SNS accounts clustering process described above is denoted by *f(y).*

$$f(y) = \left(\sum_{i=0}^{A}(\beta^{x})\right)\left(\sum_{k=0}^{C} S\right) \text{ such that } S \in \{SNS_1, SNS_2 \dots SNS_n\} \qquad (2)$$

The first summation represents the weight of gathering the associates and their respective profile details (images, names, contact addresses and other relevant information that their security setting is public) of the user in each SNS. The second summation denotes the clustering effects of results of each of the SNS into the virtual layer for overall view. Section IV will enumerate the proposed rules (IPROC-C and IPROC-F) that will be used in achieving it.

IV. Reasoning Over Foaf Using Iproc

In this section, we describe the structure of the proposed communication protocol called *iProc*. *iProc* is a two-faceted rule and is partitioned into *iProc-C* and *iProc-F*, though an all-purpose rule, listed in Table 1, that both *iProc-C* and *iProc-F* uses to precede their implementation. SWRL portends some form of limitations in its expressivity as a Semantic Web rule language. For instance, deletion of a statement, the negation of a fact, and the ordering rules in a rule set or exclusion of a specific rule from a rule set when a condition is met, is not expressible in SWRL. Moreover, the Close World Assumption (OWA) holds in both OWL and SWRL while in Open World Assumption (CWA) is not realizable in OWL and SWRL. The concept of monotonicity has a closer tie with OWA while that of non-monotonicity is more supports the CWA method. In OWA a statement which holds false from the knowledge base, may not be confidently assumed to be false because another statement subsequently added to it may contradict it. Meanwhile, in CWA, it is safe to hold that conclusion that such a statement is false or absent.

Now, because of the lack of support for describing a rule to show non-monotonicity of knowledge in a given domain, these will invariable affect the knowledge management of this knowledge base. Hence [15] presented some more operators which includes as NotExist, Exist Dominance and Mutex and also developed a prototype rule engine called JNOMO for the implementation of these operators. These operators are extensions to SWRL support the implementation of the CWA approach while the JNOMO is an extension to the Jena rule engine. Hence, this paper adopts the use of some of the new operators in [15] and also makes use of makeSet constructor demonstrated in [16].

Table 1 outlines the general rule that both *iProc-C* and *iProc-F* invoked before implementing their contextualized rule set. R1 forms an array of sets that in them consist of user profiles from some choice SNSs FOAF files highlighted in the third layer of Figure 1 while R2 specifically retrieves from all applicable SNSs the profile of this user who has login. R1 is an event that invokes the execution of its corresponding coincidence. While R2 retrieves the basic profile details of *UserName*. Note that the use of the word *UserName* is to represent this very user who have login into the application that implements the architecture proposed in this paper.

Table 1. Rule Set (General)

Comments	**Rules**
Apply to all ontology	$R1 \rightarrow foaf{:}Person(?p) \wedge \mathrm{foaf{:}mailbox}(?\mathrm{p}, ?\mathrm{ml}) \wedge \mathrm{foaf{:}Image}(?\mathrm{img})$ $\wedge\, \mathrm{foaf{:}depicts}(?\mathrm{img}, ?\mathrm{p}) \circ \mathrm{makeSet}(?\mathrm{s}, ?\mathrm{p})$
Apply to temporary ontology	$R2 \coloneqq foaf{:}Person(UserName) \wedge \mathrm{foaf{:}mailbox}(UserName, ?\mathrm{ml})$ $\wedge\, \mathrm{foaf{:}Image}(?\mathrm{img}) \wedge \mathrm{foaf{:}depicts}(?\mathrm{img}, UserName) \rightarrow sqwrl{:}select(?ml, ?img)$

Table 2 list out the set of rule that *iProc-C* is being driven by. This rule set consists of RC2, RC3, RC4 and RC5 which are events that allows for the implementation of their coincidence. RC2 invokes its coincidence which states that user ?p whose email address is *mail* and who does not have any friend association with *UserName* does not exist. One possible event RC2 could imply is a search for *UserName* in the knowledge base. If RC2 returns a truth result, then RC3 applies that *UserName* and ?p are same individuals. RC4 retrieves all the friends of the *UserName* from the selected SNS. Meanwhile, should RC2 return a false result, then RC5 indicates in the knowledge base that *UserName* and ?p are different individual and so, there is no need to cluster their accounts together.

Table 3 displays a listing for the iProc-F communication protocol and it consists of rules RF2, RF3 and RF4 all in a rule set. RF2 invokes its coincidence which states that *UserName* and an unknown user denoted by ?p are not friends in as much as the email address of ?p is same as *UserName*. Possible event RF2 could stand for is a situation when a friendship does not exist between *UserName* and ?p, or when either of *UserName* or ?p is de-friending each other. Event RF3 launches its coincident which enlist for next action, all user profiles who have no friend relation with *UserName*. Afterward, RF4 then apply the friending operation on all those user profiles selected by the coincident of RF3. And *foaf:knows* construct is used to specify this.

Table 2. Rule set for Clustering [iProc-C]

Comment	Rule
Search for user in each ontology	$RC2 \rightarrow NotExist(foaf{:}Person(?p) \wedge \mathrm{foaf{:}mailbox}(?p, mail)$ $\wedge\ \mathrm{not}(\mathrm{foaf{:}knows}(?p, UserName)))$
Apply same as	$RC3 \rightarrow foaf{:}Person(?p) \wedge \mathrm{foaf{:}mailbox}(?p, ?ml) \wedge \mathrm{foaf{:}Image}(?img)$ $\wedge\ \mathrm{foaf{:}depicts}(?img, ?p) \wedge \mathrm{sameAs}(?p, UserName)$
Cluster user's friends apply differentFrom	RC4 $\rightarrow foaf{:}Person(?p) \wedge \mathrm{foaf{:}knows}($ $RC5 \rightarrow foaf{:}Person(?p) \wedge \mathrm{foaf{:}mailbox}(?p, ?ml) \wedge \mathrm{foaf{:}Image}(?img)$ $\wedge\ \mathrm{foaf{:}depicts}(?img, ?p) \wedge$ differentFrom (?p, username)

Table 3. Rule set for Friending [iProc-F]

Comment	Rule
UserName cannot friend self	$RC2 \rightarrow NotExist(foaf{:}Person(?p) \wedge \mathrm{foaf{:}mailbox}(?p, mail)$ $\wedge\ \mathrm{foaf{:}knows}(?p, UserName))$
All profiles that are not friends with UserNames	$RF3 \rightarrow foaf{:}Person(?p) \wedge \mathrm{not}(\mathrm{foaf{:}knows}(UserName, ?p) \wedge$ $\mathrm{not}(foaf{:}mailbox(\mathrm{mail}))$
For each ?p inRF3, *UserName* knows them	$RF4 \rightarrow foaf{:}Person(?p) \wedge \mathrm{PersonalProfileDocument}(?doc)$ $\wedge\ \mathrm{page}(?doc, ?p) \wedge\ \mathrm{foaf{:}knows}(UserName, ?p)$

An algorithm in fig.3 combines the application of general rule and the two rules sets outlined above. The *def* keyword is used to define global variables and settings. The initial statement reveals that Pellet is used, and so all necessary connection to Pellet must be done here. Each of the command in the algorithm is labeled with a step number just to follow normal conventional way of writing algorithm and as well for readability. Step 1 stores in *v* the required action from the user; this will either be a friending or clustering action. In reasoning over multiple files as required by the nature of the rules stated above, Pellet's $\varepsilon - \mathrm{Connection}$ is employed in carry out this task, hence the need for Step 2. R2 in Step 3 is applied on a temporary FOAF file which resides on the server. And it retires the general profile or WebID based profile details of *UserName.* Depending on the operation chosen by the user the **then** lines of instructions are executed. For instance when multi SNSs profiles or account clustering is the users' option, steps 5 and 6 are executed otherwise, steps 7 and 8 are executed for friending operation. In clustering, the *dominance (…)* operator ensures that rules RC2, RC3, RC4 and RC5 are executed in that order. Recall that dominance and mutex operators were discussed in section 4. And for each instance of SNS's FOAF files RC2 is applied and a truth value result will eliminate RC5 from the rule set executing RC3 and RC4 while a false return value will eliminated rule RC3 applying RC5 in the rule set. However, if the friending operation is chosen, step 7 uses the dominance operator to indicate the orderly manner RF2, RF3 and RF4 will be executed. And again, for each instance of the SNS's FOAF files rules RF2, RF3 and RF4 are applied. Finally step 9 outputs the result for view on the client side.

```
            def: Using Pellet reasoner,
Step 1:     Let v = user input (clustering or friending)
Step 2:     Use $\varepsilon - \mathrm{Connection}$ to apply R1 multiple
            FOAF files (different SNS)
Step 3:     Apply R2.
Step 4:     if v == clustering then
              Step 5:   dominance(RC2, RC3, RC4,
                        RC5)
              Step6:    foreach foaf-ontology
                          t=apply RC2
                          if t then
                          mutex(RC5, RC3) apply RC4
                         else mutex(RC3, RC5)
                        end-foreach
            else
              Step 7:   dominance(RF2, RF3, RF4)
              Step8:    foreach foaf-ontology
                         apply RF2,apply RF3, apply RF4
                        end-foreach
Step9:      output result
```

Fig.3. Algorithm

V. Implementation

The implementation of this research is partly presented here as seek to deploy it as an application deplorable on mobile devices so as to enable users have a quick access to this one-stop multi SNSs profiles access point. A user's WebID file is described and a sample FOAF file that shows friending and multiple profile clustering are shown in this section.

WebID protocol combines FOAF+SSL and speaks of the certificate that is used by communicating servers to establish trust and identification of user. For example, should User A wants to communicate or connect with User B, then a HTTPS request sent to User B's server will demand for the certificate of User A. afterward, the private key embedded in User A's browser is compared with the public contained in the certificate. Then the WebID, through its URL (e.g.) http://myisp.com/whatever/foaf.rdf#UserA) stored in the Subject Alternative Name (SAN) compartment of the certificate, helps to retrieve the FOAF document which confirms that there exist a relation that shows that the public key belongs to User A .We assume that user has a reliable web host to host the WebID file – this could be a personal server or a rented server. Listing 1 exemplifies how user A's profile is being described in his FOAF file while listing out his personal details.

```
<?xml version="1.0"?>
<rdf:RDF      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/">
  <foaf:Person
rdf:about="http://myisp.com/whatever/foaf.rdf#UserA">
       <foaf:name>UserA Emmy </foaf:name>
       <foaf:title>Mr.</foaf:title>
       <foaf:givenname> UserA</foaf:givenname>
       <foaf:firstName>UserA</foaf:firstName>
       <foaf:family_name>Emmy</foaf:family_name>
      <foaf:img
rdf:resource="http://myisp.com/whatever/emmyfile/usera.jpg"/>;
   <foaf:mbox
rdf:resource="mailto:usera@myisp.com"/>
      <foaf:homepage
rdf:resource="http://www.usera.com"/>;
  </foaf:Person>
</rdf:RDF>
```

Listing 1. Web ID Profile Showing User A's Web ID and Personal Details

It was highlighted in the last paragraph that User A's WebID is http://myisp.com/whatever/foaf.rdf#UserA and Listing 1 captures this, however to link the WebID to his FOAF file, Listing 2 which is a continuation of Listing 1 uses foaf:PersonalProfileDocument to declare User A's FOAF file as http://myisp.com/whatever/foaf.rdf. This provides us to model that User A owns the FOAF file and that he is the maker of such FOAF.

```
<foaf:PersonalProfileDocument
df:about="http://myisp.com/whatever/foaf.rdf">
   <dc:title>UserA's FOAF Profile</dc:title>
   <foaf:maker
rdf:resource="http://myisp.com/whatever/foaf.rdf#UserA
" />
   <foaf:primaryTopic
rdf:resource="http://myisp.com/whatever/foaf.rdf#UserA
" />
</foaf:PersonalProfileDocument>
```

Listing 2: Connecting User A to FOAF File Ownership

```
<foaf:knows>
     <foaf:Person
rdf:about="http://userb.com/userc.com/me">
       <foaf:name>User C</foaf:name>
     </foaf:Person>
</foaf:knows>
<foaf:knows>
     <foaf:Person rdf:about="http://userb.com/me">
       <foaf:name>User B</foaf:name>
     </foaf:Person>
</foaf:knows>
```

Listing 3. Describing User A friends

One of the concepts being proposed in this paper is a virtualized platform for aggregating multi SNS profile friend network of a particular user. Listing 3 shows two friends (User C and User B) that User A knows and has created friendship with on this virtualized platform. Observe that the users that comprises of this virtualized friendship network crosses across different SNS. However, when a request to view their profile details is demanded by the user (User A) viewing it, then these full details are drawn from their respective SNS databases. As discussed above, User A's certificate will be required by the server receiving his GET command. Hence, in Listing 4, one of the W3C certificate and crypto ontology construct cert:key is being used to represent the RSA public key description for retrieval by the server that User A might be communicating with.

```
<cert:key>
  <cert:RSAPublicKey>
    <cert:modulus
rdf:datatype="http://www.w3.org/2001/XMLSchema#hexBinary">B88……D1”
     </cert:modulus>
     <cert:exponent
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
        65537
     </cert:exponent>
  </cert:RSAPublicKey>
 </cert:key>
```

Listing 4. A Segment of User A's Web ID Describing RSA Public Key

```
<foaf:Person>
-           - - - - - - -
  <foaf:holdsAccount><foaf:OnlineAccount>
     <rdf:type
rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAccount"/><foaf:accountServiceHomepage
rdf:resource="http://www.sns1_site.com/index.shtml"/>
```

```
     <foaf:accountName>usera</foaf:accountName>
<foaf:mbox_sha1sum>50960wqww…w43</foaf:mbox_s
ha1sum></foaf:OnlineAccount>
   <foaf:OnlineAccount><rdf:type
rdf:resource="http://xmlns.com/foaf/0.1/OnlineChatAcco
unt"/><foaf:accountServiceHomepage
rdf:resource="http://www.sns2_site.com/index.shtml"/>
     <foaf:accountName>usera</foaf:accountName>
<foaf:mbox_sha1sum>50960wqww…w43</foaf:mbox_s
ha1sum></foaf:OnlineAccount></foaf:holdsAccount>
-          - - - - - - - - -
</foaf:Person>
```

Listing 5. SNS Accounts Details of User A

A user may have multiple accounts on different SNSs and then configure to virtualizes the view of such accounts from a window in this propose system. Hence, the FOAF file of such user must keep track of those accounts the user owns on the SNS that applies to him. Listing 5 shows a minimal outline of some accounts UserA owns on different SNSs. This section of the FOAF file will be needed when UserA wants to cluster his profiles across all SNSs where he holds an account. Moreover, during friending, UserA can only connect with new friends which will be drawn from within the SNSs that he holds account. The foaf: holds Account is used to indicate that the Person (UserA) holds the list of accounts listed by foaf: Online Account within the foaf: holds Account open and close tags. Basically, the obfuscation of email address of UserA which is contained within foaf:mbox_sha1sum, and the username which is contained in foaf:accountName are going to be retrieved in the course of accessing each of these accounts on their various SNS.

The clustering effect or user profiles aggregation operation described in Section 3.0 is being modeled as shown in Listing 6. First, a rule through the OWL restriction to is applied to state that all members of a group named SNS 1 and must belong to a class called SNS1 and in turns must be a Person should have each members have their value of the foaf: workplace Homepage property to be http://www.sns1_site.com. The implication of this rule is to make sure that users on SNS1 are clustered into one group while users on another social network say SNS2, are also clustered into a group. This will provide means to programmatically retrieved members based on their group listing. Subsequently, we used foaf: member to list out the members of each group.

```
<foaf:Group>
 <foaf:name>SNS1</foaf:name>
<foaf:membershipClass><owl:Class
rdf:about="http://serverdname.com/groups#SNS1">
    <rdfs:subClassOf
rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <rdfs:subClassOf><owl:Restriction><owl:onProperty
rdf:resource="http://xmlns.com/foaf/0.1/workplaceHome
page"/><owl:hasValue
rdf:resource="http://www.sns1_site.com"/>
      </owl:Restriction> </rdfs:subClassOf>
  </owl:Class> </foaf:membershipClass>
 <foaf:member><foaf:Person>
<foaf:name>UserC</foaf:name><foaf:homepage
rdf:resource="http://www.sns1_site.com/people/userc/"/>
  <foaf:workplaceHomepage
rdf:resource="http://www.sns1_site.com"/>
</foaf:Person>
     -        - - - - -
 </foaf:member> </foaf:Group>
```

Listing 6. Description of Clustering Effect

Now that we have considerably discussed the implantation of the FOAF file, it must be noted that the client side and the server side that will handle the architecture discussed in Section 3.0 are still underway. As earlier stated, the client side is being anticipated to be deployed on user's mobile device so as to support ubiquity of the application.

VI. Discussion and Evaluation

The discussion and possible evaluation of this section will be centered on the two major operations this paper highlights: Friending and user profile aggregation. The architecture layout shown under Section 3.0 pictorially summarizes the workings of these two concepts. In friending, this research proves that trans-association operation among varying user profiles on different social networks is possible. Though our friending is being virtualized (that is, not recreating another database of user profiles, for that would have created data duplication and redundancy) within a user's session – except if the user saves each friending operation – it however removes the restriction placed on sending friend request to only members of the same SNS – a siloed friending operation. This demonstrates that data sharing among users of one SNS can as well be shared and viewed by users in another SNS given that they are friends on the virtualized platform. This friending operation is made possible by the combine functions of the iProc-F, Algorithm 1 and the FOAF file described in the listings in Section 5.0.

On the other hand, the user profile aggregation or federation operation provides a user with a window view of his accounts or profiles that resides within different social networks. This operation is built upon the rule named iProc-C, the FOAF file listed above and Algorithm 1. This one-window view become a springboard from which cross platform activities such as messaging, chatting, postings, and commenting can be built upon. Basically, the clustering effect discussed in this paper is simply for view and no any activities attached to it.

VII. Conclusions

In this paper, a decentralization of social networking activities is being canvassed for. This is in line with the DSSN concept slightly mentioned in previous sections. We stress that decentralizing such networks benefits the users to manage from a point, their multiple accounts and as well have the freedom to share information over different social network. It must be noted that though every social network company have some privacy settings and policies through which user profiles are shielded from attacks like account/face cloning or identity theft, digital dossier of personal information and lost more, however, a more tighter security measures must be deployed on the client side of such application as being promoted in this paper. This issue of security threats suffered by social networks users is not covered in this paper. However, it might be some holes needed to be considered as future works.

REFERENCES

[1] S. Punagin and A. Arya “Privacy in the age of Pervasive Internet and Big Data Analytics – Challenges and Opportunities” *I.J. Modern Education and Computer Science*, 2015, 7, 36-47, 2015.

[2] T. O’Reilly “O’Reilly Network: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Retrieved from http://www.oreillynet.com/lpt/a/6228”, 2005.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. “The Semantic Web.” *Scientific American*, *284*(5), 34–43, 2001.

[4] A. Passant; S. Kinsella; U. Bojars; J.G. Breslin and S. Decker. Understanding Online Communities by Using Semantic Web Technologies, 2011.

[5] D. Brickley, and M. Miller. FOAF Vocabulary Specification. Namespace Document 2 Sept 2004.

[6] J. G. Breslin, A. Harth, U. Bojars, and S. Decker. “Towards Semantically-Interlinked Online Communities.” In *Proceedings of the 2nd European Semantic Web Conference (ESWC2005).* Springer. 2005.

[7] R. Sharma. “Analyzing the Role of Semantic Web in Social Networking Sites.” *International Journal of Scientific Engineering and Technology. Vol. 1 Issue No. 3, Pg: 125-131,* 2012.

[8] S. Tramp, P. Frischmutt, T. Ermilov, S. Shekarpour, and S. Auer. “An Architecture of a Distributed Semantic Social Network.” *Universität Leipzig, Institut für Informatik, Leipzig, Germany* 2012.

[9] H. Shen and Y. Cheng. “A Semantic Context-Based Model for Mobile Web Services Access Control” *I.J Computer Network and Information Security,* 2011, 1, 18-25.

[10] M. Malhotra and T.R. Nair. “Evolution of Knowledge Representation and Retrieval Techniques” MECS *I.J. Intelligent Systems and Applications,* 2015, 07, 18-28.

[11] F. Giunchiglia, S. H. Mukta, T. Nayeem, and K. T. Hasan. “Semantic Enabled Role Based Social Network” *I.J. Intelligent Systems and Applications,* 2012, 12, 1-11.

[12] FOAF Specification. http://xmlns.com/foaf/spec/ Looked up on 27th December 2014.

[13] W. Stefan, C. Olexiy, H. Sebastian, G. Martin. “Customized Views on Profiles in WebID-Based Distributed Social Networks” Proceedings of 13th *International Conference, ICWE*, pp. 498-501, 2013.

[14] Semantic Pingback Technology. http://aksw.org/Projects/SemanticPingback.html. Looked up on 2nd January 2015.

[15] J.M.A. Calero, A.M. Ortega, G.M. Perez, J.A.B. Blaya, A.F.G Skarmeta. “A Non-monotonic Expressiveness Extension on the Semantic Web Rule Language” *Journal of Web Engineering. Volume 11 Issue 2, pp. 93-118,* 2011.

[16] M. O’Connor and D. Amar. SQWRL: a Query Language for OWL, *Stanford Center for Biomedical Informatics Research, Stanford* retrieved from*suanpalm3.kmutnb.ac.th/teacher/FileDL/supot162255416363.pdf*, on 5th January, 2015.

Authors’ Profiles

Enesi F. Aminu is presently lectures at Computer Science Department, Federal University of Technology, Minna, Nigeria. Among courses teaching are Operating Systems, Database Design and Management, Object Oriented Programming, and Seminar. He obtained both his B.Sc and M.Sc degrees in Computer Science from University of Jos, Jos and Ahmadu Bello University, Zaria respectively. His current research interest is on ontology languages and semantic web contents. Also a member of these professional bodies: Nigeria Computer Society (NCS) and International Association of Computer Science and Information Technology (IACSIT).

Olaide N. Oyelade is currently a research student (PhD) in computer science of the Department of Mathematics Ahmadu Bello University, Zaria-Nigeria. His current research interests include Semantic Web technologies, specifically ontology languages and their query and rule languages. Also, he has research interest in mobile computing. Oyelade earned his B.Sc. and M.Sc degrees in computer science from University of Jos, and Ahmadu Bello University, Zaria, Nigeria in 2010 and 2014 respectively.

Ibrahim S. Shehu currently lectures in the Department of Computer Science, Federal University of Technology, Minna, Niger State, Nigeria. He obtained his M.Sc. degree in Computer Science and Entrepreneurship from the University of Nottingham in 2012. His current research motive is in semantic web technologies and usability studies. Other research interest include; the application of Decision Support Systems (DSS), Expert systems (such as Adaptive Hypermedia Systems) in education and Multi agent systems.