A Uniform Semantic Web Framework for Co-Authorship Networks

Lule Ahmedi
Department of Computer Engineering
University of Prishtina
Bregu i diellit pn, 10000 Prishtinë
Kosova
lule.ahmedi@uni-pr.edu

Lejla Abazi-Bexheti, Arbana Kadriu
Department of Computer Science
South East European University
Ilindenska pn, 1400 Tetovë
FYR Macedonia
{l.abazi, a.kadriu}@seeu.edu.mk

***Abstract*—Modeling co-authorship networks merely in FOAF, as is common with social networks, suffers the inability to capture semantics which are specific to collaboration schemes in the scientific community. For instance, the weight of the co-author relation which expresses the magnitude of that relation between two co-authors is subject to change dynamically as new publications are assigned to the co-authorship graph. They require mechanisms like rules in Semantic Web to calculate them. We introduce here a framework of (1) extending the FOAF ontology, and (2) implementing SWRL rules to construct *weighted* co-authorship networks which will contain metrics like exclusivity, frequency, or weight, as suggested in the literature. It makes it easy for the extended FOAF to integrate it with the native SWRL rules, and provide thus a uniform framework based solely on the Semantic Web for the representation and analysis of weighted co-authorship networks. An evaluation of our framework run on real data is provided.**

Keywords-FOAF, SWRL rules, Ontology modeling, Co-authorship networks, Social networks, Reasoning over Semantic Web

I. INTRODUCTION

Semantic Web as conceived by Tim Berners-Lee [1] is permeating more and more attention not just by the research community, but also in the landscape of real-life applications. Friend of a Friend (FOAF) ontologies [2] represent a prominent application of the Semantic Web, and are extensively being used to represent social networks on the Web [3][4]. Co-authorship networks make up an important subset of the tantalizing set of data found in Web-based social networks. Applying social network analyses to the co-authorship networks has been investigated in finding interesting facts about authors of scientific publications and relationships among them, like in [5][6].

Modeling co-authorship networks merely in FOAF suffers the inability to capture semantics which are specific to collaboration schemes in the scientific community. For instance, the weight of the co-author relation which expresses the magnitude of that relation between two co-authors is subject to change dynamically as new publications are assigned to the co-authorship graph. Mechanisms like reasoning with rules in Semantic Web are thus required to calculate weights. Studies to deploying Semantic Web frameworks for the representation and analysis of social networks in general exist [7][8], rather not aimed at considering rules in Semantic Web as a mechanism to support weighted co-authorship networks as a special class of social networks.

We present here a novel approach of (1) extending the semantics of the FOAF ontology and (2) deploying the SWRL (Semantic Web Rule Language) [9] rules to represent and analyze weighted co-authorship networks.

The rest of the paper is organized as follows: We first introduce our approach in general in the context of related work, followed by a short review of the background work required to understand the rest of the paper. We then provide CO-AUTHORONTO, our extension of the FOAF ontology to capture the semantics of weighted co-authorship networks. Further we detail rule definitions in SWRL to infer values of such added semantics, and assign them to the underlying CO-AUTHORONTO ontology. Finally we conclude by highlighting with what is considered to represent the main contribution of this work and the future work to follow.

II. RELATED WORK AND OUR FRAMEWORK

A study to deploying Semantic Web frameworks for the representation and analysis of social networks exists [7]. It exploits querying in SPARQL to compute global metrics (density, betweenness, centrality, etc.) of social networks in general. Another work [8] proposes the integration of two social networks, "knows" from FOAF documents and "co-author" from the DBLP bibliography [10], into a weighted "co-author" model to detect the conflict of interest among scientific researchers. To the best of our knowledge, there is no approach to date which bases solely on the Semantic Web, be it for the representation, or for analysis of co-authorship networks as a special class of social networks. Nor is there an approach of considering rules of Semantic Web to express and reason over rich "co-author" relations characterized by metrics like weights.

We present here a novel approach of (1) extending the semantics of the FOAF ontology and (2) deploying SWRL rules to represent and analyze co-authorship networks. We have therefore adopted a directed *weighted* graph model as the inner representation model of co-authorship networks as suggested in the literature [11] and not supported by generic social network models, as will become obvious in the next sections.

III. BACKGROUND

Let us next recall the definitions according to [11] of the directed weighted co-authorship graph model which we adopted for our framework to determine the weights and other core metrics of co-authorship links.

Let A = {a1,...,an} denote the set of n authors. Let the set of m publications be denoted as P = $\{p_1,...,p_k,...p_m\}$. Let f(pk) define the number of authors of publication pk. Then the following definitions are introduced:

Definition 1 (Exclusivity per publication). If authors a_i and a_j are co-authors in publication p_k,

$$g_{i,j,k} = 1/(f(p_k) - 1)$$

$g_{i,j,k}$ represents the degree to which authors a_i and a_j have an exclusive co-authorship relation for a particular publication. This definition gives more weight to co-author relationships in publications with fewer total co-authors than publications with large numbers of co-authors, i.e., it weighs the co-authorship relation in terms of how exclusive it is.

Definition 2 (Co-authorship frequency). Another important metric over a pair of authors a_i and a_j, the co-authorship frequency:

$$c_{ij} = \sum_{k=1}^{m} g_{i,j,k}$$

will sum up the exclusivity values $g_{i,j,k}$ for that same pair i,j across all publications k (k=1..m) where they appear as co-authors.

This gives more weight to authors who co-publish more publications together, and do so exclusively.

Definition 3 (Total co-authorship frequency).

$$c_i = \sum_{k=1}^{n} c_{ik}$$

consists of the sum of all co-authorship frequency values c_{ik} over a particular author ai and all of its co-authors a_k (k=1..n) in whatever publications where a_i appears as author.

Definition 4 (Normalized weight). In order to obtain a normalized value for the weight of the co-authorship relation between two authors, the following normalization step by taking into account the total co-authorship frequency of a given author when calculating the co-authorship frequency between that author and any other co-author of him/her should be carried out as well:

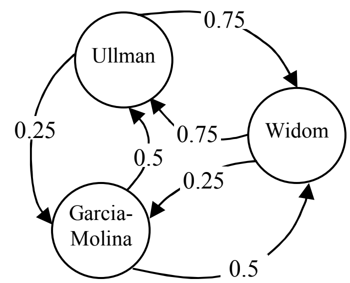
$$w_{ij} = \frac{c_{ij}}{c_i}$$

It ensures that the weights of an author's relationships sum to one.

Example 1. The running example we will use throughout this paper consists of two publications and their respective authors as follows:

Publication	Authors		
First Course in Database Systems	Ullman	Widom	
Database Systems: The Complete Book	Ullman	Widom	Garcia-Molina

The metrics calculated following the definitions given above are given in the figure below (Fig. 1).



Exclusivity per publication:

First Course in Database Systems:	1
Database Systems: The Complete Book	0.5

Total co-authorship frequency:

Ullman:	2
Widom:	2
Garcia-Molina:	1

	Co-authorship frequency:		Normalized weight:
Ullman	Widom:	1.5	1.5/2.0 = 0.75
Ullman	Garcia-Molina:	0.5	0.5/2.0 = 0.25
Widom	Ullman:	1.5	1.5/2.0 = 0.75
Widom	Garcia-Molina:	0.5	0.5/2.0 = 0.25
Garcia-Molina	Ullman:	0.5	0.5/1.0 = 0.5
Garcia-Molina	Widom:	0.5	0.5/1.0 = 0.5

Figure 1. Representation of the weighted directed co-authorship network of *Example 1*.

IV. EXTENDING THE FOAF TO SUPPORT CO-AUTHORSHIP NETWORKS

We next provide CO-AUTHORONTO, our extension of the FOAF ontology to capture the semantics of rich co-authorship networks, i.e., of directed *weighted* co-authorship networks, which allow augmenting relationships between authors with information such as weight, as well as other knowledge relevant to collaboration schemes in the scientific community as suggested in the literature [11].

Further we detail rule definitions in SWRL to infer values of such added information and knowledge, and assign them to the corresponding annotations of the underlying CO-AUTHORONTO ontology.

A. *CO-AUTHORONTO: A FOAF Extension to Model Co-Authorship Networks*

Friend of a Friend is a vocabulary which supports representing people, their profiles and social relationships among them on the Web. It is thus not focused in supporting any special community, but is flexible in extending it with special primitives aimed to capturing semantics of a given domain. A proposal [12], e.g., suggests the extension of the `foaf:knows` element in FOAF to describe the relationship between people in more detail, such as `parentOf`.

In the scientific community domain, there is a need to express more precise relationships in terms of the co-authorship than `foaf:knows` to relate people. We have extended the FOAF ontology into CO-AUTHORONTO with primitives which allow expressing co-authorship relations with additional attributes describing them such as the weight of a relationship. We have in addition used the AKT Reference Ontology [13], namely its two sub-ontologies, (1) the Portal ontology which describes people, projects, publications, geographical data, etc., and (2) the Support ontology providing basic definitions used by the portal ontology.

The integration thereby of these altogether three vocabularies, our CO-AUTHORONTO vocabulary, the AKT `portal:` and `support:` vocabularies, and the `foaf:` vocabulary (cf. Fig. 2) implied several `owl:equivalentClass` definitions as follows:

- our own introduced class `Publication` of the CO-AUTHORONTO ontology which is a subclass of the `foaf:Document` class, and the class `portal:Publication` of the AKT ontology:

```
:Publication   a   owl:Class;
   rdfs:subClassOf foaf:Document;
   owl:equivalentClass
portal:Publication.
```

- our own introduced class `Person` of the CO-AUTHORONTO ontology, and the classes `portal:Person` and `foaf:Person`:

```
:Person    a    owl:Class;
   owl:equivalentClass
      (portal:Person and foaf:Person).
```

Two main classes in our extended FOAF ontology are the class `Person` and the class `Publication` with their respective subclasses such as `Researcher`, or `Student` for the former, or, `Serial-Publication`, or `Book` for the later, all inherited readily either from the AKT ontologies, or the FOAF ontology. Some of the properties we introduced of the class `Publication` relevant to modeling our weighted co-authorship network are shown below (Fig. 3) along with the restrictions on them.

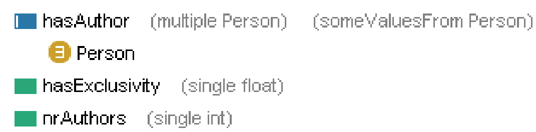


Figure 3. Some properties of the `Publication` class defined in our ontology.

Each `Publication` instance may have (multiple) values of type `Person` on the `hasAuthor` property. The `hasExclusivity` property implements the exclusivity degree on a given publication as given in Definition 1. All pairs of co-authors of the same publication should share a common value for the exclusivity. The property `nrAuthors` as its naming reveals, is designed to record data about the number of authors of a given publication. For both `hasExclusivity` and `nrAuthors` properties, their values will be inferred while reasoning on the ontology through SWRL rules as will be introduced in the next section.

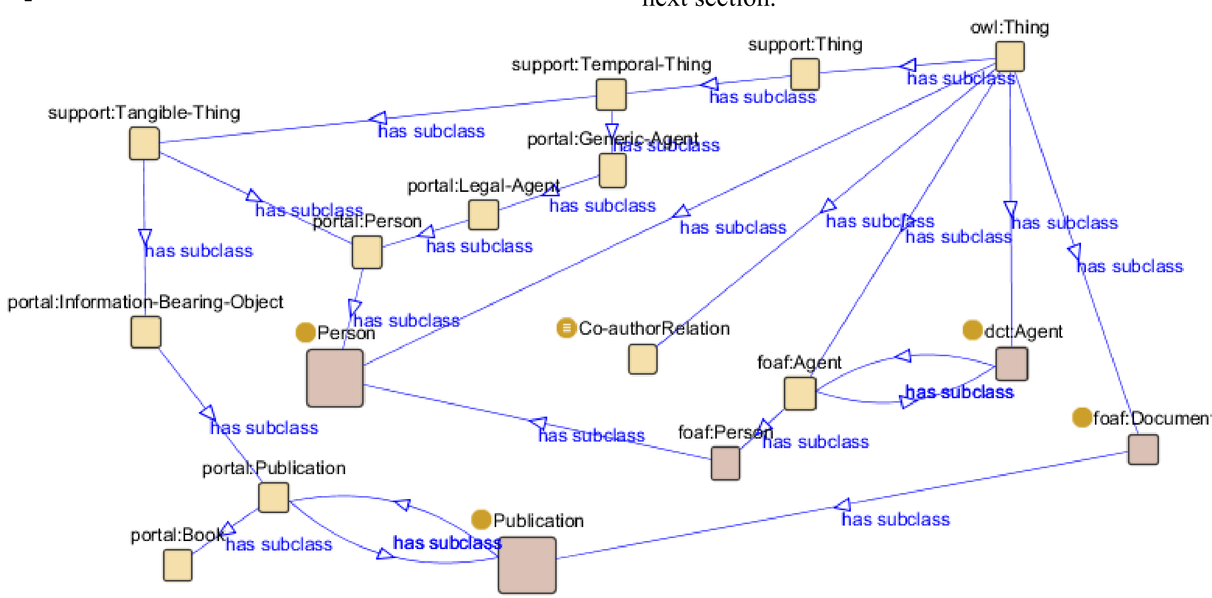


Figure 2. An excerpt of our ontology – how it is related to the `foaf`, `portal` and `support` (class hierarchy).

Representing the class `Person` on the other side requires more complex ontological primitives. The `foaf:knows` property is a binary relation which may link two individuals of type `Person`, but is however not convenient to use it to link an individual `Person` to another individual `Person` and more than just one value, as is the case with the co-author relation we aim to model. OWL [14] supports by default binary relations through properties, whereas for representing the co-author relation we need the so-called n-ary relations [15]. N-ary relations are expressive enough but cost an additional superfluous effort for dealing with them as will become obvious when considering rules in SWRL in the next section.

Fig. 4a illustrates the use of OWL n-ary relations (pattern 1) for representing the `hasCo_author` property of the `Person` class in out ontology. It links an individual `Person` to another individual `Person`, as well as to two other values, the *weight* and *frequency* which characterize such a `hasCo-author` relation.

The `SWRLInjected_2` (cf. Fig. 4b) acts somewhat as an abstract relation instance: It does not hold any information, but serves merely to link from the source instance (Ullman) to this relation instance (`SWRLInjected_2`), and from this relation instance (`SWRLInjected_2`) to the target instance (Widom) as well as to additional information about this relation instance (1.5 and 7.5).

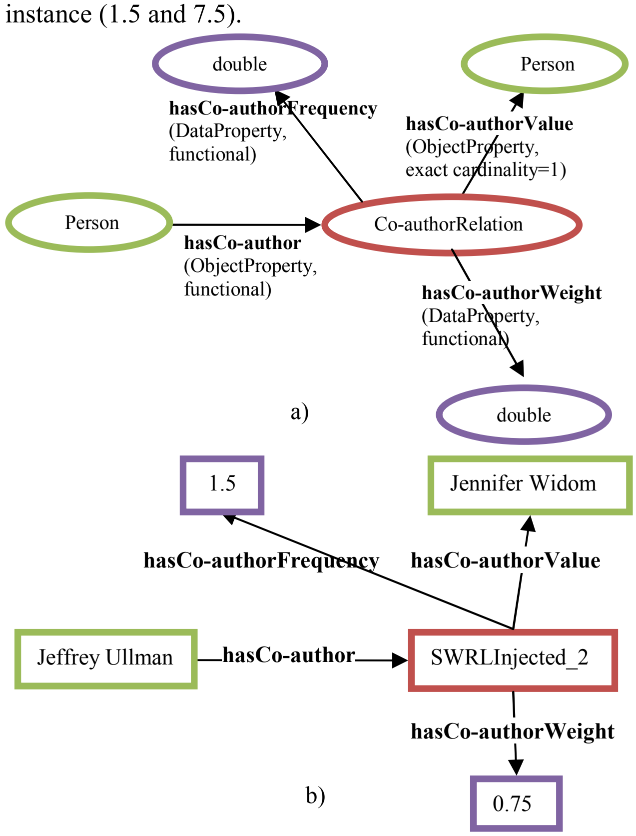


Figure 4. N-ary representation of the `hasCo-author` property in our ontology: a) class view, and b) instance view. *Example 1* excerpt: Ullman and Widom are co-authors with frequency 1.5 and weight 0.75.

In the definition of the `Person` class (of which the individual `Ullman` is an instance), we specify a property `hasCo-author` with the range restriction going to the `Co-authorRelation` class (of which `SWRLInjected_2` is an instance):

```
:Person   a  owl:Class;
    rdfs:subClassOf
    [a  owl:Restriction;
        owl:allValuesFrom:
                        Co-Author-Relation;
       owl:onProperty:    hasCo-author].
```

Here is a definition of the class Co-authorRelation in our ontology:

```
:Co-author-Relation   a  owl:Class;
  rdfs:subClassOf
  [a  owl:Restriction;
      owl:someValuesFrom: Person;
     owl:onProperty: hasCo-authorValue];
  rdfs:subClassOf
  [a  owl:Restriction;
      owl:allValuesFrom: double;
      owl:onProperty: hasCo-authorWeight];
  rdfs:subClassOf
  [a  owl:Restriction;
      owl:allValuesFrom: double;
      owl:onProperty:
                    hasCo-authorFrequency].
```

Besides the primitives we introduced, attributes which describe the type of the co-authorship relations (e.g., supervisor, colleague, same project, etc.), or the duration the co-authorship relation took place, are also easily attachable to our ontology.

B. *Rules to Construct Weighted Co-authorship Network*

If we move from ontologies upwards the Semantic Web layering architecture, we will encounter the logic layer with rules which may capture semantics that cannot be directly captured using description logic underlying OWL. Just recently, the Rule Interchange Language (RIF) [16] has been conceived as the standard rule language in the Semantic Web. We have nevertheless decided to deploy the SWRL for designing rules in our framework due to:

- its availability in form of development environments like is Protégé [17] we used here,
- its closeness to OWL as a representation model for knowledge to reason upon, as well as,
- a possibility to ever easily interchange it with RIF.

We have defined rules in SWRL to implement several metrics of the weighted co-authorship networks [11] as outlined in Section 3, i.e., exclusivity, co-authorship frequency, total frequency, and weights. Next we show each of the SWRL rules and the corresponding metrics of the weighted co-authorship network model they implement. Tests on each rule definition have also been performed by retrieving the results of each rule at query time (the SQWRL library), and will next be given as well accompanying the respective rule definition.

The values of these metrics once pre-calculated through an inference engine, they are assigned to the existing ontology annotations. These metrics are subject to change incrementally as the underlying graph structure will most probably grow over time.

Rule 1 (Co-author Relation)
Given ?a1 and ?a2 are authors of the same publication, the following SWRL rule will create an individual in the CO-AUTHORONTO ontology through the built-in predicate makeOWLThink of the swrlx library, and bind it to the Co-authorRelation class which shall relate those two authors of the given publication to indicate they are co-authors to each other.

```
Publication(?p) ∧ hasAuthor(?p,?a1) ∧
abox:hasURI(?a1,?a1URI) ∧
swrlb:substringAfter(?a1name,?a1URI,"#") ∧
hasAuthor(?p,?a2) ∧
abox:hasURI(?a2,?a2URI) ∧
swrlb:substringAfter(?a2name,?a2URI,"#") ∧
swrlb:notEqual(?a1name,?a2name) ∧
swrlx:makeOWLThing(?rel,?a1,?a2) → Co-
authorRelation(?rel) ∧ hasCo-
author(?a1,?rel) ∧ hasCo-
authorValue(?rel,?a2)
```

The above rule evaluates once for each individual of the Publication class asserted in our ontology, for each pair of its authors iteratively.

The notEqual built-in predicate does exclude a combination of an author with himself (A1, A1) as a possible co-author pair. The same could be achieved declaring the axiom differentFrom(?a1,?a2) given we have already declared an allDifferent axiom to hold over all instances of the Person class.

Testing Rule 1:

```
Publication(?p) ∧ hasAuthor(?p,?a1) ∧
abox:hasURI(?a1,?a1URI) ∧
swrlb:substringAfter(?a1name,?a1URI,"#") ∧
hasAuthor(?p,?a2) ∧
abox:hasURI(?a2,?a2URI) ∧
swrlb:substringAfter(?a2name,?a2URI,"#") ∧
swrlb:stringConcat(?rel,?a1name,"_rel_",?a
2name) ∧ swrlb:notEqual(?a1name,?a2name) →
sqwrl:select(?rel)
```

Rule 2 (Number of authors of a given publication)
Next the rule in SWRL to calculate the number of authors for each publication is presented.

```
Publication(?p) ∧ hasAuthor(?p,?a) ˚
sqwrl:makeSet(?s,?a) ∧
sqwrl:groupBy(?s,?p) ˚
sqwrl:size(?size,?s) → nrAuthors(?p,?size)
```

It will collect all authors of all existing publications into a common set ?s (the makeSet built-in predicate), group them by the publication they belong to (the groupBy built-in predicate), and then extract the number of elements in a group per each group created (the size built-in predicate). Numbers obtained will be asserted to the nrAuthors property of the corresponding publications.

Testing Rule 2:

```
Publication(?b) ∧ hasAuthor(?b,?a) →
sqwrl:select(?b) ∧ sqwrl:count(?a)
```

A similar test could be performed using the size built-in predicate instead.

Rule 3 (Exclusive co-authorship for a particular publication)
This rule computes the degree of the exclusivity of the co-authorship relation for a given publication as determined in Definition 1. It yields the same value for whatever pair of co-authors of the same publication.

```
Publication(?p) ∧ nrAuthors(?p,?nr) ∧
swrlb:greaterThanOrEqual(?nr,2) ∧
swrlm:eval(?e,"1/(nr-1)",?nr) →
hasExclusivity(?p,?e)
```

The greaterThanOrEqual built-in predicate takes care to omit calculating this metric for publications with a single author. Analogously to the previous rule, values obtained for the hasExclusivity property will be assigned attached to that particular publication in the ontology. The eval is just another built-in of the swrlm library of SWRL to evaluate the given mathematical formula.

Testing Rule 3:

```
Publication(?p) ∧ nrAuthors(?p,?nr) ∧
swrlm:eval(?e,"1/(nr-1)",?nr) →
sqwrl:select(?p,?e)
```

Rule 4 (Co-authorship frequency)
The rule which implements the co-authorship frequency (cf. Definition 2) over a pair of authors ?a1 and ?a2 is defined as:

```
Publication(?p) ∧ nrAuthors(?p,?nr) ∧
swrlb:greaterThanOrEqual(?nr, 2) ∧
hasAuthor(?p,?a1) ∧ hasAuthor(?p,?a2) ∧
hasCo-author(?a1,?rel) ∧ hasCo-
authorValue(?rel,?a2) ∧
abox:hasURI(?a1,?a1URI) ∧
abox:hasURI(?a2,?a2URI) ∧
swrlb:notEqual(?a1URI,?a2URI) ∧
sameAs(?a1,?a1) ∧ sameAs(?a2,?a2) ∧
hasExclusivity(?p,?e) ˚
sqwrl:makeSet(?s,?e) ∧
sqwrl:groupBy(?s,?a1,?a2) ˚
sqwrl:sum(?f,?s) → hasCo-
authorFrequency(?rel,?f)
```

Again as in Rule 2, the makeSet and groupBy predicates are applied, but this time targeted to collect exclusivity values, group them by a pair of authors, and finally calculate the sum (the sum built-in predicate) of exclusivity values that belong to that given group, i.e., pair of co-authors. The result will assert a value on the `hasCo-authorFrequency` property of the `Co-authorRelation` individual which connect the given co-author pairs `?a1` and `?a2` respectively, once per each such co-authors pair (i.e., group created).

Testing Rule 4:

```
Publication(?p) ∧ nrAuthors(?p,?nr) ∧
swrlb:greaterThanOrEqual(?nr,2) ∧
hasAuthor(?p,?a1) ∧ hasAuthor(?p,?a2) ∧
abox:hasURI(?a1,?a1URI) ∧
abox:hasURI(?a2,?a2URI) ∧
swrlb:notEqual(?a1URI,?a2URI) ∧
sameAs(?a1,?a1) ∧ sameAs(?a2,?a2) ∧
hasExclusivity(?p,?e) ˚
sqwrl:makeSet(?s,?e) ∧
sqwrl:groupBy(?s,?a1,?a2) ˚
sqwrl:sum(?f,?s) →
sqwrl:select(?a1,?a2,?f)
```

Rule 5 (Total co-authorship frequency)
The following rule implements the total co-authorship frequency as given in Definition 3. It reasons over values assigned to the `hasCo-authorFrequency` property of a particular co-author relationship `?rel` between `?a1` and `?a2`, uses the `makeBag` built-in predicate to collect them into a bag rather than a set (bags allow duplicate values, whereas sets do not), groups the set values by author `?a1`, and calculates the sum of values on each group separately, yielding thus the values for total co-authorship frequency of respective authors (bound to the respective author on his `hasTotalFrequency` property in our ontology).

```
Person(?a1) ∧ hasCo-author(?a1,?rel) ∧
hasCo-authorValue(?rel,?a2) ∧ hasCo-
authorFrequency(?rel,?f) ˚
sqwrl:makeBag(?s,?f) ∧
sqwrl:groupBy(?s,?a1) ˚
sqwrl:sum(?totalFa1,?s) →
hasTotalFrequency(?a1,?totalFa1)
```

Testing Rule 5:

```
Person(?a1) ∧ hasCo-author(?a1,?rel) ∧
hasCo-authorValue(?rel,?a2) ∧ hasCo-
authorFrequency(?rel,?f) ˚
sqwrl:makeSet(?s,?f) ∧
sqwrl:groupBy(?s,?a1) → sqwrl:select(?a1)
∧ sqwrl:sum(?f)
```

Rule 6 (Co-author weight)
Finally, the following rule will find the (normalized) weight of the co-author relationship between two authors (cf. Definition 4) by dividing (the `eval` predicate) the co-authorship frequency of a given author to that particular co-author with the total co-authorship frequency of a given author. This rule will iterate over all co-author pairs found in the ontology (the `?rel` individuals of the `Co-authorRelation` class), and accordingly augment values on their `hasCo-author` property which represent weights, i.e., the `hasCo-authorWeight` property.

```
hasCo-author(?a1,?rel) ∧ hasCo-
authorValue(?rel,?a2) ∧
hasTotalFrequency(?a1,?totalFa1) ∧ hasCo-
authorFrequency(?rel,?a12f) ∧
swrlm:eval(?w,"a12f/totalFa1",?a12f,?total
Fa1) → hasCo-authorWeight(?rel,?w)
```

Testing Rule 6:

```
hasCo-author(?a1,?rel) ∧ hasCo-
authorValue(?rel,?a2) ∧
hasTotalFrequency(?a1,?totalFa1) ∧ hasCo-
authorFrequency(?rel,?a12f) ∧
swrlm:eval(?w,"a12f/totalFa1",?a12f,?total
Fa1) → sqwrl:select(?rel,?w)
```

The set of rules we presented (see Fig. 5 for a system view) constitute the core of our weighted directed co-authorship network model. They may be extended to support more metrics due to the modularity of our framework. Every rule in SWRL is atomic, and extending the set of rules in our framework will not harm its initial model.

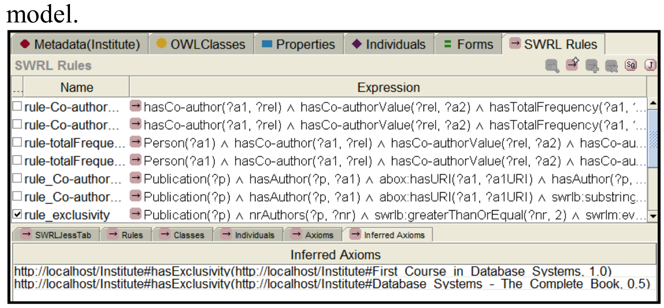


Figure 5. Evaluation of SWRL rules in our system. The Rule 3 (*exclusivity*) result against Example 1.

V. EVALUATION

After having tested our approach on a small sample running example as demonstrated in the preceding section, we also evaluated our rules against a huge real-world co-authorship network. The data set for this experiment was extracted from the library database of the South East European University (SEEU) storing the bibliography of books and other publications available for loan from the university. It contains 12774 `Publication` individuals, and 11607 `Person` individuals acting as authors of publications. The experiment was run on a 2 x Quad Core 2.5 GHz machine with 16 GB RAM and Windows Server 2003 2R installed.

The SWRL rules in our framework did infer that there are 12732 co-author relationships (the total number of `hasCo-authorValue` properties in our ontology) as a special type of the FOAF `knows` relationship. Next we present some of the results of our benchmarks related to the metrics being calculated through our framework of SWRL rules over the FOAF representation of the co-authorship network of the SEEU library test-bed of data.

Query 1 (Total co-authorship frequency)
The SQWRL query expression below is used to find the total co-authorship frequency values of authors in the SEEU library as defined by Rule 5 earlier in our system.

```
hasTotalFrequency(?a1,?tFa1)∧
hasTotalNrCo-Authors(?a1,? tNrCa1)∧
portal:full-name(?a1,?n1) →
sqwrl:columnNames("author","name","nr co-
authors","total co-authorship frequency")
∧ sqwrl:select(?a1,?n1,?tNrCa1,?tFa1) ∧
sqwrl:orderByDescending(?tFa1)
```

As the result of this query, we yield a table of cardinality 7008 with top 10 ranked authors on their total co-authorship frequency values (cf. Rule 5 in the previous section for their calculation) as shown in Tab. 1.

As one may also reason from the list in Tab. 1, there are cases when an author has more co-authors (might share diverse roles, like writer, editor, etc.) but is nevertheless ranked lower in the table: The author Rex Gibson who's total number of co-authors is equal to 11 is ranked higher in the table (position 8) than is Erion Kristo (position 10) who's total number of co-authors is equal to 12. This since the total co-authorship frequency measure takes also into account how exclusive the given co-author relations are, and assigns higher scores to more exclusive co-author relations. Thus, a sum of less but more exclusive co-author relations may outperform a higher in number but weaker in exclusivity co-author relations.

TABLE 1. TOP 10 RANKED AUTHORS IN DESCENDING ORDER.

nr	author	name	nr co-authors	total co-authorship frequency
1	http://localhost/Institute#P21542	William Shakespeare	24	18.5
2	http://localhost/Institute#P22253	Clare West	19	18
3	http://localhost/Institute#P21104	Ismail Kadare	17	14.5
4	http://localhost/Institute#P22263	Diane Mowat	14	12
5	http://localhost/Institute#P22463	Ali Aliu	12	12
6	http://localhost/Institute#P22235	John Escott	12	11
7	http://localhost/Institute#P22997	Honoré de Balzac	12	10
8	http://localhost/Institute#P23001	Rex Gibson	11	9.5
9	http://localhost/Institute#P22611	Victor Hugo	12	9
10	http://localhost/Institute#P31020	Erion Kristo	12	9

The chart in Fig. 7 depicts total co-authorship frequency values of all authors. The dominant values are 1, then 2, 3, etc., to then end up with the highest ranked value of the total co-authorship frequency, i.e., 18.5 in our example as also provided in Tab. 1.

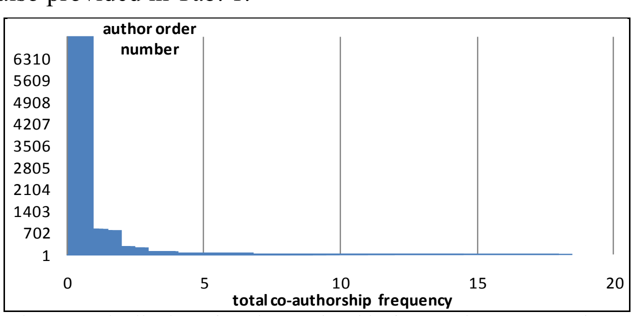


Figure 7. Distribution of total co-authorship frequencies among authors.

Query 2 (Co-author weight)
Our next query in SQWRL lists all co-author pairs for books in the SEEU library and their weights as defined by Rule 6 earlier in our system.

```
hasCo-author(?a1,?rel) ∧ hasCo-
authorValue(?rel,?a2) s∧
hasTotalFrequency(?a1,?totalFa1) ∧ hasCo-
authorFrequency(?rel,?a12f) ∧ hasCo-
authorWeight(?rel,?w) ∧ portal:full-
name(?a1,?n1) ∧  portal:full-name(?a2,?n2)
→ sqwrl:columnNames("author","co-author",
"co-authorship frequency", "weight") ∧
sqwrl:select(?n1,?n2,?a12f,?w)
```

The result, just a small fragment of a total of 12729 rows of co-author pairs and their weigh values along with the co-authorship frequency values are depicted in Tab.2.

The co-author relationship between authors Hoti and Kadare is of weight 1 (see row 1 in Tab. 2) since they share a single common publication, and Hoti owns no other publications (its total frequency is equal to 1, i.e., 1/1 = 1). Note that the weight metric is not commutative, i.e., the same co-author relation but in the opposite direction might be of a different weight due to different total co-authorship frequencies of each of the authors for that same co-author relation. In our example, the opposite Kadare to Hoti co-author relation (row 2 in the table) yields a different weight value: Kadare's total co-authorship frequency is 14.5, thus resulting into a weight equal to 1/14.5 = 0.068965517.

TABLE 2. CO-AUTHOR PAIRS AND THEIR WEIGHTS. AN EXCERPT OF THE QUERY 2 RESULT.

author	co-author	co-authorship frequency	weight
Ukshin Hoti	Ismail Kadare	1	1
Ismail Kadare	Ukshin Hoti	1	0.068965517
Brian W Kernighan	Rob Pike	1	0.333333333
Brian W Kernighan	Dennis M Ritchie	1	0.333333333
Brian W Kernighan	Scott E Gimpel	0.5	0.166666667
Brian W Kernighan	Clovis L Tondo	0.5	0.166666667

Further in Tab. 2 (rows 3 to 6), the relation of the author Kernighan to its co-authors, Gimpel and Tondo, is of half less weight than is the relation to its co-authors, Pike and Ritchie. As is obvious from Tab. 3:

- Kernighan, Gimpel and Tondo share a common publication, and thus own the same frequency values to each other, i.e. 0.5, whereas their respective co-authorship weights are 0.16.
- The pairs Kernighan and Pike, and Kernighan and Ritchie are both co-author pairs of their respective publications, and thus share frequency values of 1. This turned into weights (1/3) yields the value 0.33.

TABLE 3. KERNIGHAN'S CO-AUTHORS AND THE PUBLICATIONS IT SHARES.

author	co-author	publication title
Brian W Kernighan	Scott E Gimpel	The C answer book
Brian W Kernighan	Clovis L Tondo	The C answer book
Brian W Kernighan	Rob Pike	The practice of programming
Brian W Kernighan	Dennis M Ritchie	The C programming language

The distribution of weight values for the whole set of co-author pairs in the SEEU library is shown in Fig. 8. The maximum weight in accordance also with the formulae in Definition 4 (cf. Section 3) is 1, and is assigned to more than 3500 co-author pairs. Approximately the same distribution (ca 3500 pairs) applies for the weight 0.5. The minimum weight found in our dataset equals 0.021.

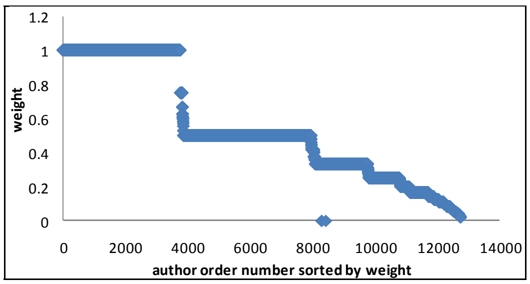


Figure 8. Weight distribution. The complete Query 2 result.

Further in the context of the evaluation of our framework, the SWRL query below proved that for every author in our library dataset, the weights of that author's relationships really sum to one, as earlier stated by Definition 4.

```
hasCo-author(?a1,?rel) ∧ hasCo-
authorWeight(?rel,?w)˚sqwrl:makeBag(?s,?w)
∧ sqwrl:groupBy(?s,?a1)˚sqwrl:sum(?Wa1,?s)
→ sqwrl:select(?a1,?Wa1)
```

VI. CONCLUSION AND FUTURE WORK

We introduced here CO-AUTHORONTO, an extension of the FOAF ontology to allow modeling sophisticated co-authorship networks as they appear on the Web, including metrics like weight, exclusivity, or frequency, as suggested in the literature. Further, our novelty lies on deploying rules in Semantic Web to infer new information and knowledge which are not a matter of assigning them manually, but are rather calculated and then assigned to the corresponding annotations of the underlying CO-AUTHORONTO ontology. Currently, SWRL is the paradigm we use to exploit logic in the Semantic Web which can be easily replaced with any other rule language of the logic layer of Semantic Web.

It makes it thus easy for the FOAF co-authorship network to amalgamate it with native SWRL co-authorship rules for providing a *uniform* weighted co-authorship network based solely on the Semantic Web.

The importance of co-authorship networks is obvious in today's digitalized world of publication archives and their impact on our everyday life. Their amalgamation into the existing tantalized collection of FOAF data on the Web will increase their availability and workaround with diverse statistics on metrics we presented.

Further, our framework is easily extensible to support social network analysis in general, and this is where our plans for future work aim to focus. Our model already allows expressing weighted `foaf:knows` relationships among people in social networks. It is just the calculation of these weights which might differ and shall be flexible for distinct problem domains. Rules in SWRL or RIF provide a future platform for expressing calculations, or what is considered as logic in the Semantic Web, and this is where our framework is mainly built upon. Expressing more metrics through rules is among the plans we are working to improve the usage of social network data on the Web.

The central idea is designed and implemented as part of the work presented here. We have shown, a comprehensive modeling and interesting analyses of co-authorships are possible with our novel approach which embraces well-established theories in the field of weighted co-authorship networks, as well as on-going trends in the Web of Data.

REFERENCES

[1] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American , 284* (5), 34--43.

[2] Brickley, D., & Miller, L. (2010, January 1). *FOAF Vocabulary Specification 0.97.* Retrieved June 2010, 28, from http://xmlns.com/foaf/spec/20100101.html

[3] Goldbeck, J. (2007). The dynamics of Web-based social networks: Membership, relationships, and change. *First Monday , 12* (11).

[4] Goldbeck, J. (2005). *Web-based Social Networks: A Survey and Future Directions.* Technical Report, University of Maryland.

[5] De Castro, R., & Grossman, J. W. (1999). Famous trails to Paul Erdös. *MATHINT: The Mathematical Intelligencer , 21*, 51--63.

[6] Nascimento, M. A., Sander, J., & Pound, J. (2003). Analysis of SIGMOD's co-authorship graph. *SIGMOD Rec. , 32* (3), 8--10.

[7] Erétéo, G., Buffa, M., Gandon, F., & Corby, O. (2009). Analysis of a Real Online Social Network using Semantic Web Frameworks. *8th International Semantic Web Conference, ISWC'09.* Wiley Publishing, Inc.

[8] Aleman-Meza, B., Nagarajan, M., Ding, L., Sheth, A., Arpinar, B. I., Joshi, A., et al. (2008). Scalable semantic analytics on social networks for addressing the problem of conflict of interest detection. *ACM Trans. Web , 2* (1), 1--29.

[9] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004, May 21). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved June 27, 2010, from http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[10] Michael, M. (2002). The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. *String Processing and Information Retrieval. 2476*, pp. 1-10. Springer LNCS.

[11] Liu, X., Bollen, J., Nelson, M. L., & Van de Sompel, H. (2005). Co-authorship networks in the digital library research community. *Inf. Process. Manage. , 41* (6), 1462--1480.

[12] Davis, I., & Vitiello, E. J. (2010, April 19). *RELATIONSHIP: A vocabulary for describing relationships between people.* Retrieved July 2011, 29, from http://vocab.org/relationship/.html

[13] *The AKT Reference Ontology*. (2003). Retrieved June 23, 2011, from (AKT), Advanced Knowledge Technologies: http://www.aktors.org/publications/ontology/

[14] McGuinness, D. L., & van Harmelen, F. (2004, February 10). *OWL Web Ontology Language Overview.* Retrieved June 27, 2010, from http://www.w3.org/TR/2004/REC-owl-features-20040210/

[15] Noy, N., & Rector, A. (2006, April 12). *Defining N-ary Relations on the Semantic Web.* Retrieved June 27, 2010, from http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/

[16] Kifer, M., & Boley, H. (2010, June 22). *RIF Overview.* Retrieved June 27, 2010, from http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/

[17] Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. *3rd International Semantic Web Conference, ISWC'04. 3298*, pp. 229-243. Springer LNCS.