

Presenting and Preserving the Change in Taxonomic Knowledge for Linked Data

Rathachai Chawuthai^{a,b,*}, Hideaki Takeda^b, Vilas Wuwongse^c, and Utsugi Jinbo^d

^a *SOKENDAI (The Graduate University for Advanced Studies), Kanagawa, Japan*

^b *National Institute of Informatics, Tokyo, Japan*

^c *Asian University, Chonburi, Thailand*

^d *National Museum of Nature and Science, Tokyo, Japan*

Abstract. Taxonomic knowledge provides scientific names to each living organisms and thus is one of the indispensable information to understand the biodiversity. On the other hand, the various perspectives of classifying organisms and the changes in taxonomic knowledge have led to the inconsistent classification information among different databases and repositories. To have the precise understanding of taxonomy, one needs to perform the integration of relevance data across taxonomic databases. This is difficult to establish due to the ambiguity in taxon interpretation. Most of the research in earlier stage employed the Linked Open Data (LOD) technique to establish the link in the taxonomy transition. However, they overlooked the temporal representation of taxa and underlying knowledge of the change in taxonomy, and it is difficult for users to gain perspective on how some identifiers of taxa are linked. To this end, this research aims at developing a model for presenting and preserving the change in taxonomic knowledge in the Resource Description Framework (RDF). Specifically, the proposed model takes advantage of linking some Internet resources representing taxa, presenting historical information of taxa and preserving background knowledge of the change in taxonomic knowledge in order to have the better understanding of organisms. We implement a prototype to demonstrate the feasibility and the performance of our approach. The results show that the proposed model is able to handle various practical cases of changes in taxonomic works and provides open and accurate access to linked data for biodiversity.

Keywords: Biodiversity Informatics, Change in Taxonomy, Knowledge Representation, Logical Model, Linked Data, Ontology, RDF, Scientific Name, Semantic Web, Taxonomic Data.

1. Introduction

Knowledge about biodiversity has been written in everywhere throughout the world. Researchers need to exchange knowledge about biodiversity across communities, so the link between communities' knowledge becomes a challenge issue, and intermediary is required. Nomenclature or a system of names is originally introduced for giving a unique and stable name, also known as a scientific name, as an identity of every Earth's organism [1,2]. Ideally, the scientific name should be unique and be a medium of linked data, however, in fact, there is the confusion in unambiguous of taxonomic notations. It leads to the change in taxonomic knowledge that becomes a serious problem. It can be a change in taxa or nomenclatures [1,2,5,23].

It therefore results in imprecise linked knowledge, and it makes the incorrect understanding of a single taxon.

In the present age, fortunately, the Internet and semantic web technologies provide a rich platform for linking data [7]. The idea of data interoperability enables a way to exchange data among different information systems. Information systems can be developed based on their own requirements and own data structures. When they publish their own datasets, they should consider the ability to link, to recognize by human, and to be consistent with standards. Moreover, local vocabularies used by an individual system should be reused from or associated to existing ones in order to build effective linked data [7].

Therefore, developing a taxonomic information system that places importance on both knowledge

*Corresponding author. E-mail: {rathachai}@nii.ac.jp.

management and linked data conduces to the better understanding of taxonomic knowledge. For this reason, our research aims to introduce a logical model for linking taxonomic knowledge based on the following objectives.

- To preserve the change in taxonomic knowledge
- To present and publish taxonomic knowledge as linked data

To accomplish these objectives, we have considered utilizing the idea of the Contextual Knowledge for Archives (CKA) approach [8] and the Meta-Ontology of Biological Name (TaxMeOn) [6] to capture the changes in taxonomic data and their context. We have, moreover, reused some taxonomic terms from Linked Open Data for ACademia (LODAC) [15], employed Simple Knowledge Organization System (SKOS) [29] vocabulary to manage the relationships between concepts, and publicized data to the Linked Open Data (LOD) Cloud [30]. In addition, we have implemented a prototype to prove the feasibility of our proposed model. Finally, we evaluated this work against the real cases of changes in taxonomy of moths under the family Saturniidae [39-41].

Next, we give background for our research in Section 2. We introduce our approach and fundamental concepts in Section 3. The prototype is presented in Section 4. The approach is evaluated in Section 5. Then we discuss the outcome of our work in Section 6. Lastly, we draw conclusions and suggest some future improvements in Section 7.

2. Background

In order to analyze the change in taxonomic knowledge and provide the basis of our model, we here review some relevance researches, then describe some details about the change in taxonomic knowledge, online databases, and linked data.

2.1. Change in Taxonomic Knowledge and Consequent Impact

A large number of species throughout the world have been described and classified with appropriate naming according to their characteristics such as morphological characters, living behaviors, and DNA sequences [1,2]. Many taxonomists have dedicated themselves to studying living organisms, and their knowledge has been published for more than hundred years. However, this knowledge has not always been shared among all

researchers around the world. In addition, there is no consensus on classification systems among all taxonomists. In other words, taxonomists might have different perspectives when it comes to classifying and naming living organisms. As a consequence, a single species is often classified and named differently [2]. To describe this situation more clearly, this part demonstrates some cases of the change in taxonomic knowledge.

The first example shows that one organism may have different names. If we take the Chinese yellow swallowtail named *Papilio xuthus* Linnaeus, 1767 as an example, we see that taxonomists at different research institutes have given this species different names, such as *xuthulus* Bremer, 1861; *chinensis* Neuburger, 1900; *koxinga* Fruhstorfer, 1908; and *neoxuthus* Fruhstorfer, 1908 [2].

Second, when two or more taxa were recognized as the same thing, only one name became accepted [45]. Thus, some species have to be reclassified and renamed due to the naming system [1,2]. For example, Hoare (2008) established the genus *Kendrickia* (ostracods). Then Kempf (2010) found that this genus was a primary junior homonym for *Kendrickia* Solem, 1985 (gastropods), and proposed the name *Dickhoarea* as a replacement name for the *Kendrickia* Hoare, 2008. This led to the subsequent change in species names; for instance, *Kendrickia asketos* has subsequently been renamed as *Dickhoarea asketos* since Kempf announced the name in 2010 [2].

Next, the progress of taxonomic studies frequently causes the redefinition of taxon concepts, i.e., the circumscription of the taxon [2]. Sometimes, it results in the change in species name. For example, the genus *Columba* (pigeons) has been split into five genera: *Patagioenas*, *Chloroenas*, *Lepidoenas*, *Oenoenas*, and *Columba*, where the latter *Columba* is narrower than the former one. Some species of the genus *Columba* have been assigned to one of these newly separated genera, for instance, *Columba speciosa* was changed to *Patagioenas speciosa* [12].

Another situation is to merge taxa such as genus level. When some genera were decided to be merged into a single taxon, their lower taxa such as species had to be transferred to the newly accepted genus [45]. According to nomenclature, these species had to be renamed to be consistent with the new genus name [1,2]. For instance, two genera of owls, *Bubo* and *Nyctea*, were merged into the prior genus *Bubo*. Following the change in these genera, the scientific name of the snowy owl *Nyctea scandiaca* has been subsequently changed to *Bubo scandiacus* in order to satisfy the zoological nomenclature [3].

Moreover, some researchers may have incorrect understanding of some taxon concepts as a result of them having been reclassified frequently, for example, a reclassification of the Baltimore oriole (*Icterus galbula* Linnaeus, 1758) and the Bullock's oriole (*I. bullockii* Swainson, 1827). In 1964, Sibley and Short argued that these two species should be merged into a single species [13]. As a result, the former name, *I. galbula*, became the accepted name, whereas *I. bullockii* was a junior synonym of *I. galbula*. By contrast, in 1995, research results regarding the DNA sequences of the two species led to the splitting of *I. galbula* into *I. galbula* and *I. bullockii* again [14]. Although these two species are currently separate, some information on *I. galbula*, especially that recorded between 1964 and 1995, might include some important details on *I. bullockii*. Researchers sometimes obtain imprecise information when they simply search for information using the name *I. galbula* only.

In these studies, we regarded that the change in taxonomic knowledge can be change in name and change in classification [1,2,6,11]. The example cases demonstrated some problems when each name reflect particular details observed by each researcher. Due to such a change in taxonomic names, when one who studies species data accesses just the information containing only the present scientific name, she or he sometimes misses important information that was recorded with its former scientific names. It means that the scientific names and taxonomy are lacked a single interpretation in biology [4,5]. Thus, to understand taxonomy thoroughly, we therefore need to know all of synonyms across multiple datasets, and then link their associated information together via the Internet [7]. Learning taxonomy with a single name may not be enough. To learn the precise knowledge of taxonomy, researchers have to pay attention to the significant of the change in taxa over time. Finding associations among background knowledge of the changes is also needed to study in order to understand the taxonomic knowledge more correctly.

2.2. The Informatics on Taxonomic Databases

In light of the issue previously mentioned, this study is an attempt to address the problem of incorrect interpretation of taxonomic data. An approach to linking taxonomic data along with precise context and the preservation of their background information is clearly needed.

Therefore, this section reviews several researches to be idea for solving this issue. A poor data model leads

to the lacking of likability among different datasets [19]. A scientific name alone is not enough for introducing a precise link [5,16-23]. The International Organisation for Plant Information (IOPI) model [16] used taxonomic name together with circumscription references as potential taxa for linking data among multiple taxonomic views. Biodiversity Information Standard (TDWG) [28,42] developed a standard for taxonomic data sharing among different datasets, adopted Life Science Identifiers (LSIDs) as Globally Unique Identifiers (GUIDs) for indexing taxa, and allowed having versions of taxon concepts. It also provided Darwin Core schema [31] containing vocabularies for describing taxonomic data. Page [17] and Jones [18] employed LSIDs for taxonomic databases, and the links of LSIDs can associate information among various data sources. The Universal Biological Indexer and Organizer (uBio) also gave LSIDs to taxa for enhancing the power of federated search engines [19]. As every taxon has been indexed with an ID, relations between taxa can be given using links between IDs [21]. Schulz [20] embedded taxonomy of living things into an ontology using Semantic technology. The hierarchy of taxon concepts was represented in Resource Description Framework (RDF) [20,42]. However, these researches have not yet mentioned about the preservation of changes in taxonomic knowledge. For this reason, TaxMeOn [6] developed a semantic web-based meta-ontology of biological name that managed and presented the changes in the scientific preposition of biological names and taxonomies such as splitting and lumping, and emphasized how the biological names were published by referring to some related publications. However, to the best of our knowledge, there is less discussion about the information structure of associations between any reasons behind the changes or background knowledge, which is needed to make a clear understanding of the taxonomic knowledge. This challenge puts forward the view that an underlying knowledge of the changes in taxonomic knowledge is required for the correct interpretation of taxonomic data. The study of biodiversity informatics should focus on the inclusion of the historical changes in taxa and context information that is essential to understand the situation regarding their changes and how names are related as well.

2.3. Taxonomic Knowledge and Linked Data

To materialize the conception of linked data, in this part, we reviewed the idea about an identifier of an Internet resource for describing a taxon. The use of

LSIDs as GUIDs promoted by TDWG [28,42] resulted in that the taxonomic data became globally available and linked. Several information models adopted the LSID as a unique key representing a taxon in their databases [16-18]. Jones [18] resolved the multiple names by assigning separated LSIDs for a name (NAMELSID) and for a taxon (TAXONLSID), and integrated the LSID into the Uniform Resource Identifier (URI). In addition, the authors of [22] compared the differences between the LSID and the URI, and recommended using a URI as a resource of taxonomic data in order to gain benefit from Linked Data approach. TaxMeOn [6] also put forward the view that the taxon concepts were always changed, so a fixed identifier might not proper for every concept. Therefore, when a taxon's circumscription was changed, that concept needed to be recognized as a new identifier. For instance, the genus *Bubo*, before merging with the genus *Nyctea*, must not have the same semantic web-based identifier as the *Bubo* after merging, because the latter *Bubo* is broader than the former one [2,3]. The model also allowed having a URI for a taxon concept and a URI for its name. It therefore had minimal redundancy and it was flexible for updating either names or concepts. Nevertheless, TaxMeOn propounded the view that a taxon concept and its name were treated as one unit in a name collection. The domain or the range of some properties is allowed to be a union of the scientific name and the taxon concept. Patterson [23] additionally introduced Global Names Architecture (GNA), and supported the view that names were keys to access biological information. GNA, which is mainly treat names with implicit taxon concepts, has three layers, but two layers are related to this topic. One is Global Names Index (GNI) that aims to collect name strings used in various information source and normalized spellings. Another one is Global Names Usage Bank (GNUB). It aims to describe name uses, which is a combination of a name and a reference, and nomenclatural issues. This name-centric model also provided features for identifying relationships between names, and it was integrated into some online official repositories of names such as ZooBank [43] and MycoBank [44]. The authors of [22] argued that it was very challenge to combine a name and a taxon concept into a single unit because it decreased the granularity of information but gave high simplicity. In addition, naming conventions for identifiers are different among different systems. Global Biodiversity Information Facility (GBIF), which is an international organization aiming to construct an information infrastructure to share information on biodiversity globally, gave reference guide for GNA. It is a

guide for an information system to select valid accept names among all names used for living being, recommended to use an unfriendly label for a persistent identifier because a taxonomic name is not stable [25,26,27,46]. The authors of [22] used non-human-readable local names in URIs. TaxMeOn [6] used human-readable URIs for taxonomic checklists and local identifiers. Furthermore, LODAC [15], which provided a linked data hub for biodiversity, denoted a URI as an Internet resource for representing a piece of taxonomic data. LODAC also considered to include a human-readable label in URI in order to make the model be lightweight, and human-friendly such as *lodac:Bubo*. It is consistent with the URIs of Internet resources used by DBpedia [24]. In this case, the human-readable URI is sometimes viewed as either a name or a taxon concept depended on context. It also gives advantage to humans especially biologists who involve with linked data, because the human-readable URI reduces gab between machines and normal users.

3. A Logical Model for Linking Taxonomic Knowledge

Here we present a logical model named Linked Taxonomic Knowledge (LTK) for preserving and presenting the change in taxonomic knowledge for linked data. To achieve the goals and issues addressed in the previous sections, our logical model is developed based on the following points.

- The model is possible to manage the change in taxonomic knowledge.
 - The model preserves the change as an event along with aspects of time and provenance.
 - The model supports the change in either taxa or association between taxa.
 - The model allows tracing the background knowledge of the changes by linking the cause and the effect between them.
- The model is possible to publish a suitable format of dataset for linked open data.
 - The linked data model deals with simple identifiers of semantic web resources in order to make the linked data be easy to recognize by both human and machine.
 - The model provides a sequence of changes in taxa.

- The model presents temporal data based on a given time point.

In this section, we illustrate the types of changes in taxonomic knowledge, some terms and descriptions, a formal definition of LTK, a use of data model, a description of the rules, and a method to utilize our approach in the Resource Description Framework (RDF). This section usually uses shorthand aliases for URIs, so their namespaces can be referred at Appendix A.

3.1. A Structure to Represent the Change in Taxonomic Knowledge

In this part, we studied how to classify a change in a taxonomic knowledge. On the basis of the changes in taxonomic knowledge that are analyzed from actual use cases [2,3,6,11,13,14,39,40,41], we summarize the practical cases in to Figure 1. The figure demonstrated that there are three main categories: changes in nomenclature, taxon concept, and relationship. First, the category nomenclature refers to the change in name containing rename, synonym, and homonym. Synonym is used when different names are assigned for the same taxon, whereas homonym is used when

the same name is assigned for the different taxa. Second, change in taxon concept denotes the change in description of a taxon. It includes the life span of the use of taxa that are initially stated (creating) and obsolete (ending), and the replacement of taxa in checklists. It also includes the change in scopes of taxa, which are merging, splitting, and change in circumscription. Merging is to lump some taxa into a single taxon, splitting is to separate a taxon into several taxa, and change in circumscription is to modify scope of a single taxon. In this case, the taxa before the change are assumed to be obsolete from the dataset, and then the other taxa after the change become newly created. Last, the change in relationship is to modify a link between concepts. In term of semantic web, it is a change in a triple. In this figure, three changes are mentioned. The change in higher taxon moves a lower taxon from a higher taxon to another one. The subdividing taxon is to create new sub taxa under the given higher taxon. It differs from splitting, because the given higher taxon remains accepted and its description does not change. For example, a species *Aus aus* was subdivided into subspecies *A. aus aus* and *A. aus bus*. Conversely, the combining taxa is to obsolete all lower

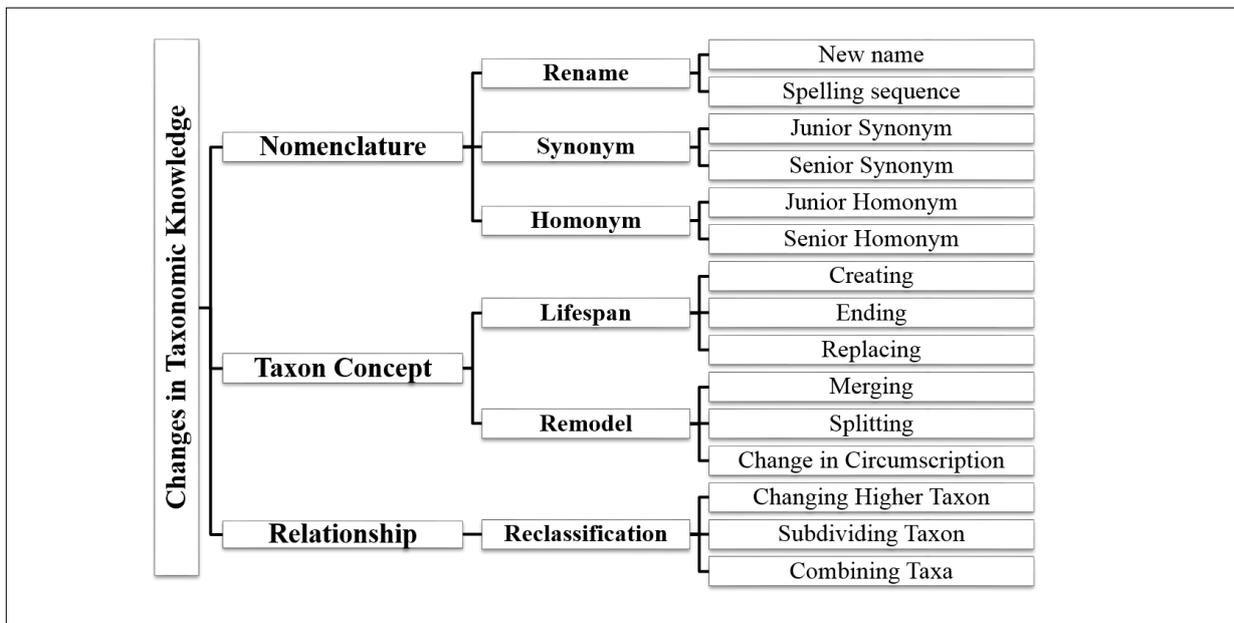


Fig. 1. Analysis of changes in taxonomic knowledge

taxa of a given higher taxon. For example, the subspecies *A. aus aus* and *A. aus bus* were combined to the species *A. aus*, so both subspecies were no longer used.

3.2. Preliminary Definitions

The way to describe the changes in taxonomy along with context knowledge is a challenging task. In this research, we primarily employed the CKA approach, which offers a logical model for presenting the change in Underlying Community Knowledge based on the theory of Flouris [9]. CKA contains data model for an event that assures some entities of changes, and binds a time interval and some references. The entity of change or the operation of change captures the change in conception such as splitting and merging, and the change in association between concepts such as changing membership. It also provides ideas for transforming the event into timeline and temporal data, which basically respond to the requirements of digital archives. However, for the field of biodiversity informatics, we have to enhance CKA approach with the specific requirements in biodiversity informatics and introduce some terms used by our research.

3.2.1. Entities for LTK

An entity in LTK is a URI for responding some specific positions; for example entities for representing taxa, operations of changes, and events describing the changes. In this case, some terms are needed to be defined and clarified.

Nomial Entity

Semantic technology encourages that everything should be represented as an Internet resource identified by a URI [7]. In this research, a nomial entity is a concept and an Internet resource used for taxonomic knowledge and it includes taxon concepts and names.

Simple Nomial Entity

Due to the change in knowledge, the role of a taxon has lifespan. In this research, when taxonomy is accepted in a given timeframe, it is considered as a taxon concept, otherwise it is viewed as a name. In this case, a simple nomial entity, a subset of nomial entity, is introduced to be an Internet resource that can act as either a taxon concept or a name. Due to an advantage from DBpedia [24] and LODAC [8], a human-readable URI makes RDF statements be human friendly in linked data graph, for example *dbpedia:Bubo* and *lodac:Bubo*. This research recommends using simple nomial entity for several reasons: (1) a model is simple and lightweight, (2) data is easy to recognize by normal users, and (3) a triple in linked data is more understandable. In addition, the issue of homonym can be solved by using a different namespace.

Contextual Nomial Entity

The change in knowledge sometimes gives impact to some representative taxa, and then their circumscription or their name may be changed. Our work deals with this problem by applying the idea of TaxMeOn [6] that created different URIs for the same taxon when its description is changed. We additionally define that every representation of taxonomy used in LTK is viewed as a version of a nomial entity. In case

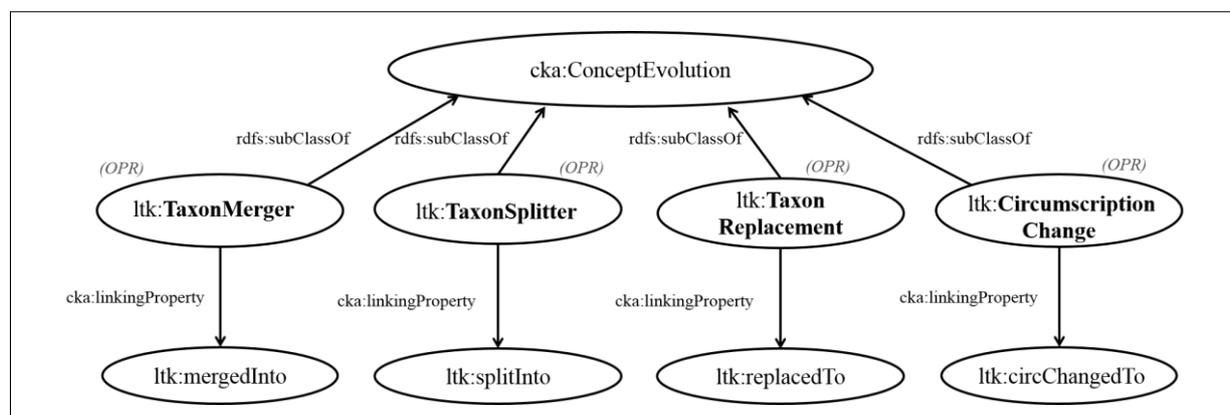


Fig. 2. Model: The declaration of operations for the changes in conception

of supporting a simple nominal entity, this research provides some following recommendations. (1) A URI should include a scientific name and a version. We recommend to use a year of the change as a version number such as *genus:Bubo_1999* (2) If a change affects the change in nomenclature, a new URI should be created and a link between the former and the latter URIs is developed to show the relationship between them. (3) In case a new URI of a taxon concept is recreated for some purposes without the change in scientific name, the version number in the URI string should be updated. The created contextual nominal entity can link to some nominal entities from external datasets in order to make data be globally linkable. According to the standard of TDWG [28], our research uses the property *dct:isVersionOf* for linking between a contextual nominal entity and an nominal entity, and the property *dct:identifier* for assigning an LSID [28].

In practice, we create a simple nominal entity a representative of an external URI for maintaining links between LTK dataset and external datasets. It is possible to link the contextual nominal entity with other taxonomic data such as URIs or LSIDs from TDWG [28], GBIF [25-27], Catalog of Life (CoL) [18], LODAC [15], and DBpedia [24] via those representatives. For example, the following statement addresses an association among the contextual nominal entity (*genus:Bubo_1999*), the simple nominal entity as the representative of the external URI (*genus:Bubo*), the external URI viewed as nominal entity (*gbif:5959091* and *lodac:Bubo*), and LSID (“*urn:lsid:ubio.org:namebank:2473659*”).

```
genus:Bubo_1999
  dct:isVersionOf genus:Bubo .

genus:Bubo
  owl:sameAs gbif:5959091 , lodac:Bubo ;
  dct:identifier
    "urn:lsid:ubio.org:namebank:2473659".
```

Change Entity (Operation)

Change Entity or operation of change is a type of change in taxonomic knowledge, which is previously described, for example replacing, merging, splitting, reclassifying, etc. In practice, the operations are subclasses of either *cka:ConceptEvolution* or *cka:RelationshipEvolution*. Our research generally uses instances of operations for managing changes in contextual nominal entities. Moreover, a link between operations can viewed as a link between background knowledge.

Event Entity

In order to reduce data redundancy, an event entity is created for assuring some operations that share the same aspects of time and provenance. Thus, time interval and references are assigned to the event entity. In practice, the event entity is an instance of *cka:CommunityKnowledge*.

For the use of each entity, it is noted that our work does not restrict the representation of URIs, a simple nominal entity, unfriendly identifier, or separation of name and taxon concept are possible to use in our model.

In addition, this research views that the nominal entity, the simple normal entity, and the contextual nominal entity are concepts, which are subclass of *skos:Concept*. Because a change usually performs an action with concepts, from now on, when this paper mentions the term “concept” in the context about change or with an operation of change, it mostly refers to a contextual nominal entity.

At last, since each entity is a semantic web resource, we add some symbols in some figures in order to distinguish the types of entities:

- (*nom*) is an instance of a nominal entity,
- (*tax*) is an instance of a taxon concept,
- (*sim*) is an instance of a simple nominal entity,
- (*con*) is an instance of a contextual nominal entity,
- (*OPR*) is a class of a change entity (operation),
- (*opr*) is an instance of an operation, and
- (*event*) is an instance of an event entity.

3.2.2. Data Models for LTK

In addition, in order to have researchers interpret data precisely, our knowledge management introduces various models of knowledge representations.

Event-Centric Model

The event-centric model is a data structure that used to preserve the change in taxonomic knowledge in RDF. It is based on the idea of CKA [8] that used *n*-ary relation for creating context-dependent RDF statements including operations, time interval, and references [8-10]. Thus, the RDF presentation of this model is quite complicated by design. Although the model is expensive, it is advantageous to various applications especially in knowledge management systems.

Transition Model

Transition model is a model for presenting the chain of change in contextual nominal entities. This model is transformed from the event-centric model by semantic

web rules. This model is presented as a general graph including only contextual nomial entities and their links, so it is simpler than the event-centric model and is easy to work with linked data, but not good for representing background knowledge in detail.

Snapshot Model

Snapshot model is a set of simple RDF statements like the transition model, but it is generated according to a given time point. This model demonstrates how information of a taxon changes over time.

3.3. A Formal Model for the Change in Taxonomic Knowledge

As mentioned in the previous section, the change in contextual nomial entities and the change in relationship between them are key players in linking taxonomic knowledge. To present general definitions for the change in taxonomic knowledge, we propose a formal model for preserving and presenting the change in taxa for linked data. Our formal model is basically derived from the CKA approach [8]. The approach introduces a basic idea how to reuse super classes from CKA to create an operation of the change in concepts and an operation of the change in relationship between two concepts, and how to map an operation with a semantic web property.

3.3.1. Change in Conception

In this part, we reviewed the function of the change in concepts, *cka:ConceptEvolution*, that deals a set of concepts before the change and a set of concepts after the change. By reusing this function, we categorize the change in contextual nomial entities into following functions: *ltk:TaxonMerger*, *ltk:TaxonSplitter*, *ltk:TaxonReplacement*, and *ltk:CircumscriptionChange*. These operations are associated with the following properties: *ltk:mergedInto* (merged into), *ltk:splitInto* (split into), *ltk:replacedTo* (replaced to), and *ltk:cirChangedTo* (circumscription changed to) respectively. Figure 2 shows the derivation of these operations, which are subclasses of *cka:ConceptEvolution*, and links to their associating properties. In some special cases, some basic changes such as merging and splitting occurring at once. This complex case is solved by the operation named *ltk:TaxonComplexChange*, which is a subclass of *cka:ConceptEvo-*

lution. This operation allows to have multiple concepts before and multiple concepts after a change, and the linking property of this operation is *ltk:cpxChangedTo* (complexly changed to). However, if it is possible to do, this research recommends clarifying the complex change into the simple operations: merging, replacing, and splitting, for better understanding.

3.3.2. Change in Relationship between Taxa

In addition to the change in conception, we considered the operation of the change in the relationship between two things. The term relationship is used as a link not only for the same rank of taxon such as synonym, but also for different ranks such as hierarchical relationship. CKA [8] provides a superclass *cka:RelationshipEvolution* for capturing the change in association between two concepts. This operation generally records the transition of a triple by changing the object of the triple, but the subject and the predicate of the triple remain unchanged. In LTK, we introduce operations to give evidence of the change in the relationship between two contextual nomial entities. For example, the change in the classification of a taxon, *ltk:ChangeHigherTaxon*, contributed a procedure to record the change in the higher taxon rank of a taxon by switching the object of a predicate named *ltk:higherTaxon* to another one. Therefore, the operation *ltk:ChangeHigherTaxon*, which is a subclass of *cka:RelationshipEvolution*, is in charge of alternating a triple containing the relationship property named *ltk:higherTaxon*, as shown in Figure 3.

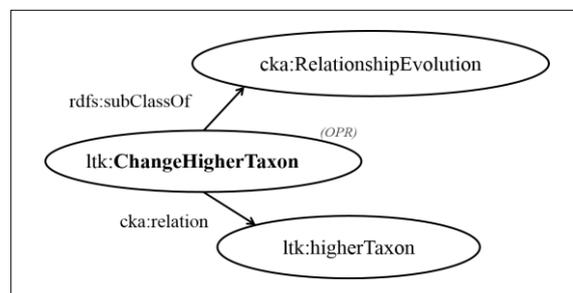


Fig. 3. Model: An example declaration of the change in relationship between taxa

3.4. Working with the Event-Centric Model

In this part, we present how to work with the event-centric model in order to capture the change in taxonomic knowledge. Here we suppose the following simple test case: there are two families Audae and Buidae, and Buidae includes one genus *Xus* at time t_1 ; then, at time t_2 , Buidae is merged into Audae and subsequently the genus *Xus* is regarded as a member of a new URI of Audae. In this case, at time t_1 , we assign URIs of contextual nominal entities for Audae, Buidae, and *Xus* that are *ex:Auidae_1*, *ex:Buidae_2*, and *ex:Xus_1* respectively. When two families are merged into Audae at time t_2 , according to the use of the contextual nominal entity, it has to create a new URI of Audae to be *ex:Auidae_2*. Then, the genus *ex:Xus_1* is transferred to the newer accepted family. In nomenclature, a taxon higher than genus level is not needed to change its scientific name when it is transferred to another higher taxon [1,45]. Thus, the current URI of the genus *ex:Xus_1* is retained. However, if the change affects the scientific name of a taxon, a new contextual nominal entity has to be created and a link between an old concept and a new concept has to be identified. Figure 4 demonstrates the changes in taxa, the change in relationship between them, and the event entity. First, the operation, *ex:merge1*, is the merging of *ex:Auidae_1* and *ex:Buidae_1* into *ex:Auidae_2*. Thus the given values of *cka:conceptBefore* are *ex:Auidae_1* and

ex:Buidae_1, while the given value of *cka:conceptAfter* is *ex:Auidae_2*. Second, the change in relationship between contextual nominal entities, *ex:reclass1*, is the reclassification of *ex:Xus_1* from *ex:Buidae_1* to *ex:Auidae_2*. Hence, *ex:Xus_1*, *ex:Buidae_1*, and *ex:Auidae_2* are assigned to *cka:child* (\equiv *cka:subject*), *cka:parentBefore* (\equiv *cka:objectBefore*), and *cka:parentAfter* (\equiv *cka:objectAfter*), respectively. Moreover, according to this scenario, *ex:merge1* contributed to *ex:reclass1*, so it can express that *cka:effect* maps *ex:merge1* to *ex:reclass1*. Last, the event entity named *ex:event1*, which is an instance of *cka:CommunityKnowledge*, confirms the two changes as mentioned above by a property named *cka:assures*, and identifies a temporal identity by a property named *cka:interval*. The temporal identity mentions a begin time point " t_1 " by a property *tl:beginAtDateTime*, and an end time point " t_2 " by a property named *tl:endAtDateTime*.

3.5. Working with Semantic Web Rules

The example mentioned in the previous section, which introduced context-dependent RDF statements, are general patterns for representing the change in taxonomic knowledge. The event-centric model is complex by design because it is used to preserve the change with context information. For the use of linked data, the complex expression detailed by the event-centric model is not suitable, because it is difficult for

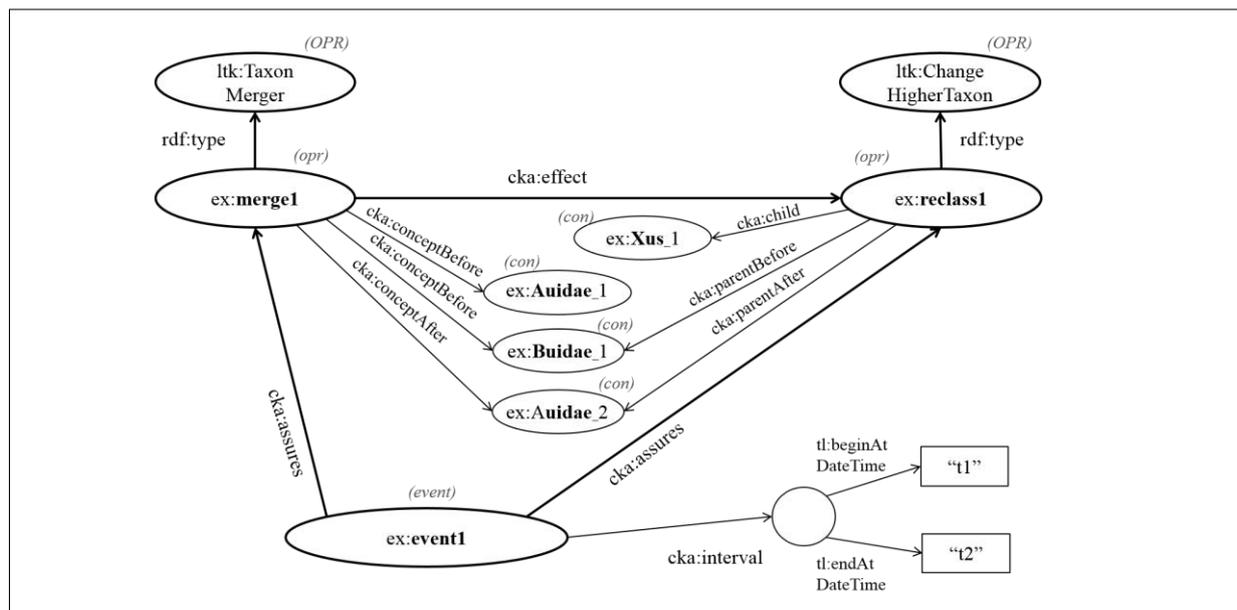


Fig. 4. Model: Example event-centric model representing changes in taxonomic knowledge

making implicit links using existing semantic reasoners. Therefore, it has to transform the event-centric model into two easily-linkable models: the transition model and the snapshot model.

3.5.1. Generating the Transition Model

First, we transform the event-centric model into the transition model. The following example semantic web rule gives a link between contextual nominal entities before and after merging.

```
TaxonMerger(?opr)
∧ conceptBefore(?opr, ?c1)
∧ conceptAfter(?opr, ?c2)
⇒ mergedInto(?c1, ?c2)
```

The rule contains symbols named $?opr$, $?c1$, and $?c2$ which are the variables of an operation of change, a contextual nominal entity before the change, and a contextual nominal entity after change, respectively. In this case, it is a merging operation, so the variable $?opr$ must be defined as an instance of the operation *TaxonMerger*. Executing this rule results in the linked data of taxa — *ex:Aus_1*, *ex:Bus_1*, and *ex:Aus_2* — as shown in Figure 5. In this research, the change is usually transformed into the transition model without any time references in order to demon-

strate a timeline graph, but the time interval is employed by the snapshot model for displaying temporal changes in a concept. Moreover, in practice, we define a generic rule for each case, so the class named *TaxonMerger* and the property named *mergedInto* have to be represented by variables instead. The following statement expresses the common rule to link concepts before the change ($?c1$) and concepts after the change ($?c2$), where the link is represented by a property ($?p$) bound with the operation of change ($?OPR$). Then, a triple containing $?c1$, $?p$, and $?c2$ is produced.

```
subClassOf(?OPR, ConceptEvolution)
∧ linkInProperty(?OPR, ?p)
∧ type(?opr, ?OPR)
∧ conceptBefore(?opr, ?c1)
∧ conceptAfter(?opr, ?c2)
⇒ ?p(?c1, ?c2)
```

3.5.2. Generating the Snapshot Model

Second, we introduce a rule to transform the event-centric model into the snapshot model. Before executing the following rule, it is necessary to use a query statement to find only some changes that contains a given concept and covers a given time point. After that, a property ($?p$), which is bound with an operation of the change in relationship ($?opr$), maps a subject ($?s$)

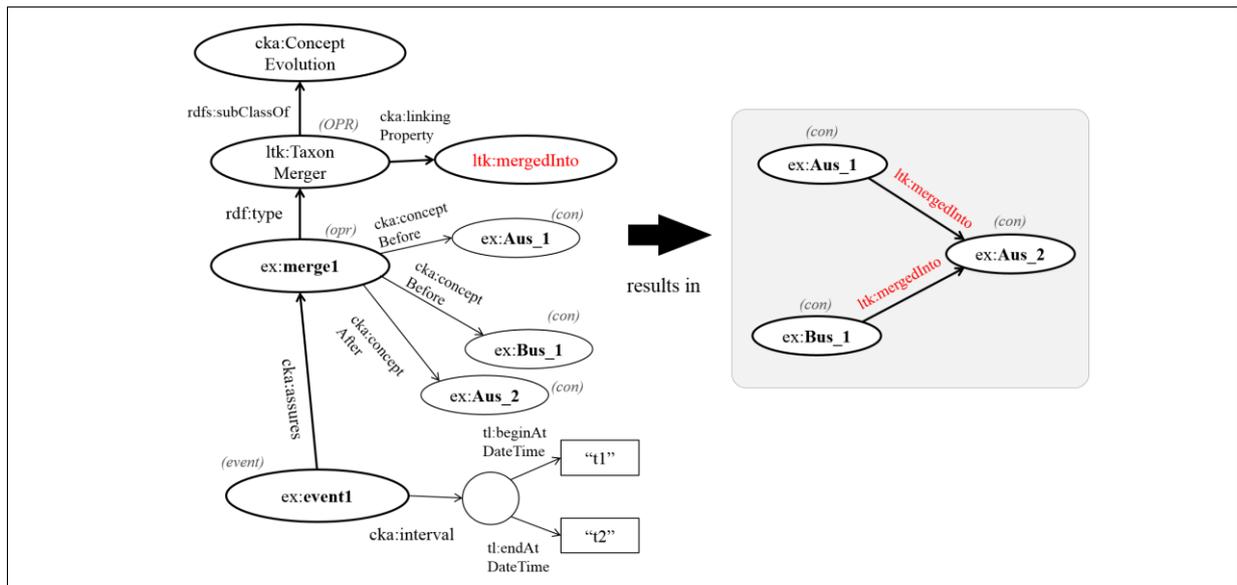


Fig. 5. Rule: Transforming the event-centric model into the transition model

and an object after the change ($?O_{after}$) to construct a triple.

```
subClassOf(?OPR, RelationshipEvolution)
∧ relation(?OPR, ?p)
∧ type(?opr, ?OPR)
∧ subject(?opr, ?s)
∧ objectAfter(?opr, ?Oafter)
⇒ ?p(?s, ?Oafter)
```

In addition, if the given time point is earlier than the begin time of a change, an object before the change ($?O_{before}$) became an object of a triple formed by the following rule. However, any changes ending before the given time point are not considered in this process.

```
subClassOf(?OPR, RelationshipEvolution)
∧ relation(?OPR, ?p)
∧ type(?opr, ?OPR)
∧ subject(?opr, ?s)
∧ objectBefore(?opr, ?Obefore)
⇒ ?p(?s, ?Obefore)
```

Consequently, Figure 6 shows that the classification of a genus $ex:Xus_1$ is interpreted variously according to different time points. The result after performing the rules is that the concept $ex:Xus_1$ is under the family $ex:Buidae_1$ before time $t1$, while $ex:Xus_1$ becomes under the family $ex:Auidae_2$ during the time between $t1$ and $t2$.

3.6. Representing the LTK Approach in RDF

Having proposed the formal description and rules, we now demonstrate how to utilize the RDF model to present and execute the change in taxonomy described in the previous sections. According to the change in the genus *Columba*, the following statements give data of *Columba* in RDF format. Initially, our work presents the relationship between a species and a genus using a property *ltk:higherTaxon*, and uses the notation *species:* and *genus:* as namespaces of species and genera respectively.

```
species:Columba_speciosa_1789
  ltk:higherTaxon
    genus:Columba_1758 .
```

Then, the following RDF statements express the event entity and operation for splitting the genus *Columba* together with a reference time point.

```
ex:event2003
  cka:interval
    [tl:beginAtDateTime "2003"] ;
  cka:assures      ex:split1 .

ex:split1
  rdf:type ltk:TaxonSplitter ;
  cka:conceptBefore
    genus:Columba_1758 ;
```

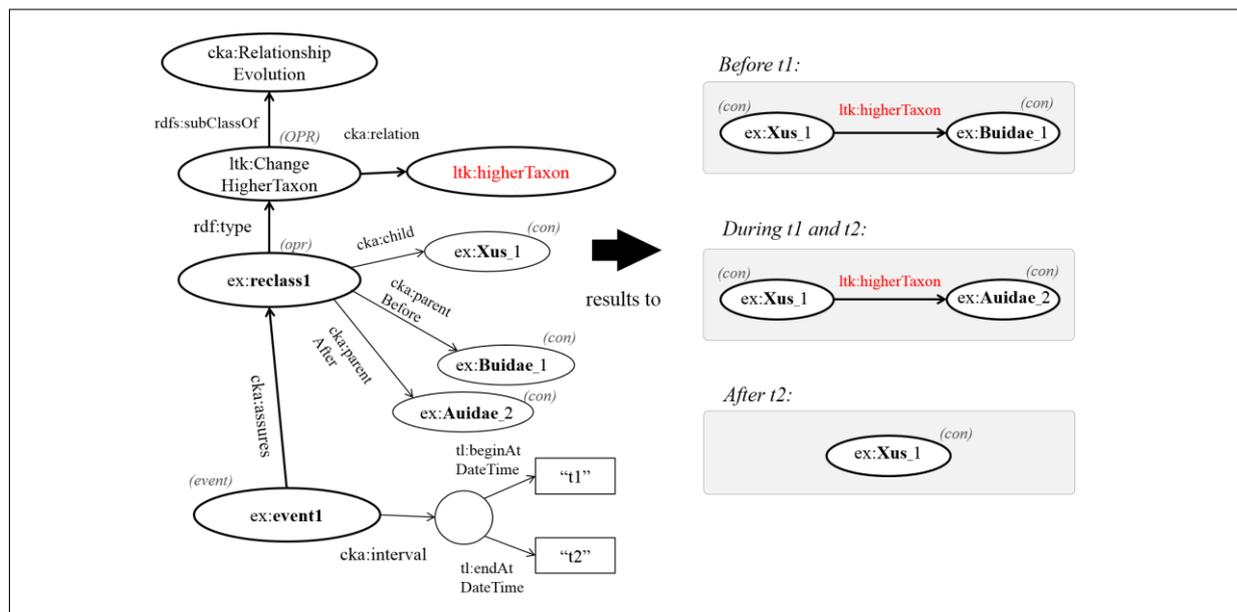


Fig. 6. Rule: Transforming the event-centric model into different snapshot models according to different time points.

```

cka:conceptAfter
  genus:Patagioenas_2003 ,
  genus:Chloroenas_2003 ,
  genus:Lepidoenas_2003 ,
  genus:Oenoenas_2003 ,
  genus:Columba_2003 .

```

Furthermore, the framework provides a technique to transform the event-centric model into the transition model along with a given concept. For example, links between the genus *Columba* and the new concepts after splitting can be shown as follows:

```

genus:Columba_1758 ltk:splitInto
  genus:Patagioenas_2003 ,
  genus:Chloroenas_2003 ,
  genus:Lepidoenas_2003 ,
  genus:Oenoenas_2003 ,
  genus:Columba_2003 .

```

3.6.1. Reusing CKA Framework

Technically, the CKA framework allows other ontologies to customize own operations of changes for particular purposes. This is done by extending either the class *cka:ConceptEvolution* for the change in a concept's scope or the class *cka:RelationEvolution* for the change in the binary relationship between two concepts. For example, the operations of the change in taxon concepts, such as *ltk:TaxonMerger* and *ltk:TaxonSplitter*, are descended from *cka:ConceptEvolution*. Thus, when there are new properties that are not a part of either CKA or LTK, such as morphological, molecular, or ecological traits, new operations need to be initiated by extending one of the mentioned classes from CKA and then binding the new operations with related properties.

3.6.2. Linking Background Knowledge

In addition, this model offers an association between related operations of changes by having two properties, *cka:cause* and *cka:effect*, to express the reason and the outcome of a change, respectively. For example, Figure 7 shows the previous information of the newly registered name *Patagioenas speciosa*. Moreover, a property *cka:detail* is sometimes used for linking some details of a concept after change such as adding metadata. Consequently, we can find the history of the name *Pa-*

tagioenas speciosa and then use its background concepts, such as the old name *Columba speciosa*, to explore more information published in LOD.

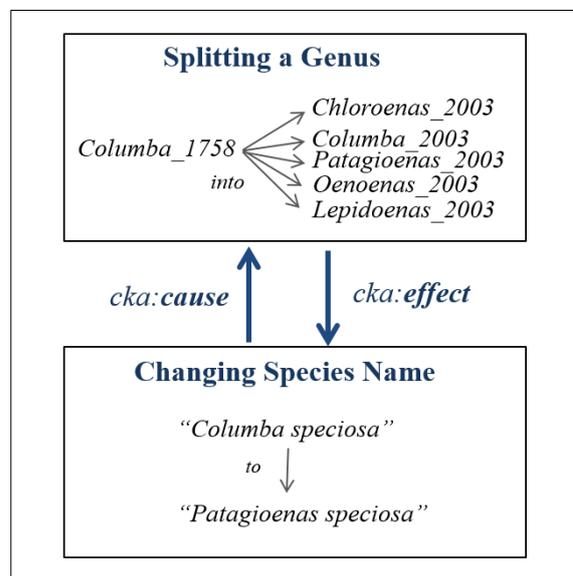


Fig. 7. Cause and effect between two changes

3.6.3. LTK Model in Practice

To link data with LOD Cloud, we have proposed useful operations that specify the change in concepts, the changes in details of a concept, the changes in relationships between concepts, and the background information of the change. All operations are defined by extending vocabularies from the well-known ontology named SKOS and properties from LODAC and CKA. The namespaces and example properties used by our model are described in Appendixes A and B. As a result, the data from our approach can be linked to data from other repositories.

For instance, the old concepts *genus:Nyctea_1826* and *genus:Bubo_1805* have been merged into a new concept named *Bubo*. As stated previously, the new identifier of the genus *Bubo* has to be initiated, because its new scope is larger than the former one. According to the recommendation, the identifier should be ended with a string representing the year in which the new URI was created, so the new identifier of *genus:Bubo_1805* becomes *genus:Bubo_1999*. To link between concepts before and after the change, LTK

provides the property named *ltk:mergedInto* to represent the a relationship between a concept before and a concept after merging. As a result, the relationship between *genus:Nyctea_1826* and *genus:Bubo_1999* remains to be associated by the property *ltk:mergedInto*. Moreover, in the case where a former concept and a latter concept have the same name or their circumscriptions are very close, a property *ltk:majorMergedInto* is recommended to demonstrate the very close relationship between them, such as *genus:Bubo_1805* and *genus:Bubo_1999*. To handle this situation, the model allows to use the property *cka:majorConceptBefore* for the operation of merging and the property *cka:majorConceptAfter* for the operation of splitting. As the genus *Nyctea* was merged into the genus *Bubo*, all species under the genus *Nyctea*, such as *N. scandiaca*, have to transfer to the genus *Bubo*, in this case the name of this species has to be changed to *B. scandiacus* according to the nomenclature [1,2,3,45]. The following RDF statements describe the merging of two genera, the renaming of a species under the genus *Nyctea*, and the change in a species under the genus *Bubo*. In this case, the *species:Bubo_scandiacus_1999* is newly generated without any higher taxa, so this event has to give it a higher taxon using the operation *ltk:HigherTaxonAddition* to originate a higher taxon of a newly generated URI.

```

ex:event1999
  bibo:performer
    pp:Wing, pp:Heidrich ;
  bibo:issuer pp:Richard ;
  dct:source pub:5224773 ;

  cka:interval
    [tl:beginAtDateTime "1999"] ;
  cka:assures ex:mgl, ex:rp1, ex:ac1 .

ex:mgl
  rdf:type ltk:TaxonMerger ;
  cka:majorConceptBefore
    genus:Bubo_1805 ;
  cka:conceptBefore
    genus:Nyctea_1826 ;
  cka:conceptAfter
    genus:Bubo_1999 .

ex:rp1
  rdf:type ltk:TaxonReplacement ;
  cka:conceptBefore

```

```

    species:Nyctea_scandiaca_1826 ;
  cka:conceptAfter
    species:Bubo_scandiacus_1999 .

ex:ac1
  rdf:type ltk:HigherTaxonAddition ;
  cka:child
    species:Bubo_scandiacus_1999 ;
  cka:parent genus:Bubo_1999 .

ex:mgl cka:cause ex:rp1 .
ex:rp1 cka:detail ex:ac1 .

```

3.6.4. Semantic Web Rules

After that, semantic web rules are implemented in order to transform the event-centric model into the transition model in RDF. For example, a Jena rule [37] that infers the merging operation using *cka:conceptBefore* of taxon concepts is

```

[rule_merge:
  (?opr rdf:type ltk:TaxonMerger),
  (?opr cka:conceptBefore ?before),
  (?opr cka:conceptAfter ?after)
->(?before ltk:mergedInto ?after) ]

```

In addition, the rule for *cka:majorConceptBefore* is the modification of the rule *rule_merge* by changing the *cka:conceptBefore* to *cka:majorConceptBefore*, and changing the *ltk:mergedInto* to *ltk:majorMergedInto*.

Moreover, the entered (*cka:entered*) and expired (*cka:expired*) time points of a concept are also generated by the following example rule. However, in practice, this rule should be split into several ones in order to handle all possible cases that contains only some of properties: *tl:beginAtDateTime*, *tl:endAtDateTime*, *cka:conceptBefore*, and *cka:conceptAfter*.

```

[rule_time_span:
  (?event cka:interval ?inv),
  (?inv tl:beginAtDateTime ?begin),
  (?inv tl:endAtDateTime ?end),
  (?event cka:assures ?opr),
  (?opr rdf:type cka:ConceptEvolution),
  (?opr cka:conceptBefore ?before),
  (?opr cka:conceptAfter ?after)
->(?before cka:expired ?begin),
  (?after cka:entered ?begin),
  (?after cka:expired ?end) ]

```

In practice, the rules for transition model are also performed, so the change in the given concept itself at the given time point is also presented. When all rules are executed, the following inferred RDF statements are produced to present the associations between changed taxa.

```
genus:Nyctea_1826 ltk:mergedInto
    genus:Bubo_1999 .
genus:Bubo_1805 ltk:majorMergedInto
    genus:Bubo_1999 .

species:Bubo_scandiacus_1999
    ltk:higherTaxon
    genus:Bubo_1999 .
species:Bubo_scandiacus_1999
    ltk:synonym
    species:Nyctea_scandiacus_1826 .

genus:Nyctea_1826 cka:expired "1999" .
genus:Bubo_1805 cka:expired "1999" .
genus:Bubo_1999 cka:entered "1999" .
species:Nyctea_scandiacus_1805
    cka:expired "1999" .
species:Bubo_scandiacus_1999
    cka:entered "1999" .
```

A transfer into a simple RDF statement containing a subject, a predicate, and an object is useful for a client. This simple format is easier for working with well-known ontologies in order to query by well-known properties as defined in Appendix B. For example, the properties *skos:closeMatch* and *lodac:hasSuperTaxon* in query statements can produce the same results as the ones from *ltk:synonym* and *ltk:higherTaxon*, respectively. This approach also allows users to check existence of a concept by inquiring either the property *cka:entered* or the property *cka:expired*.

3.6.5. Working with other Operations

Although this research focuses on the change in taxonomic data, some triples that are not changed over time are recommended to be preserved by even-centric model, because it can present some essential metadata such as a date added and some references. Moreover, if some domains require more operations of changes, the operations can be created by extending the *cka:RelationshipEvolution*. This method is also compatible

with some systems that separate a taxon concept and a name. Our model also allows having operations for either object property or datatype property. Some example properties or attributes are such as *dwc:identificationID* [31], *dwc:taxonID* [31], *dwc:scientificNameID* [31], *dwc:scientificName* [31], *foaf:depiction* [32], *skos:prefLabel* [29], and *lodac:hasCommonName* [15]. Some of them are described in Appendix C.

In conclusion, the introduced logical model includes the data model for the change in taxonomic knowledge, and semantic web rules for transforming the event-centric models into a simple linked data model. It also presents how to use the model for real-world cases of the change in taxonomic knowledge in RDF. However, if more properties are needed for a specific purpose, developers can customize their operations by extending this framework.

4. Prototype

Our proposed logical model is intended to manage the change in taxonomic knowledge represented in RDF. In order to verify the possibility and feasibility of our work, a prototype has been developed. The main purpose of its implementation is to execute and present changes in taxonomic knowledge. System architecture and a demonstration for a web application to realize our approach are also presented. In addition, information about our prototype is available at a website "<http://rc.lodac.nii.ac.jp/ltk/>".

4.1. Functionalities

The prototype is implemented on the basis of two key functions: (1) defining and executing the change in taxonomic knowledge, and (2) presenting the temporal information of an Internet resource used in taxonomic knowledge.

The first function allows users to input changes in taxonomic knowledge by recording a list of operations, their parameters, and metadata. It also offers a bulk load feature for importing the event-centric model in RDF into the system directly. When the input data is submitted, rule-based reasoning produces the relation-

ships between concepts resulted in the change in taxonomic knowledge, and then the system collects the RDF data in an RDF data store.

In addition to the execution of the event-centric model, the other function offers an interface to present temporal information and linked data of a given concept. The prototype lets users browse the URI of a given concept with a given time point in *xsd:dateTime* format, and then it displays the temporal information of the concept together with its related concepts resulted from the change and any background information regarding changes.

4.2. Implementation

In order to accomplish these key activities, we analyzed the functions, designed system architecture, employed well-known open source tools, and did the programming to implement the web application for end users and service interfaces for client applications. The system architecture of the prototype is a web-based system, as shown in Figure 8, comprising three layers: presentation layer, business logic layer, and data access layer.

The presentation layer displays information related to such services as creating and executing the change in a given concept, and presenting the taxonomic

knowledge. It communicates with other service endpoints by outputting results to users or client applications. The user can browse the information by using a web application created by PHP, whereas the client applications can access the data by using LTK web services written in Java, and SPARQL endpoint provided by OpenRDF [38].

In addition to the presentation layer, the business logic layer controls an application’s functionality by performing data processing. Knowledge Engine, a Java-based component, is the main module that manages the RDF-based event-centric models together with semantic web rules and related ontologies in order to construct taxonomic knowledge and linked data of Internet resources of taxonomic data. Technically, this component normalizes and forwards RDF data to the data store directly. It also queries RDF data via the SPARQL engine with an API from OpenRDF. Moreover, a semantic web rule engine developed using Apache Jena [37] transforms the event-centric model into the transition model and the snapshot model.

Lastly, the data access layer built for the storage and retrieval of triples collects subject-predicate-objects from components in the upper layers. Our experiment uses OpenRDF, which provides high capacity with great performance. It additionally offers an API that performs well with Jena.

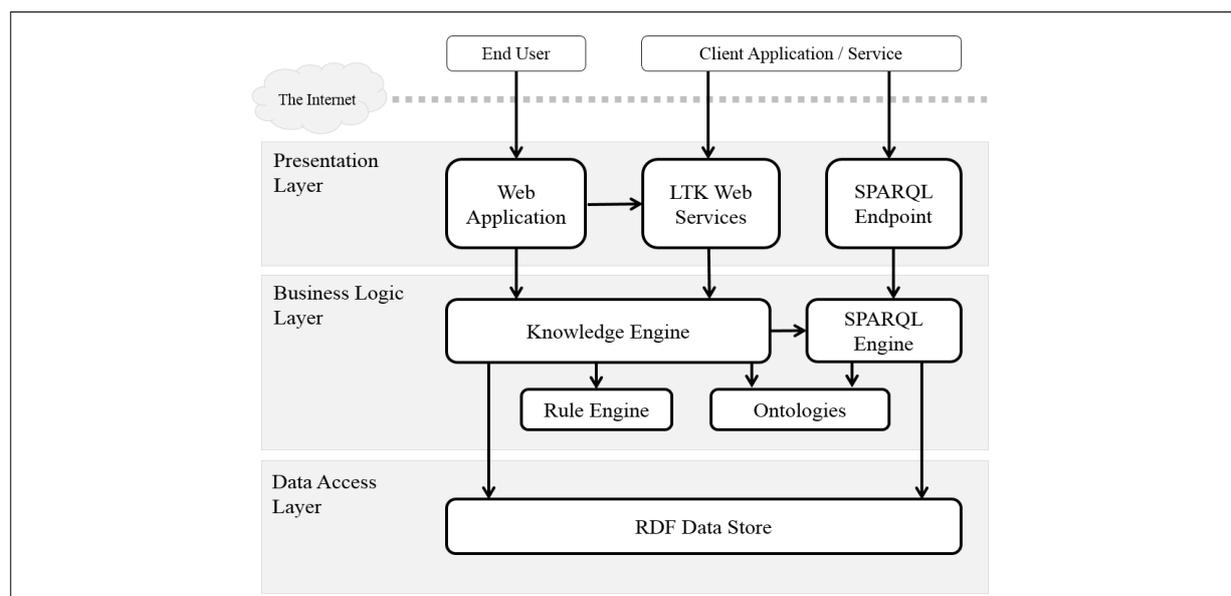


Fig. 8. Prototype: System Architecture

All of these layers run on a server that is connected to the Internet, so it is ready to serve the LTK services to end users or client applications. Moreover, the system architecture is flexible for enabling application to other domains. Developers can customize semantic web rules and ontologies to their own requirements, and publish their data for open access.

4.3. LTK Services

As a result of the services provided in the presentation layer, all interfaces are conveniently accessible over the Internet. This section illustrates how to use services from this prototype by describing web application and web services.

4.3.1. Web Application

Beginning with the web application, it contains two main parts, an administration interface and a user interface.

An administration interface provides a tool to ingest a list of changes in concepts. Every change can be done by choosing an operation such as merging, replacing, and splitting, and then assigning a concept or a value to the required properties. After that, the users

can state the relationship between changes in the case where one change relates to another change by linking them with a property named cause, effect, or detail. Finally, the prototype allows users to prepare metadata of these changes, such as a begin time point, an end time point, performers (e.g. researchers) who discovered the change, reporters who announced the change, and references such as publications.

Apart from the administration interface, a user interface is implemented as a browser for presenting the information of a given concept. The web page shows historical information of a taxon concept including point temporal data; its related concepts resulted from the change, and links of its related concepts. The user has to specify a URI of a concept together with a particular time. For this prototype, a URL pattern “*http://[ltk_domain]/*” denotes the domain name of our prototype where “[ltk_domain]” in our experiment is “*rc.lodac.nii.ac.jp*”. The pattern of a request for displaying information of a given concept in a given time point is

```
http://[ltk_domain]/ltk/concept.php?
concept=[concept]&date=[time_point]
```

where “[concept]” is a URI of a given concept, and “[time_point]” is a given time point in the format *xsd:dateTime*. For example, at a given time point

The screenshot shows the LTK web application interface for the concept *Nyctea scandiaca*. The page is divided into several sections:

- Header:** "Linked Taxonomic Knowledge" with a logo.
- Concept:** "Nyctea scandiaca" in blue text.
- Preview:**
 - An image of a white bird in flight.
 - A table showing the present status:

Present status	Expired
Entered	1826
Expired	1999
 - An input form with fields for "Concept" (URI: <http://rc.lodac.nii.ac.jp/taxon/sp>) and "Time point" (2014-12-18T02:15:00Z), with a "View" button.
- Information:** A table listing relationships:

Subject	Predicate	Object
species:Nyctea_scandiaca_1826	skos:prefLabel	"Nyctea scandiaca"
species:Nyctea_scandiaca_1826	dct:isVersionOf	ltk:Nyctea_scandiaca
species:Nyctea_scandiaca_1826	rdf:type	lodac:Species
species:Nyctea_scandiaca_1826	ltk:higherTaxon	genus:Nyctea
species:Nyctea_scandiaca_1826	ltk:replacedTo	species:Bubo_scandiacus_1999
species:Nyctea_scandiaca_1826	foaf:depiction	http://www.natgeocreative.com/comp/MI/001/1304380.jpg
- Linked Concepts:** A table showing a replacement:

Subject	Object	Date
species:Nyctea_scandiaca_1826	ltk:replacedTo	species:Bubo_scandiacus_1999 01 Jan 1999 -

Fig. 9. Model: View showing taxonomic knowledge of a taxon

“1998-01-01T00:00:00Z”, the species *Bubo virginianus* was classified into *genus:Bubo_1805*. After the merging of the two genera, *Bubo* and *Nyctea* in 1999, the species *B. virginianus* has technically been reclassified into the newer genus *genus:Bubo_1999*. Thus, the request with a time points after 1999 presents that the genus of this species is *genus:Bubo_1999*. In addition, users can request only

`http://[ltk-domain]/taxon/[rank]/[name]`

in the web browser directly; where “[rank]” is a taxonomic rank, and “[name]” is a taxonomic name string including a version label. The Accept request-header being “text/html” redirects to a webpage with current date time, while the request with “text/plain” returns an N-Triples format of RDF. Another example is indicated in Figure 9 that shows temporal information of a species *Nyctea scandiaca*. This page includes three main sections. First, a photo of the species is displayed together with its present status, entered date, and expired date. Second, the section “Information” displays temporal data, which can be classification, description, label, etc., that retrieve from the snapshot model and the transition model at the given time point. The last section “Linked Concepts” demonstrates the transition model of the given concept. Moreover, the back-

ground knowledge of the change in concepts is described when users choose a button labeled “i”. A web document titled “Background of the Change” appears and reveals the detail of change, reason behind the change, and metadata. The Figure 10 presents the changes in the *Nyctea scandiaca* that was caused by the merging of the two genera, *Bubo* and *Nyctea*. It also gives reference information, such as, researchers, academic papers, website, etc. in order to confirm evidence of that particular change.

4.3.2. Web Services

In addition to the web application, there are LTK web services and a SPARQL endpoint that provide data to client applications. Example datasets were loaded into OpenRDF [38] storage via LTK web service. The SPARQL endpoint for querying the links between concepts resulted from the changes can be accessed at

`http://[ltk_domain]/ltk-service/sparql/ltk`

This endpoint also offers the ability to query for temporal data of a given concept. However, LTK-Service provides a service to present the temporal information

Linked Taxonomic Knowledge

Background of the change

Detail of change:

Change Operation	ltk:TaxonReplacement
cka:conceptBefore	species:Nyctea_scandiaca_1826
cka:conceptAfter	species:Bubo_scandiacus_1999

Caused by:

Change Operation	ltk:TaxonMerger
cka:majorConceptBefore	genus:Bubo_1805
cka:conceptBefore	genus:Nyctea_1826
cka:conceptAfter	genus:Bubo_1999

Description:

Property	Value
Begin at	1999
End at	-
Performed by	person:Wink person:Hedrich
Reported by	person:Richards
References	http://www.aou.org/checklist/suppl/AOU_checklist_suppl_44.pdf

Fig. 10. Prototype: Background information about the change

of a given concept at a given time point in the N-Triples format by requesting the following URL.

```
http://[ltk_domain]/ltk-service/context?
concept=[concept]&time=[time_point]
```

The background knowledge of the change that relates to a link of two concepts is available at

```
http://[ltk_domain]/ltk-service/reason?
subj=[subject_concept]&obj=[object_concept]
```

where “[subject_concept]” and “[object_concept]” are URIs of two associated concepts.

5. Evaluation

In this section, we proved the feasibility of our approach by making an experiment on the prototype. We firstly evaluated our approach against the use cases from the domain experts and found that our research covers the practical use cases. Second, we tested that the complexity of the event-centric model, which consumes much system resources, did not affect the overall performance of a system.

5.1. Evaluation against Use Cases

In this evaluation, we imported the example cases from Section 2 and some data of Japanese moths of the family Saturniidae published by three checklists (list of names): Inoue in 1982 [39], Jinbo in 2008 [40], and Kishida 2011 [41]. One of the authors, Jinbo, analyzed the difference among these three checklists, and finalized them into the changes in taxa among these checklists. The data cover operations of changes with are creating a concept, obsoleting a concept, replacing a taxon, merging taxa, splitting a taxon, linking synonym, changing a higher taxon, subdividing a taxon, and combining taxa. This experiment contains forty instances of operations together with sixty taxa from several taxonomic ranks: family, subfamily, genus, species, and subspecies. Here, we pick up one example. In [39], the species *Caligula boisduvalii* has two subspecies *Caligula boisduvalii falax* and *Caligula boisduvalii jonasii*. In the subsequent study, this species was transferred from the genus *Caligula* to *Saturnia*, and its two subspecies *boisduvalii* and *jonasii* were raised into two distinct genus. Hence, in that study, *Caligula boisduvalii* in [39] is redefined as two species *Saturnia boisduvalii* and *Saturnia jonasii*. At

the same time, the latter species was split into two subspecies *Saturnia jonasii jonasii* and *Saturnia jonasii fallax*. These changes are adopted in the second checklist [40]. After a few years, both subspecies are combined into the species *S. jonasii* in [41]. These changes resulted in many links of synonyms. Even though these events are described in taxonomic papers, information on events are not included in each name and thus cannot be captured by the databases of scientific names. Some entities of background knowledge of the change in *S. jonasii* were linked so users could browse to the accurate history of taxa, which is difficult to access for non-taxonomic experts. Therefore, the benefit driven by the management of the change in concepts, such as presenting the links between concepts in the chain of the changes in taxonomic knowledge, temporal information about them, and the underlying knowledge to that change, was convenient for gathering correct data along with precise context; so it reduced confusion and helped avoid misunderstanding arising with respect to taxonomic data. This experiment proved that the LTK approach could deal real-world situation of changes in taxonomy.

5.2. Performance Analysis

In addition to the usability evaluation, the performance of the prototype is tested. Our model essentially transforms a basic triple containing a subject, a predicate, and an object into a complex structure to express an event of a change in either a concept or a triple along with the reference time. As it consumes many more triples than the traditional form to present the same fact, the issue of performance becomes a key point in this research. We therefore verified the model with a great number of data and evaluated the query execution time by making comparison between our approach and a simple query as a baseline.

According to the data model, one event-centric model including ten operations required about one hundred triples. In this experiment, the number of test data in the repository was increased to up to 1,000,000 triples. For every increase of 100,000 triples, we measured the performance and recorded all the results in a chart. All steps in this experiment were performed on Linux 3.11.0-12 (64 bit) installed on an Intel quad-core i5 3.40 GHz PC with 32 GB of memory. The changes in data were stored in OpenRDF SESAME Ver. 2.7.7. To optimize query performance, RDF Schema and Direct Type Hierarchy Inferencing were enabled, so sequence triples were automatically generated from ones containing properties *rdf:type*,

rdfs:subClassOf, and *rdfs:subPropertyOf*. As a result, the dataset contains more than 5 million triples including inferred statements. The RDF repository additionally built two indexes: a subject-predicate-object-context (spoc) key pattern, and a predicate-object-subject-context (posc) key pattern, where a context is generally viewed as a graph name [38].

Our verification step was performed by comparing the result from our approach to baseline speed. To determine the basic speed of the SPARQL engine in our test, a baseline experiment was conducted using the following simple SPARQL statement for searching information on a species:

```
SELECT ?p ?o WHERE
{ species:Nyctea_scandiaca ?p ?o . }
```

Afterward, based on our approach, we made an inquiry for the same information on the same species that is valid at a given time by using LTK web services. As the result is returned in accordance with a time input, the system has to produce the result on the fly depended on the defined time point. The service transforms data from the event-centric models into the snapshot model using SPARQL statements together with semantic web rules as mentioned in the previous sections. The performance was measured by recording the response time of the web method. For accuracy result, data caching was disabled, and a given concept and a given time point were differently changed for every service request.

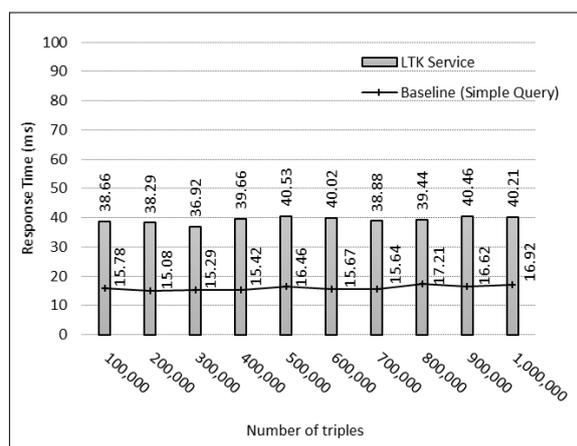


Fig. 11. Query execution time in the dataset

The result of the experiment are shown in Figure 11, which shows that the execution time from our approach is almost constant at about 0.039 second for every 100,000 input triples added into the repository, while the value from the baseline is approximately 0.016 second. A closer look at the result indicates that

our approach consumes slightly more execution time than a simple query by a millisecond unit. The results of our experiment provide confirmatory evidence that our framework is not causing application performance problems in the current software development even if dealing with millions of data.

6. Discussion

Many approaches [16-20] are usually focusing on keeping up-to-date taxonomic data. In practice, the change in knowledge is necessary for comprehensively studying biodiversity, however several previous works of taxonomic databases focused on the collection of name strings with proper identifiers at the first step of the integration of taxonomic information. Thus, the change in taxonomic knowledge is less discussed. Our work, LTK, provides a framework for preserving and presenting the change in taxonomic knowledge for linked data. We introduce operations for capturing the changes, such as merging, splitting, replacing, changing a higher taxon, etc. as shown in Appendix C. We discuss the values of our approach from four perspectives: knowledge representation, user engagement, system integration, and limitation.

6.1. Knowledge Representation

In term of knowledge representation, our research responds several requirements in order to have better understanding of biodiversity by giving different viewpoints of the change in taxonomic knowledge.

6.1.1. Historical Change in Taxa

Browsing chain of change in concepts is a feature that users can learn the historical change in the given taxon. LTK provides some properties indicating dynamic change in taxa for this feature. In other works, taxonomic Concept Schema (TCS) [47] is one of the well-known approaches to describe taxon concept in a way of informatics. This approach is attempted to be expressed as RDF by the work titled Describing Taxon Concept as RDF [48]. TCS regarded each concept as more static and classified operations of change into proper categories, so most operations seem to be more static than LTK. In term of using properties for representing dynamic changes in conception of taxa, our work introduced the hierarchy and configuration of the properties in Appendix D. Some properties such as *ltk:mergedInto*, *ltk:splitInto*, and *ltk:replacedTo* can

be simply used in the query statement. These properties are asymmetric and non-transitive object properties, so the query result returns only directed-adjacent nodes of the given concept. LTK can also present the main concepts in the time line by properties *ltk:majorMergedInto* and *ltk:majorSplitInto* which are sub properties of *ltk:mergedInto* and *ltk:splitInto* respectively. The *ltk:majorMergedInto* and *ltk:majorSplitInto* show that their subject and object are dominant in the change, so the concepts connected by these properties have stronger relationship than concepts linked by *ltk:mergedInto* and *ltk:splitInto*. In addition to getting the adjacent concepts, finding all concepts having the same history can be queried using properties *cka:serialLinkTo* and *cka:semanticLink*. The former, *cka:serialLinkTo*, is a transitive and asymmetric object property, so all concepts in only one direction in timeline occurring before or after the change in the given concepts can be queried. Besides, if it needs to find out all concepts in the same history, the query expression should be mention the property *cka:semanticLink*, which is a transitive and symmetric property and also a super property of *cka:serialLinkTo*.

6.1.2. Temporal Information of Taxa

The use of temporal data allows users learn the change in taxonomic knowledge in term of the change in triples, for example the changes in classification, membership, metadata, etc. Operations of changes that are found in the same publication or the event are grouped into one event-centric model, and aspects about time and provenance are assigned. Each operation assured

by the event entity can be transformed into two triples: one happening before the begin time point, and the other one happening during the begin and the end time points, but no triple generated after the end time point. However, these two triples are not directly stored in the database, so it needs to use query expression with semantic web rules to produce the snapshot model of a given concept at a given time point. In case a concept is given without a time point, the system assigns a current time by default. Although the event-centric model consumes many triples, the performance analysis from the previous section confirms that it is not an issue for current SPARQL engines. Thus, users do not only learn the association between data but also understand the precise context of the linked data by temporal information and references. They also recognize triples added or removed in different time, so they can learn the progress of taxonomic knowledge along with time.

6.1.3. Background Knowledge of the Change

Our approach responses similar objectives as TaxMeOn [6] in term of managing the change and linked data, but both works are technically different due to some specific purposes. TaxMeOn regularly presents a change using one triple containing an old taxon concept, a property indicating taxonomic change, and a new taxon concept; and it sometimes uses instances of operations of changes such as lumping and splitting. In this way, data model gives a simple and easily understandable timeline of the changes in taxon concepts. However, in case of using only one triple for representing a change, it is limited to give a link between

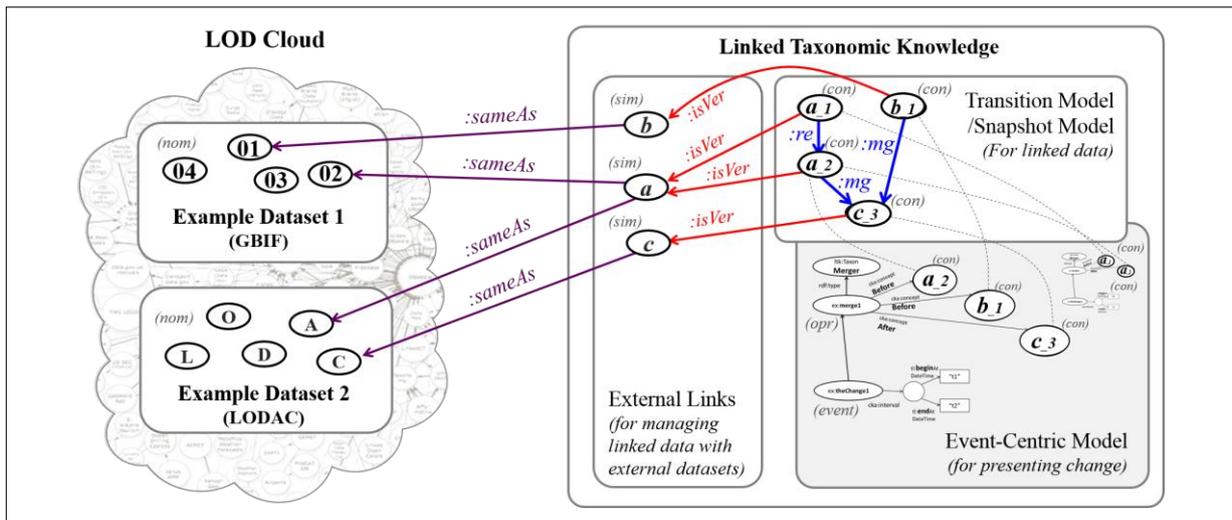


Fig. 12. The role of LTK (right) in LOD Cloud (left) containing example datasets.

Ovals with a single alphabet or an id number are general concepts, ovals with version are versions of general concepts, dashed lines show the same URIs, *:same* is *owl:sameAs*, *:isVer* is *dct:isVersionOf*, *:re* is *ltk:replacedInto*, and *:mg* is *ltk:mergedInto*.

changes, so associations between background knowledge cannot be implemented directly. In this case, the event-centric model becomes more advantageous to this requirement, because an instance of an operation can also be regarded as background knowledge, so the link between operations allows users to trace back to information behind the change. The properties *cka:cause* and *cka:effect* are used in a query string to find the reason and the result of a particular change respectively. Our prototype demonstrates how two concepts are related by finding some operations that are background knowledge of a link between the given subject and object.

6.1.4. Ability to Publish Linked Data

LTK approach is developed on the basis of the semantic web and the underlying community knowledge [8,9], so it can act as a medium that collects links among taxonomic data from different datasets, and provides background knowledge about how concepts are changed or linked. We encourage linking contextual nominal entities with external nominal entities from known datasets that are commonly referred by many applications and publications such as GBIF [27], CoL [18], uBio [19], and LODAC [15], using a property *dct:isVersionOf* in order to enable global access on data. The role of LTK in term of linked data is demonstrated in Figure 12. In the figure, LTK becomes medium of linked data having three parts. The first part consists of external links for representative concepts and links to external datasets; the second part includes the transition model and snapshot model; and the second part contains the event-centric model that acts as background knowledge of the change. On the basis of using open access data via SPARQL, making URIs be dereferenceable, and linking data to known dataset, our approach is possible to publish data to LOD cloud [7].

6.2. User Engagement

Another important task of building a taxonomic information system is to encourage users such as taxonomists, ecologists, and molecular biologists to participate in providing and utilizing data. However, many of them are non-computer-expert users. Since linked data, semantic web, and RDF syntax are relevant to each other; as far as we know the current situation of semantic web, we recommend users to understand basic RDF syntax in order to gain benefit from linked data. This research intends to keep taxonomic knowledge representation as simple as possible under the boundary of RDF framework.

6.2.1. Human Readability

Since the event-centric model is considered to represent data in various dimensions, so the RDF representation is complicated by designed. However, the simplicity of the model can be improved by the simple use of identifiers, and then the transition model and the snapshot model become consequently simpler. In term of human readability, the uses of the contextual nominal entity and the simple nominal entity are consistent with the idea of GNUB, which describes the usage of a name, and GNI, which collects name strings, respectively [23]. Thus, normalized and valid readable names are tied to checklist such as CoL [18]. In another viewpoint, GBIF [25,26] suggested that the persistent identifiers of taxa should be unfriendly to read, and a taxon concept and a name should be presented separately, so the identifiers are still enduring while the names have changed. This idea is basically consistent with the normalized database design that eliminates the difficulty of updating data, but the data model is much more complex for accessing. In this research, we are more focusing on accessing linked data, but updating is less emphasized because the change in knowledge is recorded by appending a new revision. Working with the revision of knowledge, an identifier is not necessary to view as a persistent thing. This viewpoint leads that the design of data model is more relax than the use of persistent identifiers. Thus, it is possible to encapsulate a taxon concept and a name within a single identifier, and using a human-readable string in a URI is also possible to do. The simple representation comes with several advantages: lightweight data, recognizable URIs, and understandable linked data. Although it results in the slightly decrease of the information granularity, it brings about to the improvement of user satisfaction for contributing and consuming data. However, this model does not put restriction on the use of URIs, either separating a taxon concept and a name or using unreadable URIs is possible to implement.

6.2.2. Data Preparation

In this field, data are usually provided by domain experts especially taxonomists. The prototype provides form-based web application with text fields for user to input. It is good for small data in practice. However, when dealing with a large number of data, we recommend users to upload a text file containing event-centric models. Since this research does not aim at user experience design, in this phase, we encourage users

to understand the basic syntax of N-Triples of RDF. The data preparation steps are simply demonstrated as follows: (1) giving contextual nominal entities for every taxon with every change, (2) creating an event entity with a time interval and references, (3) creating instances of some proper operations for every changes, (4) assigning contextual nominal entities before and after the change, (5) giving links for causes and effects between some operations, (6) creating the representatives of external nominal entities for all taxa, (7) giving links between contextual nominal entities and the representatives of external nominal entities, (8) searching taxa from the Internet, and (9) giving links between the representatives and the external URIs. Since all operations are employed in similar ways and URIs are human-readable, non-computer-expert users are possible to create data and import them into the system. However, we learn that finding available URIs from known online datasets consumes much effort. In the future, we will consider some proper solutions to support this task and create a spreadsheet template for bulk upload.

6.3. System Integration

For the design of data model, apart from satisfying the present requirements, the viewpoints for framework enhancement and data exchange are discussed.

6.3.1. Extensibility

There are a lot of kinds of relationships available in taxonomic documents such as comprehensive relationships documented by TCS [47]. There are many minor relationships between names and concepts, but usually these relationships are summarized as valid (accepted), invalid (not valid but correctly proposed), and unavailable (neither valid nor correctly proposed). Some properties collected by TCS are such as is-homotypic-synonym-of, is-later-homonym-of, is-validation-of, is-vernacular-for, has-conserved-name, is-second-parent-for, is-hybrid-parent-of, etc. However, our present work is mainly focused on the changes in taxonomic knowledge with simple situations, and the introduction of more terms is a future challenge. In this case, our framework allows increasing the capability of a system with other vocabularies by creating operations under either the classes of the change in conception (*cka:ConceptEvolution*) or the change in triple

(*cka:RelationshipEvolution*), and reusing or adapting the semantic web rules.

6.3.2. Interoperability

Thanks to the progress of semantic web technology, current RDF repositories can maintain billions of data. However, in reality, it does not rely on a single data source. The integration among taxonomic information systems is able to do via the Internet using either web services or SPARQL endpoints together with commonly accepted data models.

6.4. Limitation

In this research, we assume that every change in taxonomy is clearly described. The representations of any changes are based on explicit evidence such as publication. In our experiment, before creating RDF data presenting the changes, the domain expert has to analyze the difference between several checklists, finding how names are different, and summarize into operations of changes. For this reason, the precision of the model relies on the completeness and the correctness of collected data. However, even existing references such as books and publications contain only insufficient information. For example, synonymic catalogue, also called synonym list, is a standard way in taxonomy to present historical summary of taxonomic studies on each species, including unaccepted names, misidentifications, references, etc. Here, a statement from the synonymic catalogue [39],

Adela Latreille, 1796
35. *reaumurella* (Linnaeus, 1958),
Syst. Nat. (Edn 10) 1:540 (*Phalaena*).
viridella (Scopoli, 1763), Ent. Carniolica: 250 (*Phalaena*).

is interpreted that the species *Phalaena viridella* is a synonym of the accepted species *Adela reaumurella*, but the reason behind this synonym is not available. There are many possible ways about the rejection of either *Phalaena* or *P. viridella*, while our model preferred only explicit fact to be recorded. In other words, our present approach is not designed for dealing with any incomplete and inconsistent data. Although our data model can document these kinds of data using contextual nominal entities as fragments of historical data; it cannot guarantee the precise interpretation of taxonomy if some linked fragments are disconnected or mistakenly connected. In this case, it needs a relax data model that can handle any implicit taxonomic knowledge and inspect correct knowledge from fuzzy explanation.

7. Conclusions

Our paper presents a logical model and ontology for linking concepts that comprise a series of changes, a diversity of taxonomic classifications, and a variety of scientific names. For the purpose of linking data, we developed our model by employing an ontology of contextual knowledge evolution together with widely accepted ontologies such as SKOS. A single and readable Internet resource for representing a version of concepts used in taxonomic knowledge can be viewed as either a name or a taxon concept. It results in that triples become lightweight, simple, and easy to understand by both machines and non-computer-expert users. Our model can deal with both complex format of the event-centric model and easily-linkable triples from the transition model and the snapshot model in RDF, and hence can trace the background knowledge of the given associated concepts. In addition, we implemented a prototype that utilizes the proposed model for managing the change in taxonomic knowledge and offering open access in order to give opportunity to link our data to LOD cloud. As a consequence, other applications that need linked concepts can readily connect to these data. By giving links to and reusing some existing URIs from well-known taxonomic databases, it is possible to associate our dataset to the large amount of taxonomic data across repositories in order to discover a broader knowledge of biology.

Our approach is mainly designed based on the test cases in zoology. Some requirements from these domains such as botany and mycology sometimes differ from zoology. Thus, some operations of changes, some configurations of property, and some semantic web rules have to be improved in order to satisfy the needs of those domains. Moreover, this approach provides features for managing change in taxonomic knowledge in RDF, however building the practical taxonomic information system requires some additional functionalities. In order to have a full functioned system, developers have to consider some further points. First, the RDF data can be contributed by many providers. To encourage non-computer-expert users to involve with the system, an application should have a good user experience design. Next, to have a proper data management, it needs high quality functions of authentication, authorization, and administration that can manage user privileges and access controls at the data level. The license of data is also properly declared. In addition, automate data matching is would be necessary to be provided, otherwise data providers have

to collect external URIs and link own contextual nominal entities with them in order to link to LOD cloud. Last, the data converter that can migrate other legacy datasets into the LTK model should be developed.

Acknowledgements

We thank Dr. Isao Nishiumi of the National Museum of Nature and Science and Dr. Takeshi Yamasaki of the Yamashina Institute for Ornithology, who provided us with valuable comments on the taxonomy of birds. We also thank the researchers from the Asian Institute of Technology, Chulalongkorn University, Kasetsart University, Thammasat University, the Consortium of International Agricultural Research Centers, and the National Institute for Agro-Environmental Science for the many valuable ideas that helped us improve our research. We appreciate the financial support from the National Institute of Informatics that fully funded our research activities.

Appendix A: Example Namespaces used by LTK

This section gives information about prefixes and namespaces used in this paper.

Prefix	Namespace
<i>bibo:</i>	Bibliographic Ontology [36] http://purl.org/ontology/bibo/
<i>cka:</i>	Contextual Knowledge for Archives [8] http://www.cka.org/2012/01/cka-onto#
<i>dct:</i>	Dublin Core Terms Namespace [35] http://purl.org/dc/terms/
<i>dbpedia:</i>	DBpedia Namespace [24] http://live.dbpedia.org/resource/
<i>dwc:</i>	Darwin Core [31] http://rs.tdwg.org/dwc/terms/
<i>foaf:</i>	Friend of a Friend Ontology [32] http://xmlns.com/foaf/0.1/
<i>gbif:</i>	Global Biodiversity Information Facility [27] http://www.gbif.org/species/
<i>genus:</i>	Namespace for genera used in LTK http://rc.lodac.nii.ac.jp/taxon/genus/
<i>lodac:</i>	LODAC Species [15] http://lod.ac/species/
<i>ltk:</i>	Linked Taxonomic Knowledge Ontology http://rc.lodac.nii.ac.jp/ns/ltk#
<i>skos:</i>	Simple Knowledge Organization System Namespace [29] http://www.w3.org/2004/02/skos/core#

<i>species:</i>	Namespace for species used in LTK http://rc.lodac.nii.ac.jp/taxon/species/
<i>soic:</i>	Semantically-Interlinked Online Communities Core Ontology [34] http://rdfs.org/soic/ns#
<i>tl:</i>	Timeline Ontology [33] http://purl.org/NET/c4dm/timeline.owl#
<i>tmo:</i>	Meta-Ontology of Biological Name [6] http://www.yso.fi/onto/taxmeon/

Appendix B: Example LTK Properties

This section gathers some properties provided by LTK.

Properties	rdfs:subPropertyOf
<i>ltk:higherTaxon</i>	<i>cka:higherClass</i> , <i>skos:broaderTransitive</i> , <i>tmo:isPartOfHigherTaxon</i> , and <i>lodac:hasSuperTaxon</i>
<i>ltk:replacedTo</i>	<i>cka:serialLinkTo</i> , <i>tmo:congruentWithTaxon</i> , and <i>skos:exactMatch</i>
<i>ltk:mergedInto</i>	<i>cka:serialLinkTo</i> , and <i>skos:broadMatch</i>
<i>ltk:majorMergedInto</i>	<i>cka:serialLinkTo</i> , and <i>skos:closeMatch</i>
<i>ltk:splitInto</i>	<i>cka:serialLinkTo</i> , and <i>skos:narrowMatch</i>
<i>ltk:majorSplitInto</i>	<i>cka:serialLinkTo</i> , and <i>skos:closeMatch</i>
<i>ltk:dsynonym</i>	<i>skos:exactMatch</i> , and <i>lodac:hasSynonym</i>
<i>ltk:synonym</i>	<i>skos:exactMatch</i> , <i>ltk:dsynonym</i> , and <i>lodac:hasSynonym</i>
<i>ltk:cpxChangedTo</i>	<i>skos:relatedMatch</i>
<i>ltk:circChangedTo</i>	<i>skos:closeMatch</i>
<i>ltk:subdividedInto</i>	<i>skos:narrowMatch</i>
<i>ltk:combinedInto</i>	<i>skos:broadMatch</i>

Appendix C: Example LTK Operations

Operations provided by LTK ontology and their parameters have been declared in the following list. An italic symbol in the parentheses of each parameter indicates its cardinality for every operation. The symbol “(1)” allows only one value, the symbol “(2..*)” expects at least two values required, and the symbol “(0..1)” presents one optional value.

• ltk:TaxonMerger

Description	For merging some concepts (before) into one concept (after).
Parameters	<ul style="list-style-type: none"> • <i>cka:conceptBefore</i> ^(2..*) • <i>cka:majorConceptBefore</i> ^(0..1) • <i>cka:conceptAfter</i> ⁽¹⁾
Example input RDF	<pre>ex:opr rdf:type ltk:TaxonMerger . ex:opr cka:conceptBefore ex:be1, ex:be2 ; cka:majorConceptBefore ex:mb0 ; cka:conceptAfter ex:af1 .</pre>
Example result	<pre>ex:be1 ltk:mergedInto ex:af1 . ex:be2 ltk:mergedInto ex:af1 . ex:mb0 ltk:majorMergedInto ex:af1 .</pre>
Example entailment	<pre>ex:be1 skos:broadMatch ex:af1 . ex:be2 skos:broadMatch ex:af1 . ex:mb0 skos:closeMatch ex:af1 .</pre>

• ltk:TaxonSplitter

Description	For splitting a concept (before) into new concepts (after).
Parameters	<ul style="list-style-type: none"> • <i>cka:conceptBefore</i> ⁽¹⁾ • <i>cka:conceptAfter</i> ^(2..*) • <i>cka:majorConceptAfter</i> ^(0..1)
Example input RDF	<pre>ex:opr rdf:type ltk:TaxonSplitter . ex:opr cka:conceptBefore ex:be1 ; cka:conceptAfter ex:af1, ex:af2 ; cka:majorConceptAfter ex:ma0 .</pre>
Example result	<pre>ex:be1 ltk:splitInto ex:af1 . ex:be1 ltk:splitInto ex:af2 . ex:be1 ltk:majorSplitInto ex:ma0 .</pre>
Example entailment	<pre>ex:be1 skos:narrowMatch ex:af1 . ex:be1 skos:narrowMatch ex:af2 . ex:be1 skos:closeMatch ex:ma0 .</pre>

• ltk:TaxonReplacement

Description	For replacing one concept (before) to another one (after).
Parameters	<ul style="list-style-type: none"> • <i>cka:conceptBefore</i> ⁽¹⁾ • <i>cka:conceptAfter</i> ⁽¹⁾
Example input RDF	<pre>ex:opr rdf:type ltk:TaxonReplacement . ex:opr cka:conceptBefore ex:be1 ; cka:conceptAfter ex:af1 .</pre>

Example result	ex:be1 ltk:replacedTo ex:af1 .
Example entailment	ex:be1 skos:exactMatch ex:af1 . ex:be1 tmo:congruentWithTaxon ex:af1 .

• ltk:TaxonComplexChange

Description	For a complex case that many concepts (before) are merged and split into many other concepts (after).
Parameters	<ul style="list-style-type: none"> • cka:conceptBefore ^(2..*) • cka:conceptAfter ^(2..*)
Example input RDF	<pre>ex:opr rdf:type ltk:TaxonComplexChange . ex:opr cka:conceptBefore ex:be1, ex:be2 ; cka:conceptAfter ex:af1, ex:af2 .</pre>
Example result	<pre>ex:be1 ltk:cpxChangedTo ex:af1 . ex:be1 ltk:cpxChangedTo ex:af2 . ex:be2 ltk:cpxChangedTo ex:af1 . ex:be2 ltk:cpxChangedTo ex:af2 .</pre>
Example entailment	<pre>ex:be1 skos:relatedMatch ex:af1 . ex:be1 skos:relatedMatch ex:af2 . ex:be2 skos:relatedMatch ex:af1 . ex:be2 skos:relatedMatch ex:af2 .</pre>

• ltk:CircumscriptionChange

Description	For changing circumscription of one concept (before) to another one (after).
Parameters	<ul style="list-style-type: none"> • cka:conceptBefore ⁽¹⁾ • cka:conceptAfter ⁽¹⁾
Example input RDF	<pre>ex:opr rdf:type ltk:CircumscriptionChange . ex:opr cka:conceptBefore ex:be1 ; cka:conceptAfter ex:af1 .</pre>
Example result	ex:be1 ltk:circChangedTo ex:af1 .
Example entailment	ex:be1 skos:closeMatch ex:af1 .

• ltk:ChangeHigherTaxon

Description	For reclassifying a lower concept (child) by moving from a higher concept (before) to another one (after).
-------------	------------------------------------------------------------------------------------------------------------

Parameters	<ul style="list-style-type: none"> • cka:child ⁽¹⁾ • cka:parentBefore ⁽¹⁾ • cka:parentAfter ⁽¹⁾
Example input RDF	<pre>ex:opr rdf:type ltk:ChangeHigherTaxon . ex:opr cka:child ex:c1 ; cka:parentBefore ex:p1 ; cka:parentAfter ex:p2 .</pre>
Example result	<pre>ex:c1 ltk:higherTaxon ex:p2 . ex:p2 ltk:lowerTaxon ex:c1 .</pre>
Example entailment	<pre>ex:c1 skos:broaderTransitive ex:p2 . ex:p2 skos:narrowerTransitive ex:c1 . ex:c1 lodac:hasSuperTaxon ex:p2 .</pre>

• ltk:SubdivideTaxon

Description	For subdividing a higher taxon (source) into some lower taxa (target).
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ⁽¹⁾ • cka:targetConcept ^(2..*)
Example input RDF	<pre>ex:opr rdf:type ltk:SubdivideTaxon . ex:opr cka:sourceConcept ex:h1 ; cka:targetConcept ex:c1, ex:c2 .</pre>
Example result	<pre>ex:h1 ltk:subdividedInto ex:c1 . ex:h1 ltk:subdividedInto ex:c2 .</pre>
Example entailment	<pre>ex:h1 skos:narrowMatch ex:c1 . ex:h1 skos:narrowMatch ex:c2 .</pre>

• ltk:CombineTaxa

Description	For combining lower taxa (source) into a higher taxon (target).
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ^(2..*) • cka:targetConcept ⁽¹⁾
Example input RDF	<pre>ex:opr rdf:type ltk:CombineTaxa . ex:opr cka:sourceConcept ex:c1, ex:c2 ; cka:targetConcept ex:h1 .</pre>
Example result	<pre>ex:c1 ltk:combinedInto ex:h1 . ex:c2 ltk:combinedInto ex:h1 .</pre>
Example entailment	<pre>ex:c1 skos:broadMatch ex:h1 . ex:c2 skos:broadMatch ex:h1 .</pre>

• **ltk:DirectSynonymLink**

Description	For identifying a synonym (target) of a concept (source). It is a directional synonym, which is always used in botany.
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ⁽¹⁾ • cka:targetConcept ⁽¹⁾
Example input RDF	ex:opr rdf:type ltk:DirectSynonymLink ex:opr cka:sourceConcept ex:c1 ; cka:targetConcept ex:c2 .
Example result	ex:c1 ltk:dsynonym ex:c2 .
Example entailment	ex:c1 skos:exactMatch ex:c2 . ex:c2 skos:exactMatch ex:c1 . ex:c1 lodac:hasSynonym ex:c2 .

• **ltk:SynonymLink**

Description	For identifying a synonym (target) of a concept (source). It is a bidirectional synonym, which is generally used in many domains especially in zoology.
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ⁽¹⁾ • cka:targetConcept ⁽¹⁾
Example input RDF	ex:opr rdf:type ltk:SynonymLink. ex:opr cka:sourceConcept ex:c1 ; cka:targetConcept ex:c2 .
Example result	ex:c1 ltk:synonym ex:c2 .
Example entailment	ex:c2 ltk:synonym ex:c1 . ex:c1 ltk:dsynonym ex:c2 . ex:c2 ltk:dsynonym ex:c1 . ex:c1 skos:exactMatch ex:c2 . ex:c2 skos:exactMatch ex:c1 . ex:c1 lodac:hasSynonym ex:c2 . ex:c2 lodac:hasSynonym ex:c1 .

• **ltk:SeniorSynonymLink**

Description	For identifying a senior synonym (target) of a concept (source).
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ⁽¹⁾ • cka:targetConcept ⁽¹⁾
Example input RDF	ex:opr rdf:type ltk:SeniorSynonymLink. ex:opr cka:sourceConcept ex:c1 ; cka:targetConcept ex:c2 .

Example result	ex:c1 ltk:seniorSynonym ex:c2 . ex:c2 ltk:juniorSynonym ex:c1 .
Example entailment	ex:c1 ltk:synonym ex:c2 . ex:c2 ltk:synonym ex:c1 . ex:c1 skos:exactMatch ex:c2 . ex:c2 skos:exactMatch ex:c1 . ex:c1 lodac:hasSynonym ex:c2 . ex:c2 lodac:hasSynonym ex:c1 .

• **ltk:HomonymLink**

Description	For identifying a homonym (target) of a concept (source).
Parameters	<ul style="list-style-type: none"> • cka:sourceConcept ⁽¹⁾ • cka:targetConcept ⁽¹⁾
Example input RDF	ex:opr rdf:type ltk:HomonymLink. ex:opr cka:sourceConcept ex:c1 ; cka:targetConcept ex:c2 .
Example result	ex:c1 ltk:homonym ex:c2 . ex:c2 ltk:homonym ex:c1 .

Appendix D: Example part of LTK Ontology

This section shows an example part of LTK ontology that deal with the transition model. The hierarchy and type of properties are defined as follows:

```

ltk:majorMergedInto
  rdfs:subPropertyOf ltk:mergedInto .

ltk:majorSplitInto
  rdfs:subPropertyOf ltk:splitInto .

ltk:mergedInto
  rdfs:subPropertyOf cka:serialLinkTo .

ltk:splitInto
  rdfs:subPropertyOf cka:serialLinkTo .

ltk:replacedTo
  rdfs:subPropertyOf cka:serialLinkTo .

ltk:serialLinkTo
  rdf:type owl:TransitiveProperty ;
  rdfs:subPropertyOf cka:semanticLink .

ltk:semanticLink
  rdf:type owl:TransitiveProperty ,
           owl:SymmetricProperty .

```

References

- [1] Linnaeus C: *Systema Naturae*, 10th Edition, Holmiae; 1758.
- [2] Winston JE: *Describing Species: Practical Taxonomic Procedure for Biologists*. New York: Columbia University Press; 1999.
- [3] Wink M, Heidrich P : Molecular evolution and systematics of the owls (Strigiformes). In *A Guide to Owls of the World*. Edited by König C, Becking JH. Yale: Yale University Press; 1999: 39-57.
- [4] Mallet J: Species, Concept of. In *Encyclopedia of Biodiversity*. Volume 5. Edited by Levin SA, Academic Press; 2001: 427-440.
- [5] Ytow N , Morse D , Roberts D: Nomenclator: a nomenclatural history model to handle multiple taxonomic views. *Biol J Linnean Soc*; 2001: 73.1: 81-98.
- [6] Tuominen J, Laurenne N, Hyvönen E: Biological names and taxonomies on the semantic web: managing the change in scientific conception. In *Proceedings of the 8th Extended Semantic Web Conference 29 May - 2 June 2011; Greece*. Edited by Antoniou G, et al; 2011: 255-269.
- [7] Heath T, Bizer C: Linked Data: Evolving the Web into a Global Data Space. In *Synthesis Lectures on the Semantic Web: Theory and Technology*. Edited by Hendler J, Harmelen FV. Morgan & Claypool Publishers; 2011
- [8] Chawuthai R, Wuwongse V, Takeda H: A Formal Approach to the Modelling of Digital Archives. In *The Outreach of Digital Libraries: A Globalized Resource Network*. Edited by Chen HH, Chowdhury G. Heidelberg: Springer; 2012: 179-188.
- [9] Flouris G, Meghini C: Terminology and Wish List for a Formal Theory of Preservation. In *Proceedings of the PV2007 International Conference 9-11 October 2007*; 2007
- [10] Gutierrez C, Hurtado C, Vaisman R: Temporal RDF. In *The Semantic Web: Research and Applications*. Edited by Aroyo L, et al: Heidelberg: Springer; 2005: 93-107.
- [11] Franz N, Peet R: Towards a language for mapping relationships among taxonomic concepts. In *Systematics and Biodiversity*. Volume 7.1. Edited by Rosen B: Taylor&Francis; 2009:5-20.
- [12] Banks RC, Cicero C, et al: Forty-fourth supplement to the American Ornithologists' Union checklist of North American birds. In *The Auk* Volume. 120.3: The American Ornithologists' Union; 2003: 923-931.
- [13] Sibley CG, and Short LLJ. 1964. Hybridization in the Orioles of the Great Plains. In *The Condor*. Volume 66.2: JSTOR; 1964: 130-150.
- [14] Freeman S, Zink RM: A phylogenetic study of the blackbirds based on variation in mitochondrial DNA restriction sites. *Syst Biol* 1995, 44: 409-420.
- [15] Minami Y, Takeda H, Kato F, et al: Towards a Data Hub for Biodiversity with LOD. In *Proceedings of the Second Joint International Conference: 2-4 December 2012; Japan*. Edited by Takeda H, Qu Y, Mizoguchi R, et al: Springer; 2013: 356-361.
- [16] Berendsohn WG: A taxonomic information model for botanical databases: the IOPI Model. *Taxon*; 1997: 46:283-309.
- [17] Page RDM: Taxonomic names, metadata, and the Semantic Web. In *Biodiversity Informatics 2006*, 3:1-15.
- [18] Jones AC, White RJ, Orme ER: Identifying and relating biological concepts in the Catalogue of Life. *Journal of Biomedical Semantics*; 2011; 2:7.
- [19] Sarkar IN: Biodiversity informatics: organizing and linking information across the spectrum of life. *Briefings in Bioinformatics*; 2007: 8(5):347-357.
- [20] Schulz S, Stenzhorn H, Boeker M: The ontology of biological taxa. *Bioinformatics*; 2008: 24(13):i313-i321.
- [21] Kennedy J, Kukla R, Paterson T: Scientific Names Are Ambiguous as Identifiers for Biological Taxa: Their Context and Definition Are Required for Accurate Data Integration. In *The 2nd International Conference on Data Integration in the Life Sciences (DILS): 20-22 July 2005; San Diego, California*. Edited by Ludascher B, Raschid L, Springer-Verlag; 2005: 80-95.
- [22] Laurenne N, Tuominen J, Saarenmaa H, Hyvönen E: Making species checklists understandable to machines—a shift from relational databases to ontologies. *Journal of Biomedical Semantics*; 2014: 5(1), 40.
- [23] Patterson DJ, Cooper J, et al: In Names are key to the big new biology. *Trends in ecology & evolution*; 2004: 25(12), 686-691.
- [24] Lehmann J, Isele R, et al: DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. In *Semantic Web Journal*; 2014.
- [25] Richards K, White R, Nicolson N, Pyle R: A beginner's guide to persistent identifiers. In *Tech*

- Rep Global Biodiversity Information Facility (GBIF), Copenhagen, Denmark; 2011.
- [26] Cryer P, Hyam R, Miller C, et al: Adoption of persistent identifiers for biodiversity informatics: Recommendations of the GBIF LSID GUID task group, 6. November 2009. In Tech Rep Global Biodiversity Information Facility (GBIF), Copenhagen, Denmark; 2010.
- [27] The Global Biodiversity Information Facility (GBIF) [<http://www.gbif.org>]
- [28] Biodiversity Information Standards (TDWG) [<http://www.tdwg.org>]
- [29] Simple Knowledge Organization System [<http://www.w3.org/TR/skos-primer/>]
- [30] Linked Open Data [<http://linkeddata.org/>]
- [31] Darwin Core [<http://rs.tdwg.org/dwc/terms/>]
- [32] Friend of a Friend [<http://xmlns.com/foaf/0.1/>]
- [33] The Timeline Ontology [<http://motools.sourceforge.net/timeline/>]
- [34] Semantically-Interlinked Online Communities Core Specification [<http://www.w3.org/Submission/sioc-spec/>]
- [35] Dublin Core Metadata Initiative Terms [<http://dublincore.org/documents/dcmiterms/>]
- [36] Bibliographic Ontology [<http://bibliontology.com>]
- [37] Apache Jena [<http://jena.apache.org/>]
- [38] OpenRDF [<http://www.openrdf.org>]
- [39] Inoue H, Sugi S, et al: Moths of Japan, Volume 2: Plates and synonymic catalogue. Kodensha Co. Ltd, Tokyo; 1982.
- [40] Jinbo U: List-MJ: A checklist of Japanese moths 2004-2008. [<http://listmj.mothprog.com>]
- [41] Kishida Y: The Standard of Moths in Japan II. Gakken Education Publishing; 2011.
- [42] Taxonomic Names and Concepts Interest Group: Taxonomic concept transfer schema; 2005. [<http://www.tdwg.org/standards/117/>]
- [43] Richard LP, Ellinor M: ZooBank: Developing a nomenclatural tool for unifying 250 years of biological information; 2008.
- [44] Crous PW, Gams W, et al: MycoBank: an online initiative to launch mycology into the 21st century. Studies in Mycology; 2004.
- [45] International Commission on Zoological Nomenclature. International Code of Zoological Nomenclature. 4th Edition. The International Trust for Zoological Nomenclature, London, UK, pp 306; 1999.
- [46] Remsen DP, Döring M, Robertson T. Copenhagen: GBIF GNA Profile Reference Guide for Darwin Core Archives, version 1.2; 2011.
- [47] Taxonomic Concept Schema Complementary Documentation for Draft Standard. [http://tdwg.napier.ac.uk/doc/tdwg_tcs.doc]
- [48] Describing Taxon Concepts as RDF (draft) [<https://code.google.com/p/tdwg-rdf/wiki/TaxonInRDF>]