A Rule-Based Citation System for Structured and Evolving Datasets

Peter Buneman
University of Edinburgh
opb@inf.ed.ac.uk

Gianmaria Silvello*
University of Padua
silvello@dei.unipd.it

Abstract

We consider the requirements that a citation system must fulfill in order to cite structured and evolving data sets. Such a system must take into account variable granularity, context and the temporal dimension. We look at two examples and discuss the possible forms of citation to these data sets. We also describe a rule-based system that generates citations which fulfill these requirements.

1 Motivation

Citations are one of the most significant tools used in the creation and propagation of knowledge; through citations we can give credit or attribution to researchers, convey a brief indication of the contents (such as a title), and provide location information for retrieval of the referenced work. Over the years, a well-developed system (with minor variations) has evolved for the citation of sources such as books, journals and scholarly papers. On the other hand, there is no well-defined mechanism for publishing, accessing and citing datasets [9, 1, 8]. Datasets are difficult to catalog [8] and, consequently, they are difficult to cite, retrieve and reuse. We should note that publishing datasets that are subsequently cited is beneficial for scientists, who may receive credit for their work of collecting and assuring quality of the dataset and for organizations such as digital libraries or funding bodies that want to build accessible data collections for the benefit of designated communities [6].

We start with the observation that a citation is more than a persistent identifier such as a URI or *Digital Object Identifier (DOI)*[1] which allow us to *locate* the data of interest; citations also carry certain kinds of metadata such as a title, authorship, date – information that is immediately useful in judging the quality or currency of the cited information. Moreover, citations are increasingly used to judge the reputation of a work or person. We believe that these functions of traditional citation should carry over to data citation, but in order to pursue this idea we need to make the following assumptions: (a) that there is an accepted system of persistent identifiers, (b) that the referenced material does not change and (c) that it is hierarchical in structure. All three of these require some consideration. For (a) there is still no universally accepted system. Many data sets change in time, and old versions are not always available in order to satisfy (b). While (c) is required for the mechanisms we propose

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

*Work of this author was performed at the University of Edinburgh, while visiting from the University of Padua.

[1] http://www.doi.org/ Note that this is neither a persistent identifier nor a citation.

and is usually true for scientific data sets, it is not clear how it applies to ontologies, something that we briefly discuss at the end of this paper.

In a conventional citation, information such as ISBN, journal number, page number etc. serve uniquely to identify a work and also may provide some help in retrieving the work (e.g. once you are inside the library.) A persistent identifier such as the DOI provides unique identification and may also be associated with a means of retrieval such as a URL. In order to fulfill the requirements of a conventional citation we clearly have to associate further information with the DOI. While it is common practice to do this for digital objects, there are no conventions or standards and, most importantly for data citation, we are typically citing a part of some larger collection. Here neither the location information within the collection nor the metadata associated with the cited component can be carried with the DOI. In fact, we believe the situation is the reverse. The DOI should be a part of the citation.

We consider the citation issues related to the field of Digital Libraries and of Curated Databases. In the first case we consider how to cite digital archives described by means of the international standard for archival description which is the *Encoded Archival Description (EAD)*[2]. For the latter case we consider a widely used curated database which is the IUPHAR DB[3] incorporating detailed pharmacological, functional and pathophysiological information.

In this paper we present the requirements that a citation system must fulfill in order to guarantee the consistency and the integrity of citations. Furthermore, we take into account the concept of *citable unit*, explaining how it is adopted in the citation practice of traditional publications and how it can be exploited in the citation of structured and evolving datasets. We show how this concept is implicitly adopted by different citation systems based on the use of unique identifiers. We present the advantages but also some issues of these citation systems; in particular we put in relation the EAD files and the IUPHAR DB with these citation systems pointing out where a new solution is necessary to overcome some citation issues. The final section of this paper presents the rule-based citation system that we have designed and developed[4], showing how it addresses the presented issues and how it can be further extended.

2 Two examples: EAD and the IUPHAR Database

Of our examples, the EAD represents a widely-adopted standard to represent and describe archival resources; IUPHAR is a curated database that resembles a conventional publications such as a reference manual in that it represents the work of a large number of experts who both create and revise its content. These two datasets are quite different in their nature and use, but they both have a hierarchical structure that can be exploited for the definition of an automatic citation system. Furthermore, they represent two different means of representing and disseminating the information which has the same citation issues and that can benefit from the solution we propose.

In order to understand the characteristics of the **EAD** file we need to introduce some basic concepts about archives. An archive is a complex organization which is not just a series of (digital) objects that have been accumulated over time. Archives have to keep the context in which their documents have been created and the network of relationships among them in order to preserve their informative content and provide understandable and useful information over time. This is achieved through a richly annotated hierarchical classification system.

In a digital archive the components are described by the use of metadata that can express and maintain this structure and relationships. The standard format of metadata for representing the complex hierarchical structure of the archive is EAD [11], which reflects the archival structure and relationships between documents in the archive. To maintain all this information an EAD file turns out to be a large *eXtensible Markup Language (XML)*

[2] `http://www.loc.gov/ead/`

[3] `http://www.iuphar-db.org/`

[4] The Java API can be found here: `http://www.dei.unipd.it/~silvello/datacitation/`

file with a deep hierarchical internal structure. In Figure 1 we can see how the XML structure of an EAD file resembles the structure of the archive[5].

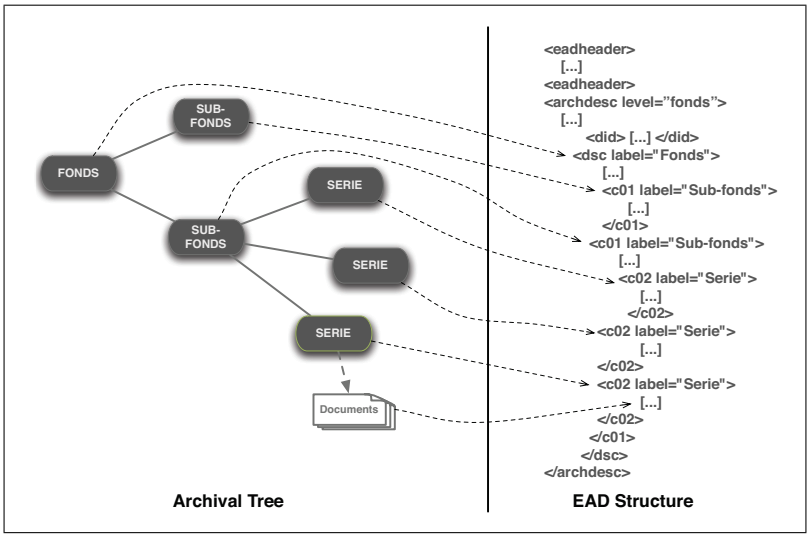

Figure 1: The hierarchical structure of an archive mapped into an EAD file.

We can see that every node in the archival hierarchy is mapped in a specific tag of the EAD XML file, where: The `<eadheader>` contains metadata about the archive descriptions and includes information about them such as title, author and date of creation. The `<archdesc>` contains the archival description itself and constitutes the core of EAD; it may include many high-level sub-elements, most of which are repeatable. The most important element is the `<did>` or descriptive identification, which describes the collection as a whole; it is composed of numerous sub-elements that are intended for brief, clearly designated statements of information and they are available at every level of description. Finally, the `<archdesc>` contains an element that facilitates a detailed analysis of the components of a fond, the `<dsc>` or description subordinate components. The `<dsc>` contains a repeatable recursive element, called `<c>` or component. A component may be an easily recognizable archival entity such as series, subseries or items. Components are not only nested under the `<archdesc>` element, they usually are nested inside one another and they are indicated with `<c...>` tag.

The IUPHAR database incorporates detailed pharmacological, functional and pathophysiological information on G Protein-Coupled Receptors, Voltage-Gated and Ligand-Gated Ion Channels. Although the database is internally represented as a relational database, its structure as see by both the users and the contributors is essentially hierarchical, where the root is the database as a whole and it has the list of receptor families as children nodes; each receptor family node has the receptors as children nodes. Each receptor is described by a Web page containing the main technical information; these Web pages contain information such as lists of "agonists", "antagonists" for each receptor.

3 Requirements of a Citation System

In order to define the requirements that a citation system for structured and evolving datasets has to satisfy, we need to identify which are the "citable" elements in a dataset. To accomplish this purpose we have to introduce an important concept, i.e. the "citable unit". The concept of citable unit has been used in general terms to indicate the target of a citation such as a book, a chapter, a journal or a paper [10]. In [2] the concept of citable

[5]For the uninitiated a *fond* is a term derived from French that describes the collection of records in an archive that originate from the same creator. Please see: `http://www.archivists.org/glossary/term_details.asp?DefinitionKey=196`

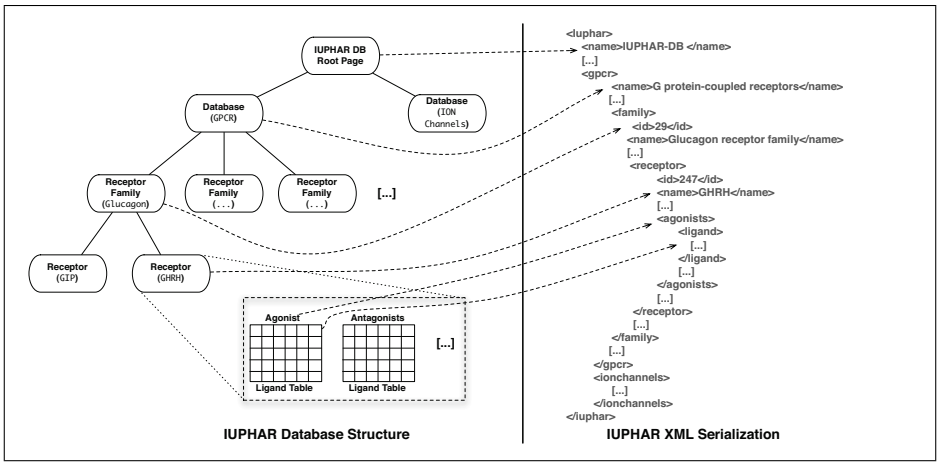

Figure 2: The structure of the IUPHAR Database and its XML serialization.

unit is exploited in the law documentation context to indicate each element of an "Act of Parliament" that must be unambiguously identified. While the object of interest may be a simple fact or a single number, the citable unit is the context in which that object occurs. Note that what constitutes a citable unit is usually understood: a book or an article, but there are cases in which it may not be clear. For example, in a book which is a tightly integrated collection of chapters by different authors, is the citable unit the book or the chapter; or should both be allowed?

For the hierarchical structures we are considering we believe that it should be possible to cite nodes at various levels in the database. For example, in IUPHAR, the citation for "The IUPHAR database does not contain X" is the whole database, however for "Y is an agonist for X it is the section on receptor X (the authorship for each family of receptors is different). Thus it should be possible to have one citable unit contained in another. Furthermore, we believe that the onus is on the *publishers* of the data to define the citable units.

Most of the data citation proposals [9, 1, 8] are what we call "identifier-based" – they start with the idea of assigning a unique identifier (e.g. a DOI) to the objects that need to be independently cited by and by associating appropriate metadata for that object. The main differences between these proposals are in the forms of metadata to be associated with the identifier. These solutions implicitly adopt the concept of citable unit as a foundational idea; indeed, they identify each citable unit with a unique identifier and then use the associated metadata for citations. They appear not to take into account the hierarchical nature of the objects being cited. It should also be noted that citable units can be very large (a whole book or database) and that it is often useful to have a means for locating a given "small" element of information – a fact or figure – within that unit. Thus some location system may be needed for this, but as we shall see, this does not necessitate an independent global digital identifier for each such element.

Let us see how these *identifier-based* solutions behave with the presented datasets; we can point out two main requirements that a citation system has to support: deep citations and temporal invariance. Deep citations can be divided into two parts: variable granularity and context. Each section in an EAD file may be identified by means of a unique identifier with associated metadata. At the same time we may want to cite a particular document contained in a node of the archive (e.g. a document contained in a series); in this case we need to have an identifier and a metadata for each document described in the EAD file and not only for the main parts representing the archival divisions. If we consider the IUPHAR database, as we have just seen there are good reasons for making nodes at different levels available as citable units. Moreover the metadata associated with each unit will change: the "authorship" of the whole database is different from that of its sub-units; the "title" will be different; and the currency (the date of last modification) may also differ.

Even though the number of citable units in a hierarchy may be limited, we may want – in a citation –

to provide information about a node that is not a citable unit, but is part of one. Typically this is location information for that node *within* the citable unit. When a citation is used in the form "In C it is stated that X", it is often useful to have some further information, e.g., "page 305", about where in X is located in C; but we do not want to use X (or its location) itself as the citable unit. This is would be use out of context. The same is true of the data sets we have been discussing: a single data value is not itself a citable unit, but having its location within a citable unit can be enormously useful for verification.

The second requirement is that *once a dataset has been cited, it must remain stable and always accessible in the cited form*; this involves the **temporal issues** connected with evolving datasets such as curated databases. In order to address this issue a reasonable solution is to employ a versioning system that permits us to access a specific version of a dataset or of a component of a dataset. In any case, we have to take into account that the adoption of a versioning system is not a one-size-fits-all solution; for instance, with highly dynamic datasets the volume of changes can be so large or frequent to make tracking back difficult to manage [8]. The temporal requirement is particularly relevant in the case of the IUPHAR DB because it changes quite often; indeed, the information about a receptor may be updated or new families can be added, some errors can be corrected etc. In the case of EAD files this is less frequent because archival information is relatively stable. For this reason we present our citation system starting from its use with the EAD files and then with the IUPHAR database which requires additional work to deal with the evolution of the data set.

4 A Rule-Based Citation System

We now describe a simple rule-based system for the automatic generation of citations directly from the data. We believe this is important not only for consistency and accuracy, but also for efficiency. It may be better to generate citations "on the fly" than to store them in the database. The system was originally sketched in in [3] but we have added important extensions that provide location information of the citable units themselves and of nodes within the citable units.

In the current system, those nodes that correspond to citable units are tagged with a rule (or reference to a rule) that can be used to generate citations. For EAD we exploited the `<prefercite>` that is intended to contain a textual citation. In order to avoid any conflicts we specify an attribute (`<prefercite type="citationRule">`) that allows us to distinguish between an element used for common textual citation and the element containing a citation rule that will be interpreted by the citation system. In the IUPHAR XML serialization we use a `<citation>` tag with no attribute because there is no risk of ambiguity with other elements. Another possibility would be not to modify the XML hierarchy but to use XPATH that can be generated from the rules themselves to describe the citable units, but this introduces some complications and we leave it for later work.

The form of a rule suggested in [3] is $C \leftarrow P$ where C provides the concrete syntax of a human or machine-readable citation and P is a pattern that is expressed in the syntax of XPath augmented with decorated variables. The purpose of P is not, as in XPath, to identify nodes, but to bind these variables that can then be used in C. The syntax of C is not important here: there are literally hundreds of conventions for human-readable citations [6] and a good number of machine-readable forms. For illustration we shall use a simple name-value pair syntax: $\{a_1 = \$x_1, \ldots, a_n = \$x_n\}$ in which the variables $\$x_1, \ldots, \$x_n$ will be instantiated by the rule to produce, for example, {Author="Smith", Title="...", ...}.

The general form of a pattern is $P = /t_1[p_1^1 = \$x_1^1, \ldots, p_1^{k_1} = \$x_1^{k_1}]/\ldots/t_n[p_n^1 = \$x_n^1, \ldots, p_n^{k_n} = \$x_n^{k_n}]$ in which the t_i are tag names and the p_i^k are fully specified downward paths consisting of a sequence of tag names. The variables $\$x_1^1, \ldots, \$x_1^{k_n}, \ldots, \$x_n^1, \ldots, \$x_n^{k_n}$ are all distinct. Although this follows the syntax of XPath, its purpose is different. XPath identifies nodes in an XML tree. This pattern assumes that the node is given, and it binds values – XML fragments that are typically character strings – to these variables. Of course

[6] `http://citationstyles.org/styles` claims more than a thousand

we expect the given node to be specified by the XPath expression $/t_1/\ldots/t_n$, but this is where our first kind of variable plays a role. *Key* variables, denoted by $\$'$ are there to specify location information. If we remove all other variables from a pattern, to leave a pattern of the form $/t_1[p_1^1 = \$'x_1^1, p_1^2 = \$'x_1^2 \ldots]/t_2[\ldots]/\ldots$ we expect there to be a unique binding v_i^j for each variable $'x_i^j$. That is, each of the paths p_i^j is unique from its context node. More importantly, after substitution of these values, the XPath expression $/t_1[p_1^1 = v_1^1, p_1^2 = v_1^2 \ldots]/t_2[\ldots]/\ldots$ *uniquely* specifies the node at which the citation rule is attached. In other words, these bindings, together with the citation rule, provide location information[7].

It is assumed that the bindings of the key variables will be available and possibly expressed as an XPath expression in the citation C. We now describe to other useful bindings of variables. While for key variables we expect the paths p_i^j to specify a node uniquely, in general such a path will specify a *set* of nodes. If we take document order, there is a list V_i^j of XML values associated with each such path. The following decorations are constraints on V that we expect to be useful in generating citations.

- $\$.x_i^j$: $|V_i^j| = 1$. The path p_i^j is unique, and $.x_i^j$ is bound to the single member of V_i^j. One might, for example, insist on a unique title that is to be expressed in the citation. Note that there is no implication that a $\$.$ variable will serve as a key.
- $\$?x_i^j$: $|V_i^j| \leq 1$. Here, $?x_i^j$ is bound to the unique member of V_i^j or to some designated null value if V_i^j is empty.
- $\$ * x_i^j$: No constraints; $*x_i^j$ is bound to the list V_i^j. A possibly empty list of keywords, for example.
- $\$ + x_i^j$: $|V_i^j| \geq 1$. Again, $+x_i^j$ is bound to the non-empty list V_i^j. A non-empty list of authors for example.

With a little cunning these constraints can be checked in a single scan of the XML tree. The time taken to generate a citation will depend on the physical representation of the XML; in the worst case it can again be done with a linear scan through the whole document, but it can obviously be performed more efficiently if, for example, the document is represented as a main-memory tree. As a first example of the citation system at work, in EAD we considered each element representing a node in the archival hierarchy to be a citable unit. In Figure 3 we can see a set of rules that generates a machine-readable citation and an example of what it generates; the input of the system is the XML node representing the citable unit to be cited.

We can see that we exploit the hierarchical structure of the EAD file to build a citation and we use only the information already present in the document. When we have decided which archival element we need to cite, the system starts creating the XPaths from the rule corresponding to that element and then going up generating absolute XPaths from the rules of each ancestor element. The citation is created by executing the conjunction of the recursively generated XPaths. Finally, from the citation we can reconstruct the path from the document root to the cited element reversing the mechanism (i.e. $C \rightarrow P$). By means of this procedure we do not need any further metadata specifically created for the data citation. Furthermore, the system can be easily customized because the citation rules can be defined and tuned by the publishers who decide which information should be presented in the citation and how the citation should be defined.

Figure 4 is taken from IUPHAR and shows an example of how different rules are invoked for citable units at two levels of granularity. In contrast to EAD, in IUPHAR, we may want to cite nodes that are not citable units. The specification of a citation is exactly as before, with a pattern of the general form: $/t_1[p_1^1 = \$x_1^1, \ldots, p_1^{k_1} = \$x_1^{k_1}]/\ldots/t_n[p_n^1 = \$x_n^1, \ldots, p_n^{k_n} = \$x_n^{k_n}]$ which must include at least enough key ($\$'x$) variables to specify a key for that node, precisely as was required before. The system now searches up the path to the root to find the lowest citable unit and combines (with appropriate renaming of variables) the two patterns. Note that on the key variables, the two patterns must agree at any node that occurs at or above the citable unit. Now C will contain

[7] In the presence of a key constraint [4] this check could be performed against the schema rather than the data

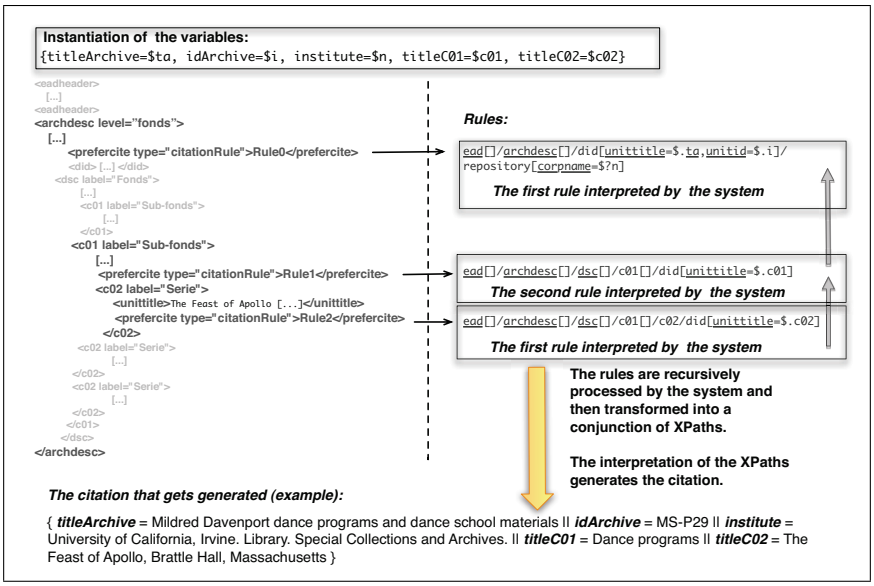

Figure 3: EAD: A set of rules that generates a machine-readable citation and an example of what it generates.

all the information for the citable unit in addition to some extra information about the part of that unit that is of interest. At this point we should observe that our simple name/value pair concrete syntax for the expression of citations is probably inadequate. A more sophisticated syntax may be required to delineate location information and what parts of a citation refer to the citable unit.

Note that if, as is the case in IUPHAR, the document is "fully keyed" [4]; that is, there is already a defined key for every node in the document, there is no overhead for adding location information for nodes that are not citable units. As an example suppose that we want to cite a specific datum within a ligand in an agonists table: we want to cite the fact that the action of the ligand named "CT (salmon)" (i.e. id=2097) is to be a "full agonist"; then, once the system has created the citation for the receptor (i.e. id="43") containing the ligand, it adds E' = `/agonists[]/ligand[$.i]/action[]` that once interpreted generates the following relative XPath: `/agonists/ligand[id=2097]/action` allowing the system to cite the specific datum and then to reach the cited datum from the citation. As we have seen by means of this system we do not have to define a specific rule for each datum someone may want to cite, but only for a restricted set of citable units. Every part of a dataset is then citable without requiring any additional effort.

The presented system is compliant with versioning systems which take track of the changes of a dataset; in particular we can adopt the archiving system presented in [5] and implemented by Heiko Muller's XArch[8]. As we have stated before the EAD files are not very prone to change, instead this is the case of the IUPHAR database in which corrections are made. Our system can provide in the citation a version number, for instance {*database=IUPHAR-DB, Version=15, Family=Glucagon receptor family*}. The version number allows the system to select the correct snapshot of the database stored in the archival system and than to retrieve the cited element in the form in which it was cited. In this paper, as well as in [3], we chose to use a version number to record it at the database level (i.e. the coarsest citable unit). The temporal issues of data citation lead to some interesting questions (e.g. Why do we use a version number and not time? Why should the version number be recorded at the coarsest level?) that we shall further investigate providing an extension of the presented system comprising the temporal dimension.

[8] `http://www.dcc.ac.uk/resources/tools-and-applications/xarch`

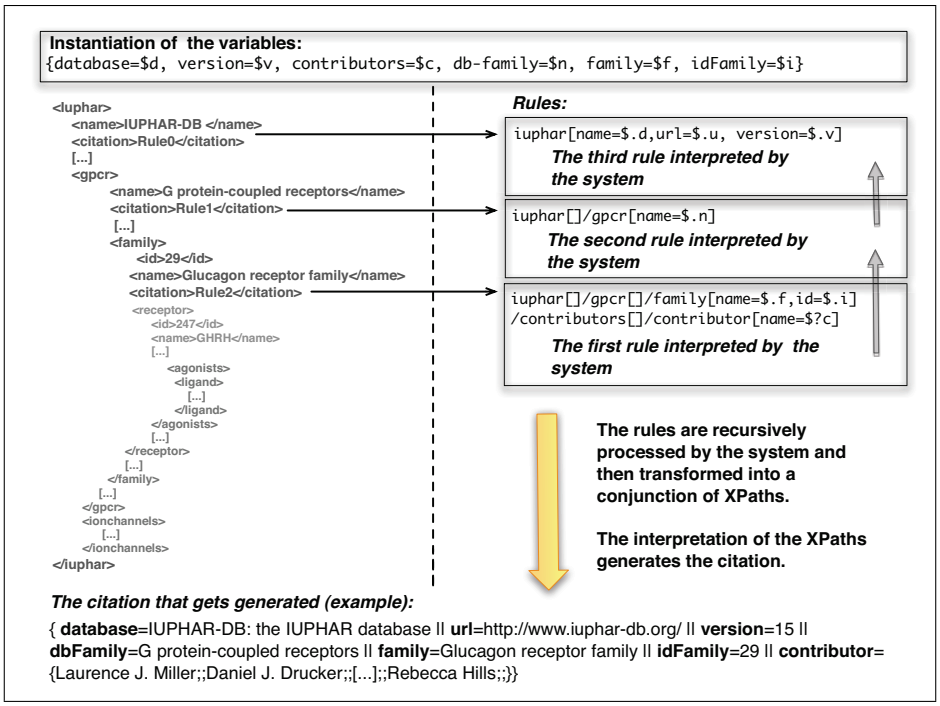

Figure 4: IUPHAR: A set of rules that generates a machine-readable citation and an example of what it generates.

5 Final Remarks

In this paper we outlined the main requirements a citation system must fulfill in order to cite structured and evolving datasets. We have argued that persistent object identifiers alone do not satisify the the conventional requirements of citations and have shown how a simple rule-based system can be used both to extract location information and other relevant data from a hierarchically structured data set.

An assumption we have made is that the data set is hierarchical and that this hierarchy is intimately connected with the structure of citations. This is certainly true for most scientific data sets. However there is a growing movement to express data by "linking": that is, through ontologies in which the underlying structure is a large graph and the notion of a citable unit or authorship is not immediately obvious. There are some proposals [7] to place a layer on top of this graph in order to deal with issues of provenance and trust. Whether this will lead to the hierarchical structures that are suitable for defining citations or whether ontologies will give rise to a form of scholarship in which citations are not needed remains to be seen.

References

[1] M. Altman and G. King. A Proposed Standard for the Scholarly Citation of Quantitative Data. *D-Lib Magazine*, 13(3/4), March/April 2007.

[2] T. Arnold-Moore and R. Sacks-Davis. Databases of Legislation: the Problems of Consolidations. Technical report, Collaborative Information Technology Research Institute (CITRI), 1994.

[3] P. Buneman. How to cite curated databases and how to make them citable. In *Proc. of 18th Int. Conf.: on Scientific and Statistical Database Management, SSDBM 2006*, pages 195–203. IEEE Comp. Soc., 2006.

[4] P. Buneman, S. B. Davidson, W. Fan, C. S. Hara, and W. C. Tan. Keys for xml. *Computer Networks*, 39(5):473–487, 2002.

[5] P. Buneman, S. Khanna, K. Tajima, and W. C. Tan. Archiving Scientific Data. *ACM Trans. Database Syst.*, 29:2–42, 2004.

[6] S. Callaghan, F. Hewer, S. Pepler, P. Hardaker, and A. Gadian. Overlay Journals and Data Publishing in the Meteorological Sciences. *Ariadne*, 2009.

[7] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. In *Proc. of the 14th Int. Conf. on World Wide Web*, pages 613–622, New York, NY, USA, 2005. ACM.

[8] T. Green. We Need Publishing Standards for Datasets and Data Tables. OECD Publishing White Paper, Org. for Economic Co-operation and Development, 2010.

[9] J. Klump, R. Bertelmann, J. Brase, M. Diepenbroek, H. Grobe, H. Höck, M. Lautenschlager, U. Schindler, I. Sens, and J. Wächter. Data Publication in the Open Access Initiative. *Data Science Jour.*, 5:79–83, 2006.

[10] M. B. Line and A. Sandison. Progress in Documentation: 'Obsolescence' and Changes in the Use of Literature with Time. *J. of Docum.*, 30(3):283–350, 1974.

[11] D. V. Pitti. Encoded Archival Description. An Introduction and Overview. *D-Lib Magazine*, 5(11), 1999.