DINAH, a philological platform for the construction of multi-structured documents

Pierre-Édouard Portier and Sylvie Calabretto

Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
{pierre-edouard.portier,sylvie.calabretto}@insa-lyon.fr

Abstract. We consider how the construction of multi-structured documents implies the definition of structuration vocabularies. In a multi-users context, the growth of these vocabularies has to be controlled. Therefore, we propose using the trace of users activity to limit this growth and document the vocabularies. A user will, for example, be able to follow and annotate the track of a vocabulary concept: from its creation to the last time it was used. From a broader point of view, this work is grounded on our Web based philological platform, DINAH, and is mainly motivated by our collaboration with a group of philosophers studying the handwritten manuscripts of Jean-Toussaint Desanti.

1 Introduction

We study how multi-structured documents are constructed in a multi-users context composed of philologists. Our work is based on experience gained working with philosophers who are building a digital edition of the handwritten archives of French philosopher Jean-Toussaint Desanti (1914-2002). Digital editing covers the whole editorial, scientific and critical process that leads to the publication of an electronic resource. In the case of manuscripts, editing mainly consists in the transcription and critical analysis of digital facsimiles, that is to say the creation of a textual document associated with the images of a handwritten manuscript. We found that the problem of constructing multi-structured documents was at the heart of their work. Indeed, they need to let coexist a multiplicity of structures in order to be able to access a document according to many interpretations. First, we will describe a methodology that promotes the emergence of multiple structures in a multi-users context. Then, we will introduce a dynamic documentation mechanism that can be used to control the growth of structuration vocabularies.

2 Construction of multi-structured documents

We define the notion of multi-structured documents and describe the problem of their representation. Then, we introduce a methodology for their construction.

2.1 Multi-structured documents

Definitions

A resource is anything uniquely identified by an URI. Fragments, intervals, zones, terms, classes, binary relations, structuration vocabularies and documents are resources.

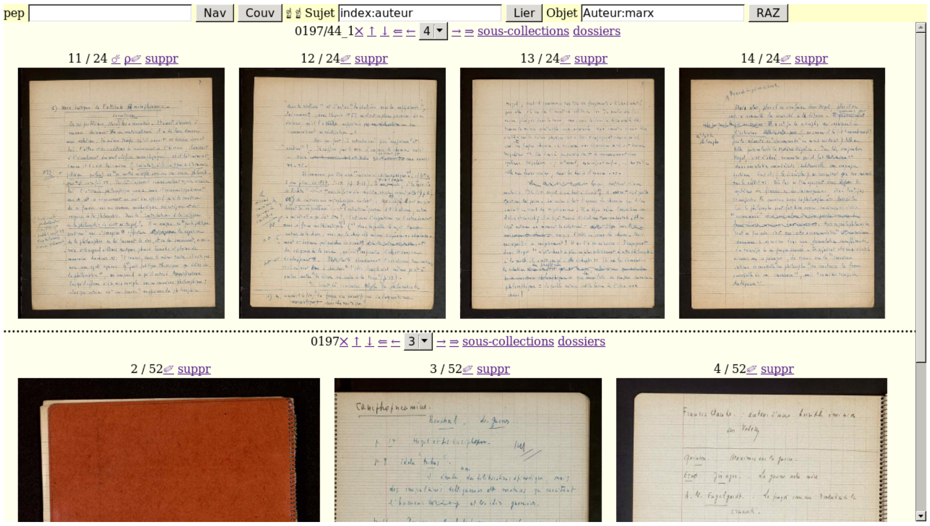
A fragment is a part of document content. Our documents are textual documents and manuscripts images. In the case of textual documents a fragment is the pair $(D, (inf, sup))$ where D is a document identifier, and (inf, sup) is an *integer interval* addressing a part of the document. In the case of images a fragment is the pair $(I, ((x1, y1), (x2, y2))$ where I is an image identifier and $((x1, y1), (x2, y2))$ are the coordinates of a *rectangular zone* of the image.

A term is a string of characters and a *class* is a set of terms. A *binary relation* $R(x, y)$ links together two resources and a *structuration vocabulary* is a set of binary relations. Finally, a *multi-structured document* is a document with fragments participating in relations that belong to multiple structuration vocabularies.

Before proceeding further, we should exemplify the previous definitions. It is also the occasion to introduce some functionalities of our philological software platform named DINAH. Consider the following scenario: a philologist finds a consistent subset about Marx inside a stack of pages of consequent size. He isolates this subset by creating a new collection (see figure 1). He creates a relation "mainSubject" between this collection and the term "marx" from the class "Author". He begins to transcribe the collection and also creates relations, such as "quotation", "citationTitle", between intervals of the transcribed text and the document (see figure 2). He discovers later that this collection is in fact a preparation for another work he found in the archive. He creates a relation "preparationFor" between the two collections (see figure 3). Etc. Etc. These newly created relations dynamically update the faceted navigation interface that can be used to find specific collections or pages by iterative refinement (see figure 4).

How is it that, for example, a user chooses to place the relation "citationTitle" within the "citations" vocabulary while he affects the relation "hasLine" to the "physicalStructure" vocabulary? In a multi-users context, how a user will know the meaning of a relation created by someone else? We will address the first question in the remaining parts of this section, and the second question in the next section. We should now recall some characteristics of the existing models for the representation of multi-structured documents.

Existing models



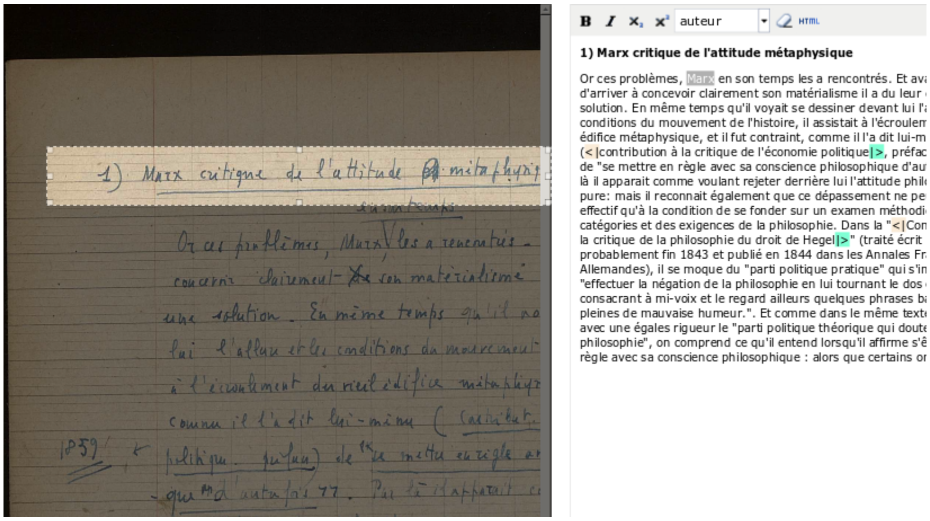
Sujet: index:auteur
Objet: Auteur:marx

Fig. 1. Creation, reordering, navigation and annotation of collections of images. Subject (or object) of the relation is dragged on the subject (or object) label, the relation itself is choosen (or created if it didn't exist) from an autocomplete menu

Multi-structured documents have to be analyzed in their historical context where the most used formalisms for documents representation (first SGML then XML) implied tree structures. That is why this problem has so far been considered under the technical point of view of overlapping hierarchies. From our previous example, let say a page has been transcribed and relations have been created to indicate where citations occur. Then, the lines of text are isolated in order to align the transcription with the manuscript facsimile. It might happen that a quotation overlaps two lines and there would be locally a graph structure: a natural use of XML becomes impossible (see figure 5). We now describe different solutions for the representation of multi-structured documents.

We divide the set of existing solutions into four classes: historical solutions, ad-hoc solutions, models not compatible with XML and finally models compatible with XML. We characterize each solution according to four criteria. The first one is the "genericity" and determines, when a model exists, if we can modify it in order to manage problems outside of the initial scope of multi-structured documents representation. The second criterion measures the quality of the implementation of the solution. The third is about the existence and effectiveness of "query mechanisms" for multi-structured documents. The last criterion determines if the model is robust to change of document content or document structures.

CONCUR [1] is a feature of SGML designed to allow the integration inside a same document of tags extracted from different DTDs. Thus, if the definitions



1) Marx critique de l'attitude métaphysique

Or ces problèmes, Marx en son temps les a rencontrés. Et ava
d'arriver à concevoir clairement son matérialisme il a du leur
solution. En même temps qu'il voyait se dessiner devant lui l'a
conditions du mouvement de l'histoire, il assistait à l'écroulem
édifice métaphysique, et il fut contraint, comme il l'a dit lui-m
(<|contribution à la critique de l'économie politique|>, préfac
de "se mettre en règle avec sa conscience philosophique d'au
là il apparait comme voulant rejeter derrière lui l'attitude phil
pure: mais il reconnait également que ce dépassement ne pe
effectif qu'à la condition de se fonder sur un examen méthodi
catégories et des exigences de la philosophie. Dans la "<|Con
la critique de la philosophie du droit de Hegel|>" (traité écrit
probablement fin 1843 et publié en 1844 dans les Annales Fr
Allemandes), il se moque du "parti politique pratique" qui s'in
"effectuer la négation de la philosophie en lui tournant le dos
consacrant à mi-voix et le regard ailleurs quelques phrases ba
pleines de mauvaise humeur.". Et comme dans le même text
avec une égales rigueur le "parti politique théorique qui doute
philosophie", on comprend ce qu'il entend lorsqu'il affirme s'ê
règle avec sa conscience philosophique : alors que certains or

Fig. 2. Transcription and annotation of a manuscript page

of the overlapping tags appear in different DTDs, the representation problem of multi-structured documents is solved. However, because of its complexity, this SGML proposal has never been entirely implemented.

The TEI [2][1] describes different syntactic solutions for the representation of multiple hierarchies into the same text (as the use of milestones or the fragmentation of elements, etc.). The main disadvantage of this solution is the impossibility to effectively use the standard XML tools (XQuery, XPath, ...) with the resulting multi-structured documents.

Since the main problem for the representation of multi-structured documents seems to be the syntactic limitations of XML, some solutions are based on models with alternative syntaxes. However they cannot profit from the galaxy of tools offered by XML. Among those solutions, we can distinguish LMNL [3] and TexMecs [4] which are alternatives to XML (formal models and syntaxes) specifically designed for the representation of overlapping structures, from propositions that take advantage of the native graph model of RDF to represent multi-structured documents. Among these, the most convincing certainly is EARMARK [5]. The notions of ”location”, ”range”, ”markup item”, etc. used for modelling multi-structured documents are precisely defined in an OWL ontology. Moreover, the SPARQL language can be used to query the documents. It is to be noted that the origins of the EARMARK proposal are to be found in two previous works: annotations graphs [6] are used, in the context of linguistic research, to represent documents as graphs so as to avoid the overlapping hierar-

[1] Text Encoding Initiative. http://www.tei-c.org/

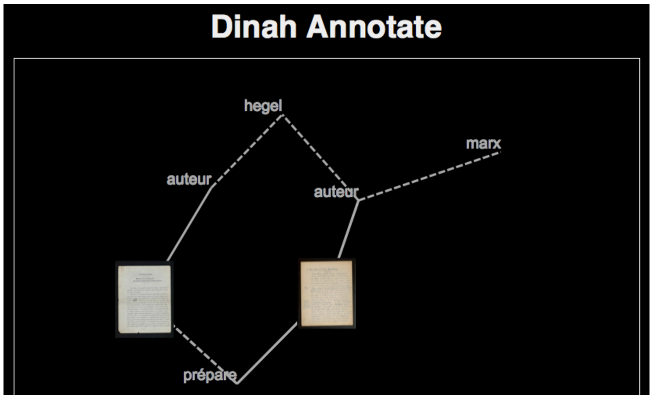
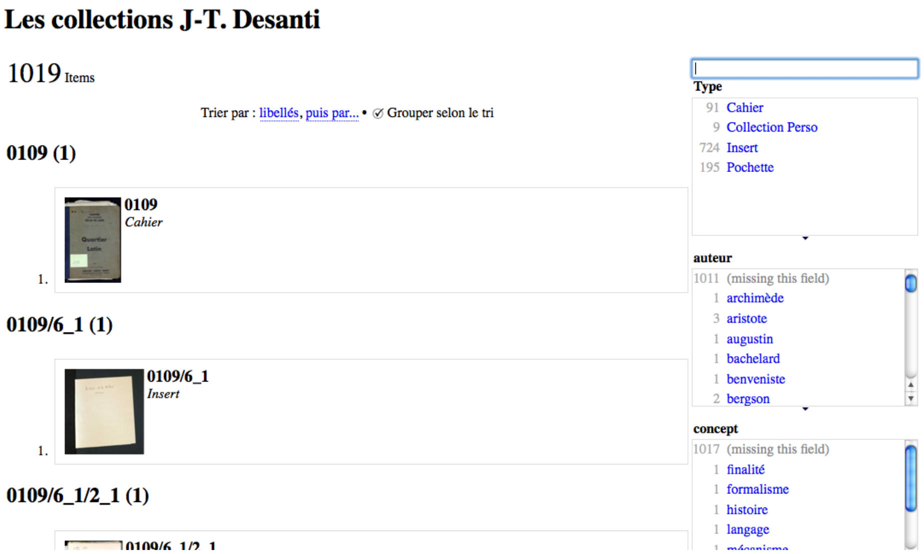


Fig. 3. Visualization of relations

chies problem ; RDFTef [7] can be seen as an adaptation of annotations graphs for the RDF standard formalism.

Finally there are solutions that remain compatible with XML but either extend the XML model itself or modify some XML tools (such as XPath and XQuery) to work with multi-structured documents. Representatives of the first category, the multi-colored trees [8] and the delay nodes [9] solutions have very similar models based on an extension of the core XML model to consider documents as set of XML trees. But unlike multi-colored trees, for delay nodes no XPath extension is necessary in order to navigate inside the structures. We now introduce members of the second category (documents syntactically expressed with XML but accompanied by modified XML tools to operate on them). GODDAG [10] (General Ordered Descendant Directed Acyclic Graph), MSXD [11], MonetDB [12] and MultiX [13] are similar proposals since in each case several trees are defined over the same textual content by sharing their leaves (textual fragments). MSXD introduces for the first time the idea of a schema for multi-structured documents. The MonetDB proposal is an extension to the MonetDB/XQuery XML SGBD with optimized query operators added to XPath with four new axis steps. These steps have been implemented very efficiently by using a region index and fast algorithms. MSDM is a lightweight solution that needs no more than a few specialised XQuery functions. Each one of these four previous solutions fails at managing change in content or structures since the entire structures have to be reconstructed every time modifications happen. MuLaX [14] is an adaptation of the previously described SGML CONCUR option to the XML world. An editor has been developed as an Eclipse plugin for the creation of MuLaX documents, but no query mechanism has been defined. Finally, feature structures [15] are a general purpose knowledge representation format that can be used as a representation format for XML documents annotated with heterogeneous tag sets, it was adopted as a standard by the TEI in 2006. Feature structures have solid mathematical foundations. In particular the two operations of unification and



Les collections J-T. Desanti

1019 Items

Trier par : libellés, puis par... • ☑ Grouper selon le tri

0109 (1)

1. 0109
Cahier

0109/6_1 (1)

1. 0109/6_1
Insert

0109/6_1/2_1 (1)

0109/6_1/2_1

Type

91	Cahier
9	Collection Perso
724	Insert
195	Pochette

auteur

1011	(missing this field)
1	archimède
3	aristote
1	augustin
1	bachelard
1	benveniste
2	bergson

concept

1017	(missing this field)
1	finalité
1	formalisme
1	histoire
1	langage
1	mécanisme

Fig. 4. Navigation among collections

generalisation are well defined and offer very interesting perspectives for the combination of multi-structured documents. However, there is no specialised query mechanism and no way of managing change in content or structures.

Table 1 summarizes the analysis by affecting, as objectively as possible, a score from 0 to 3 to each criterion (genericity, quality of implementation, query mechanisms, management of changes in data and structures), for each solution.

2.2 A strategy for the construction of multi-structured documents

The previous solutions help us understand what multi-structured documents are and how they can be represented, but none of them seem to be interested in the way structures appear! They must appear in the process of document construction. In a previous work [16] we designed a methodology for the creation and maintenance of multi-structured documents. It was based on a set of Haskell (a functional programming language) functions. Since then, significant changes occured. We will explain on the previous example of a multi-structured document (see figure 5) how we now model this process of document construction. First of all, we have to say that from the previous analysis we choose to represent our documents in the RDF formalism but, as it will be understood in the following explanation, we voluntarily impose each structure to be hierarchical (as for the MultiX, MSXD and GODDAG solutions).

We saw that the technical issue of multi-structured documents is the one of overlapping hierarchies. Moreover, if we do not consider the documents as immutable objects but as dynamic objects that have to be constructed, we must

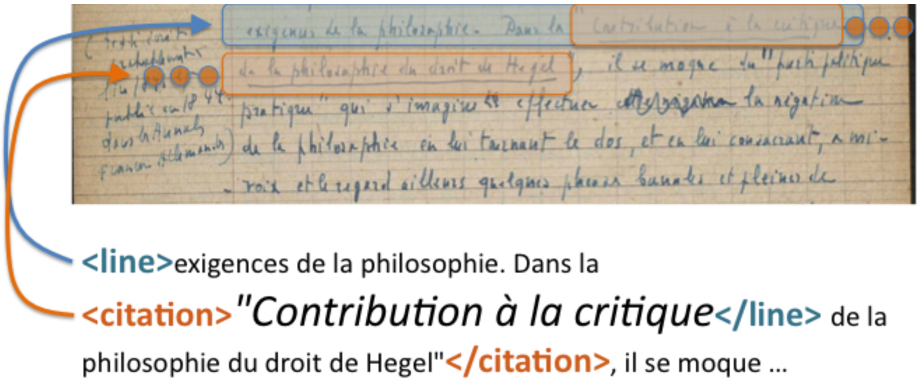

<line>exigences de la philosophie. Dans la
<citation>"Contribution à la critique</line> de la
philosophie du droit de Hegel"</citation>, il se moque ...

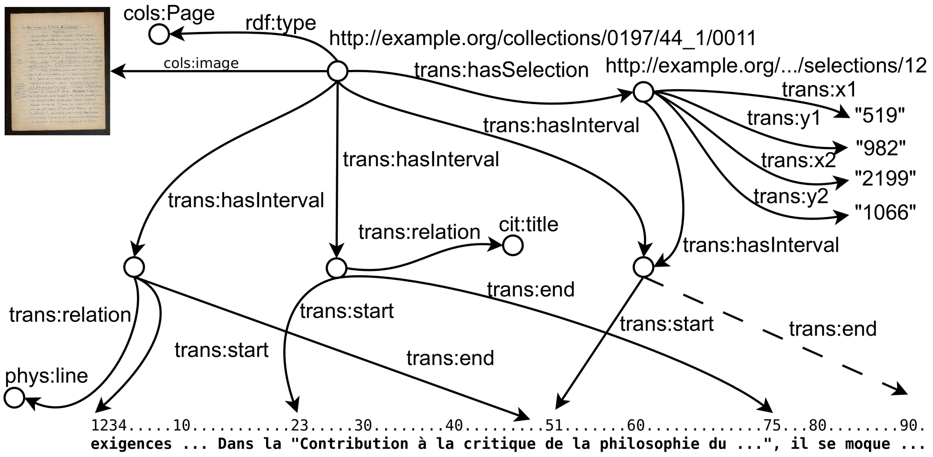
Fig. 5. Illustration of overlapping hierarchies

Table 1. Rating of existing solutions for the representation of multi-structured documents

model		genericity	implementation	query mechanisms	structure and data changes
TEI Guidelines	redundant encoding	0	1	0	0
	empty elements	0	1	0	0
	virtual elements	0	1	0	0
CONCUR		0	1	0	2
MuLaX		0	2	1	2
TexMECS		0	2	1	2
LMNL		0	2	0	2
Delay Nodes		1	2	2	0
Annotations Graphs		2	2	2	2
RDF (RDFTEF)		3	1	1	2
EARMARK		3	1	3	3
MonetDB		1	3	3	1
MCT		2	2	2	1
Features Structures		3	1	1	1
MSXD		2	2	3	0
GODDAG		3	2	3	2
MSDM/MultiX		3	2	3	2

admit the fact that overlapping hierarchies must happen at precise times. We should take an example. Let say a user annotated some citations titles and quotations he found in his transcription of a manuscript. Later he is told that in order to precisely align his transcription with the original facsimile he should annotate each line of the manuscript. So, he begins this new annotation task and since the ”line” relation did not exist he adds it to the current vocabulary (the one already containing ”citationTitle”, ”quotation”, etc.). At some time, while he has already marked some lines, a new line he would like to describe overlaps with an existing citation title. Our system (DINAH) will then alert him about an incompatibility between the relations "citationTitle" and "line" and advice him to assign either "citationTitle" or "line" to another, and possibly new, vocabulary. In this case, he may assign "line" to a "physical structure" vocabulary. Figure 6 is a sample of the resulting graph.

Finally, our strategy for the management of multi-structured documents promotes the construction of a multiplicity of structures that should reflect the perspectives adopted by the users while accessing the documents. Each user has the liberty to create new vocabularies. Moreover, when overlapping hierarchies are detected they are encouraged to solve the problem by introducing a new vocabulary. In our multi-users context, this liberty could lead to an uncontrolled growth of vocabularies with lots of duplicate usages, synonyms, etc. That is why the next section present a proposal for the dynamic documentation of structuration vocabularies.



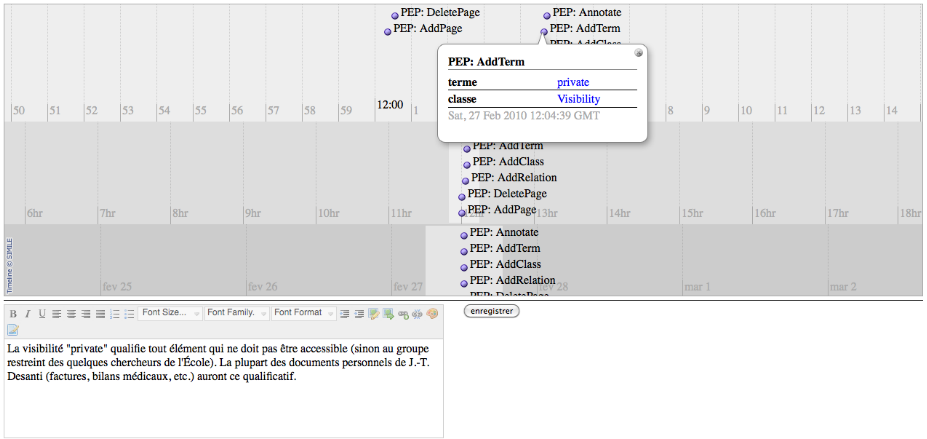
```
PREFIX trans: <http://desanti.org/schemas/transcriptions#>
PREFIX cols:  <http://desanti.org/schemas/collections#>
PREFIX cit:   <http://desanti.org/schemas/citations#>
PREFIX phys:  <http://desanti.org/schemas/physicalStructure#>
```

Fig. 6. Sample from our RDF representation of multi-structured documents

3 Reflexions on structuration vocabularies

3.1 Dynamic documentation

Our idea for the dynamic documentation of structuration vocabularies relies on the monitoring of user actions. When a user wants to know how to use a term or a relation he can ask for a representation of the trace of users actions centered on the action that leads to the term (or relation) creation or any instances of its use. This trace can itself be annotated. Users benefit from this last kind of annotations to document the vocabularies (see figure 7). Most of the time the user who document a term or a relation is the one who first created it. In case of multiple annotations they are ordered by the name of the annotator.



PEP: AddTerm

terme	private
classe	Visibility

Sat, 27 Feb 2010 12:04:39 GMT

La visibilité "private" qualifie tout élément qui ne doit pas être accessible (sinon au groupe restreint des quelques chercheurs de l'École). La plupart des documents personnels de J.-T. Desanti (factures, bilans médicaux, etc.) auront ce qualificatif.

Fig. 7. Visualization of the trace of user activities:

3.2 Trace model

Existing approach There are few works dealing with the use of activity traces for knowledge management ([17] being one of the most representative). They insist on the reflexive nature of the "use traces" as a way to share knowledge. They also define generic (and quite complex) activities models and transformations rules to go from the original trace to one with the right granularity level in order to be meaningful to the user. However, we choose to adopt a more lightweight approach well adapted to our needs.

A lightweight model We define a simple RDF vocabulary to represent actions (see listing 1.1 in the turtle RDF syntax). The only requirement is that each time a developer add a new Action to the system he has to create sub-properties of the "withArgument" property for each argument of the new action. We then use simple SPARQL queries to build representations of the trace (see figure 7).

Listing 1.1. Our trace model

```
PREFIX users: <http://desanti.org/schemas/users#>
PREFIX traces: <http://desanti.org/schemas/traces#>
INSERT INTO <http://desanti.org/> {
  traces:Action           a              rdfs:Class .
  traces:hasDoer             a                     rdf:Property .
  traces:hasDoer             rdfs:domain           traces:Action .
  traces:hasDoer             rdfs:range            users:User .
  traces:hasTimestamp        a                     rdf:Property .
  traces:hasTimestamp        rdfs:domain           traces:Action .
  traces:withArgument        a                     rdf:Property .
  traces:withArgument        rdfs:domain           traces:Action .
  traces:documentation       a                     rdf:Property .
  traces:documentation       rdfs:domain           traces:Action .
  traces:documentation       rdfs:range            rdfs:Literal .
  traces:withInterval        rdfs:subPropertyOf    traces:withArgument .
  traces:withInterval        rdfs:range            trans:Interval .
  traces:withInterval        rdfs:label            ”intervalle” .
}
```

4 Comparison with existing philological platforms

Though this work deals mainly with the creation of multi-structured documents, it remains generic enough and can be compared to other philological platforms. We divide them in two categories: first platforms of historical interest, next Web based platforms.

4.1 Historical platforms

BAMBI [18] (Better Access to Manuscripts and Browsing of Images) is, according to the authors, ”an hypermedia system allowing historians to read and transcribe manuscripts, write annotations, and navigate between the words of the transcription and the matching piece of image in the facsimile of the manuscript”. It was the first philological software platform. It does not allow typed annotations.

Part of the DEBORA [19] (Digital Access to Books of the Renaissance) project consisted in a digital library system with collaborative features. It introduced the notion of ”virtual books”. A virtual book is the representation of a path among pages of the entire archive. But they are not resources themselves, they cannot be annotated. However we can consider this system as a first step towards a reflexive system that places users in front of their own activities.

HyperNietzsche [20] (today named Nietzschesource) was a pioneer digital library platform. A path mechanism is present, very similar to the virtual books of the DEBORA project. However as for the virtual books, the paths are not resources and thus cannot truly enter in a collaborative process that would allow to exchange and annotate them.

4.2 Web based platforms

Collate [21], TALIA [22], PINAKES [23], BRICKS [24] and JeromeDL [25] are philological platforms based on semantic Web technologies. They offer high quality mechanisms for collaborative annotations. But they do not provide convergence mechanisms to isolate and document annotations vocabularies.

Armarius [26] is used to classify and annotate collections of manuscripts. It only provides untyped generic annotations. But it offers a view of all the user actions that occurred during the current session and plans to apply graph matching algorithms in order to, for example, deduce probabilities for the next actions. Thus, it can be compared with our use of traces.

5 Conclusions

We introduced the little-studied problem of multi-structured documents construction. We did not follow the conventional view that considers the heart of the problem to be the technical difficulty of representing overlapping hierarchies. On the contrary, we chose to consider overlapping hierarchies events as triggers for the creation of new structures. Furthermore, in order to manage the growth of structuration vocabularies we introduced a dynamic documentation mechanism based on the users traces of actions. Finally, all the propositions have been implemented in our philological software platform named DINAH.

References

1. Goldfarb, C.F.: The SGML handbook. Oxford University Press, Inc., New York, NY, USA (1990)
2. Burnard, L., Bauman, S.: Tei p5: Guidelines for electronic text encoding and interchange. (2007)
3. Tennison, J., Piez, W.: The layered markup and annotation language (lmnl). In: Extreme Markup Languages. (2002)
4. Huitfeldt, C., Sperberg-McQueen, M.: Texmecs: An experimental markup metalanguage for complex documents. (2003)
5. Peroni, S., Vitali, F.: Annotations with earmark for arbitrary, overlapping and out-of order markup. In Borghoff, U.M., Chidlovskii, B., eds.: ACM Symposium on Document Engineering, ACM (2009) 171–180
6. Maeda, K., Bird, S., Ma, X., Lee, H.: Creating annotation tools with the annotation graph toolkit. In: Proceedings of the Third International Conference on Language Resources and Evaluation. (Apr 2002)
7. Tummarello, G., Morbidoni, C., Pierazzo, E.: Toward textual encoding based on rdf. In: ELPUB. (2005) 57–63
8. Jagadish, H.V., Lakshmanan, L.V.S., Scannapieco, M., Srivastava, D., Wiwatwattana, N.: Colorful xml: one hierarchy isn't enough. In: SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2004) 251–262
9. Le Maitre, J.: Describing multistructured xml documents by means of delay nodes. In: DocEng '06: Proceedings of the 2006 ACM symposium on Document engineering, New York, NY, USA, ACM (2006) 155–164
10. Sperberg-McQueen, C.M., Huitfeldt, C.: Goddag: A data structure for overlapping hierarchies. In: DDEP/PODDP. (2000) 139–160
11. Bruno, E., Murisasco, E.: Multistructured xml textual documents. GESTS International Transactions on Computer Science and Engineering **34**(1) (november 2006) 200–211

12. Alink, W., Bhoedjang, R.A.F., de Vries, A.P., Boncz, P.A.: Efficient xquery support for stand-off annotation. In: XIME-P. (2006)
13. Chatti, N., Kaouk, S., Calabretto, S., Pinon, J.M.: MultiX: an XML-based formalism to encode multi-structured documents. In: Proceedings of Extreme Markup Languages'2007, Montréal (Canada). (August 2007)
14. Hilbert, M., Witt, A., Québec, M., Schonefeld, O.: Making concur work. In: Extreme Markup Languages. (2005)
15. Stegmann, J., Witt, A.: Tei feature structures as a representation format for multiple annotation and generic xml documents. In: Proceedings of Balisage: The Markup Conference 2009. Balisage Series on Markup Technologies, vol. 3 (2009). doi:10.4242/BalisageVol3.Stegmann01. (august 2009)
16. Portier, P.E., Calabretto, S.: Creation and maintenance of multi-structured documents. In: DocEng '09: Proceedings of the 9th ACM symposium on Document engineering, New York, NY, USA, ACM (2009) 181–184
17. Laflaquière, J., Settouti, L.S., Prié, Y., Mille, A.: Trace-based framework for experience management and engineering. In: KES (1). (2006) 1171–1178
18. Bozzi, A., Calabretto, S.: The digital library and computational philology: The bambi project. In: ECDL. Volume 1324 of Lecture Notes in Computer Science., Springer (1997) 269–285
19. Nichols, D.M., Pemberton, D., Dalhoumi, S., Larouk, O., Belisle, C., Twidale, M.B.: Debora: developing an interface to support collaboration in a digital library. In: European Conference on Digital Libraries, Springer (2000) 239–248
20. D'Iorio, P.: Nietzsche on new paths: The hypernietzsche project and open scholarship on the web. In: Maria Cristina Fornari, Sergio Franzese (ds.), Friedrich Nietzsche. Edizioni e interpretazioni, Pisa ETS. (2007)
21. Stein, A., Keiper, J., Bezerra, L., Brocks, H., Thiel, U.: Collaborative research and documentation of european film history: The collate collaboratory. In: In International Journal of Digital Information Management (JDIM), special issue on Web-based collaboratories from centres without. (2004) 30–39
22. Hahn, D., Nucci, M., Barbera, M.: The talia library platform - rapidly building a digital library on rails. In: 4th Workshop on Scripting for the Semantic Web. (2008)
23. Scotti, A., Nuzzo, D.: Pinakes – a modeling environment for scientific heritage database applications. In: Proc. of Reconstructing science – Contributions to the enhancement of the European scientific heritage Workshop, Ravenna, Italy (2001)
24. Bertoncini, M.: On the move towards the european digital library: Bricks, tel, michael and delos converging experiences. In: Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings. Volume 4675 of Lecture Notes in Computer Science., Springer (2007) 440–441
25. Kruk, S.R., Woroniecki, T., Gzella, A., Dabrowski, M.: Jeromedl - a semantic digital library. In Golbeck, J., Mika, P., eds.: Semantic Web Challenge. Volume 295 of CEUR Workshop Proceedings., CEUR-WS.org (2007)
26. Doumat, R., Egyed-Zsigmond, E., Pinon, J.M., Csiszar, E.: Online ancient documents: Armarius. In: ACM DocEng'08. Proceeding of the Eighth ACM Symposium on Doucument Engineering, ACM (September 2008) 127–130