Indexing and Querying Overlapping Structures

Faegheh Hasibi
Advised by Prof. Svein Erik Bratsberg and Adj. Assoc. Prof. Øystein Trobjørnsen
Norwegian University of Science and Technology
Trondheim, Norway
faeghehh@idi.ntnu.no

ABSTRACT

Overlapping is a common phenomenon that can be seen when structural components of a digital object cannot be neatly nested into each other. Due to the intrinsic complexity of overlaps, hierarchies are not sufficient to describe overlapping components. For the same reason, tree-based indexing and query processing techniques cannot be used for overlapping structures. The current research on overlapping structures revolves around encoding and modelling data, while indexing and query processing methods remain unsolved. Our research focuses on indexing overlapping structures to provide rapid response for large scale search engines. In this paper, we describe overlapping structures and the need for indexing these non-hierarchical structures. We also describe our proposed data model among the existing overlapping data models, which is the first step towards indexing and querying.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

overlapping structures, indexing, query processing, unstructured data

1. INTRODUCTION

Overlapping structure is the intrinsic feature of any kind of digital data, where several independent structural items refer to the same content. The most common example is a document with two distinct structural views, when the logical view is `section/subsection/paragraph` and the physical view is `page/column`. Each single structural view of this document is a hierarchy and the components are either disjoint or nested inside each other. The overlapping issue arises when one structural element cannot be neatly nested into others. For instance, when a paragraph of this starts in one page and terminates in the next page. Similar situations can appear in videos and other multimedia contents, where temporal or spatial constituents of a media file may overlap each other [19].

The most used model for expressing structure of documents is based on hierarchies, which ensures that each region is nested within another and the regions can be accessed by use of parent-child or ancestor-descendant relationships. This tree data structure requires organizing structural information of digital objects in a single tree, which is not applicable for overlapping structures. In other words, tree-based markup languages (e.g. XML) and the corresponding indexing and retrieval techniques are not sufficient for documents with overlapping structures.

As a consequence of hierarchical insufficiency for overlapping structures, scholars have introduced several solutions for overlapping problem. TEI (Text Encoding Initiative) [7] suggests several methods to deal with non-hierarchical structures in SGML or XML context. However, these methods are just syntactical solutions to represent non-hierarchical structures and are not based on a well-defined data model. Unlike XML solutions, most of non-XML languages are based on a specific overlapping data model. SGML CONCUR [20], TexMECS [13] and LMNL [17] are some of markup languages that are based on Multiple hierarchies, GODDAG and LMNL data model, respectively.

Although there exists some solutions for modelling overlapping structure, the main issue is indexing these structures to provide rapid response for large scale search engines [11]. This research focuses on finding a profound method for indexing and query processing of overlapping structures.

This paper is organized as follows. Section 2 motivates this research by providing use cases and applications of overlapping structures. The background and research questions are presented in Section 3 and 4, respectively. Section 5 describes research methodology of this research and the conclusion is presented in Section 6.

2. USE CASES AND APPLICATIONS

Overlapping is a situation that is more common than it may be thought of. Lots of scholars encounter overlaps in the area of computational linguistic, speech and complex text analysis. Some of these situations are as follows:

Structural and Literal Annotations of Documents. Annotating analytic notations of text files is a frequent case that needs to handle overlapping structures. These notations can be any kind of information, such as structural

SIGIR' 12, July 28-August 1, 2013, Dublin, Ireland

```
<sample>
 <video>
  <scene id="Intro" start="00:00" end="00:08"/>
  <scene id="Interview" start="00" end="01:04"/>
  <scene id="Outro" start="01:04" end="01:34"/>
 </video>
 <audio>
  <music artist="Beatles" start="00:00" end="00:45"/>
  <music artist="Bach" start="01:00" end="01:34"/>
 </audio>
</sample>
```

Figure 1: Annotations for the video example in XML (causing overlapping annotations)[2]

views (e.g. physical and logical structures), phonetic features, grammatical structure and part-of-speech tagging [5]. Encoding all these structures in the same document causes overlapping. For instance, literature containing verses and nested quotations in which a quote overlaps several verses.

Digitizing old manuscripts. This task is another domain that needs dealing with overlapping structures. Building electronic version of old manuscripts requires encoding massive amounts of information, such as textual content, physical location of texts, linguistic information, visibility of characters and information about damages of manuscript [8]. In general, annotating several aspects of an object in a markup language implies overlapping structures. The following example represents various encodings of a text fragment from an old manuscript, where a *word* overlaps a *line* element.

```
<line>hu bu me haefst afrefredne
<w>aeg </line>
<line> ber </w> ge <dmg>mid</dmg> binre smealican
</line>
```

Annotating non-textual objects. Overlapping structures can be seen not only in documents, but also while encoding another kinds of digital data. Annotating of non-textual objects is mostly done by use of stand-off annotations, which separates source document from the structural views. Alink and et al. [3] applied stand-off annotations to BLOBs (Binary Large objects) to manage and query forensic data. In this work, BLOBs are hard disk images and they are annotated in different hierarchies using XML documents. Figure 1 illustrates video annotations in a stand-off XML document.

Search in Indexed Document Collections. FSIS (FAST Search for Internet Sites) [1]is a Microsoft search platform which provides a number of tools for content processing, indexing, search and query processing. It has multiple document parsers to detect the content and properties of unstructured or semi-structured documents. One of the document parsers of FSIS, extracts structure and semantic information of documents and outputs this information as annotations. These annotations contains lots of structural components that overlap each other.

3. BACKGROUND AND RELATED WORK

The background of this topic is stated in four different aspects. First, we describe the most important types of non-hierarchical structures that are needed to be considered for handling and indexing overlapping structures. The second part is about the main existing data structures for modelling overlapping structures. In the third part, the approaches for querying documents with overlapping structures is introduced and the last part is about graph indexing methods.

3.1 Non-Hierarchical Structure Types

Non-hierarchical structures have different types and they can appear while annotating structures of digital data. Every non-hierarchical data model and markup language is designed to capture all or some of these types, which are summarized here.

Classic Overlap. Classic overlaps are cases in which one element does not neatly nest inside another one. These cases are the main focus of most research on overlapping structures. The following example illustrates classic overlaps:

```
<a> John <b> likes </a> Mary </b>
```

Self-overlap. Self-overlaps are cases when two components of the same structure and with the same name overlap each other. A typical situation is when, two distinct reviewers annotating the same text region. In this case, since the comments belongs to the same hierarchy and have the same type, self-overlapping problem arises [16]. An example is as follows:

```
<a>
 <comment id="1">John
   <comment id="2">likes
 </comment> Mary
    </comment>
</a>
```

Discontinuous Elements. Discontinuous elements refers to the situations, where a single logical region is broken into multiple physical elements. These discontinuous elements can be virtualy reconstituted by use of virtual elements [7].

3.2 Overlapping Data Models

The difficulty of handling overlaps is that overlapping structures are not hierarchies and the popular markup languages, such as XML and SGML are based on hierarchies. As a result of lacking an adequate overlapping data model, several data structures have been proposed to describe overlapping structures. In the following, we discuss these data structures and their abilities to model different types of overlaps are discussed.

3.2.1 Multiple Hierarchies (CONCUR)

The most straightforward model for the overlapping problem is to keep multiple hierarchies in a single document. This model is captured by the CONCUR feature of SGML, which maintains multiple structural views of a document. It actually extends the SGML/XML data model to a model, where multiple trees (with the same frontier) can be encoded within a single document.

The CONCUR model is represented as a part of SGML and consequently it is a legible and maintainable approach for overlapping problem. However, this model is not considered widely as a solution of overlaps. Here is a list of CONCUR drawbacks, expressed by literature [21, 9, 4].

- The model is not able to constrain relations among DTDs and as a consequence, data update (such as insert, delete and reordering of data in various views) cannot be modelled.
- CONCUR does not provide self-overlaps. Whenever two elements with the same name coinciding each other, one element have to be moved to another hierarchy. This means CONCUR needs to support unpredictable DTDs to handle self-overlaps.
- CONCUR is not recommended for encoding speech-oriented and verse-oriented and documents. Such documents have complex structural information and their structural views are not necessarily fixed to multiple DTDs.

3.2.2 LMNL

LMNL (Layered Markup and Annotation Language) is a data model associated with a markup language which was first presented at 2002 by Tennison and Piez [17]. LMNL data model is based on layers rather that hierarchies. Hence, it represents documents without forcing elements into a hierarchy, though there may exist hierarchical structure in the document (as usually is). LMNL has three main definitions: Ranges, Annotations and Atoms[18].

- **Range**: Ranges are analogous to XML elements, which means a set of ranges define LMNL. However, ranges in LMNL have no specific relations with each other (i.e parent-child or descendant-ancestor relations) and they can be nested or overlapping. Each range may have annotations.
- **Annotation**: Annotations describe properties of a range. They can support any features that a LMNL document can support and therefore each annotation can be a document itself. Annotations of a LMNL ranges are ordered and it is allowed that a single range be assigned by annotations with a same name.
- **Atom**: Atoms are elements in document that cannot be presented by characters, such as graphics and glyphs. Atoms in LMNL have their own symbols and can represent information that might be represented by empty tags in XML. They have location as well as "occupy space" that is included in the value of ranges they belong to. LMNL also supports empty tags, but unlike atoms, they only have locations.

In addition to classic overlaps, LMNL can capture self overlaps. However, the model is unable to represent discontinuous components of a text.

3.2.3 GODDAGs

GODDAG [21] is a directed acyclic graph (DAG), which is introduced to represent documents with overlapping structures. The principle behind GOODAG is that "overlap is tree-like graph, in which nodes can have multiple parentage" . GODDAG stands for Generalized Ordered Descendant Directed Acyclic Graph (GODDAG) which means each non-terminal node has ordered descendants. An Example of restricted GODDAG is shown in Figure 2.

GODDAG has different variations that are explained below.

Restricted and Generalised GODDAG.

Restricted GODDAG [21] adds some constraints to the model, as follows:

- Leaf nodes are ordered.
- Each non-terminal dominates a contiguous subsequence of leaves.
- No two nodes dominate the same subsequence of the frontier.

Restricted GODDAG is capable of representing overlaps; however its constraints rules out the possibility of modelling non-contiguous elements and self-overlaps. On the contrary, generalized GODDAG removes the restrictions by these rules:

- For each node n, arcs (n $\rightarrow$ x) are ordered.
- Leaves need not have any ordering; no contiguity rule for non-terminals.
- Two non-terminals may dominate the same set of leaves.

Generalized GODDAG can model virtual elements and discontinuous elements as well as classic overlaps.

3.3 Querying of Overlapping Structures

Querying over overlapping structures requires a mechanism that relates structural regions to each other. Iacob et al. [14] extended XPath as EXPath to query overlaps over GODDAG structures. XIRAF [2] is another system that allows querying over overlapping annotations by moving from one hierarchy to another hierarchy. XIRAF's query approach is based on Burkowski's [6] work, which adds four operations to XPath queries, that are: Select Narrow, Reject Narrow, Select Wide and Reject Wide.

It should be noted that all of these query approaches are developed for domain specific applications. However, query processing for large scale search engines needs to be investigated on indexing structures.

3.4 Graph Indexing

Overlapping data structures can be modelled by either graphs or tree-like structures, such as GODDAGs. To the best of our knowledge, there is no research directly investigating on the overlapping indexing, however there has been large number of studies on XML and graph indexing.

Accroding to [10], there are two main classes of structural indexes of XML data: numbering schemes and index graph schemes. The former is used for path joinning, while the latter is for path selection in answering XML queries. Zhang et al. [23] proposed a numbering scheme for XML documents, name *PrePost* encoding. This model is capable of precessing parent-child as well as ancestor-descendant relationships. *Dewey coding* [22] is another famous numbering scheme, which can be maintained easier than PrePost method. Jin [15] introduced a 3-hop indexing scheme, which is targeted for directed graphs with high edge-vertex density.

4. RESEARCH QUESTIONS

Our research starts with the following research question:
RQ: *How can overlapping structures be indexed and queried?*

The research will focus primarily around this research question. In order to catch all the challenges related to this

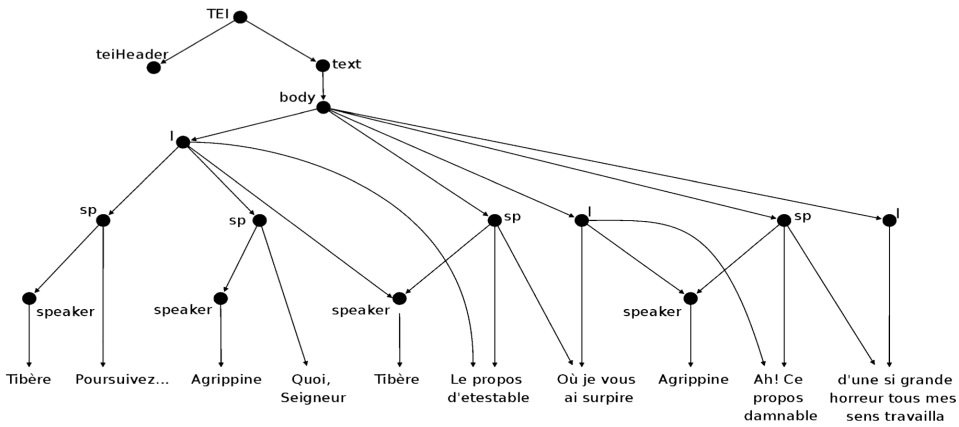

Figure 2: Restricted GODDAG data model[16]

question we need to split the principal research question to the following sub-questions:

RQ1: *Which data model is suitable for indexing of overlapping structures?*

RQ2: *Since overlapping structures are mostly encoded by use of stand-off annotations, how stand-off annotations can be converted to that data model?*

RQ3: *Which indexing approaches are more suitable for overlapping structures and how they should be adopted?*

RQ4: *Which query approaches can be used to process queries with overlapping axes? How join algorithms should work to support overlapping structures?*

So far the data model for the first research question is chosen [12] and the attempt is to find solutions for other questions.

5. RESEARCH METHODOLOGY

The process to overcome this research is to do indexing and query processing for a subset of overlapping structures and queries and then extend the solutions for more general and complicated cases.

Analogous to XML documents, overlapping structures need a data model for indexing. Our intended data model for indexing overlaps is restricted GODDAG, which is a tree-like graph, where nodes can have multiple parentage. Since this model supports simple inheritance, we can define the depth of each node (node level). Moreover, this model is the core of a framework [16] that mutually converts different overlapping formats to each other. According to this work, seven different overlapping formats can be translated to restricted GODDAG. It must be noted that self-overlaps and discontinuous elements cannot be modelled by restricted GODDAG, but these situations are not our main focus, as they are less frequent than classic overlaps.

The next step of this research is to choose an indexing method and apply it to the restricted GODDAG data model. Since overlapping structures cannot be described by trees, we cannot use XML indexing approaches. However, we intend to adopt an XML indexing approach (such as PrePost [23]) to GODDAG data structure.

In order to experiment with the indexing approaches, we have to find an appropriate dataset, which is one of the main challenges of this research. There exists some freely available document corpora for overlapping structures, but these collections are encoded using different overlapping markup formats. For instance, Cambridge Wittgenstein Archive[1] is a collection which is based on GODDAG but encoded by non-XML markup language[2] [16]. Using non-XML dataset requires additional tools to parse documents.

Our interest is to find a dataset which uses XML to annotate overlapping structures. The reason is that annotations can be used for encoding overlapping structures of both textual and non-textual (e.g video and audio) files. Moreover, FAST [1] search server of Microsoft - as an example for industrial search engines - uses annotations to represent structure of documents.

In order to implement our indexing and query processing methods, we need an appropriate framework. Existing open-source search engines can index hierarchical structures, but we need to index non-hierarchical structures. To overcome this challenge, there are two possibilities. One is to write a customized search engine for indexing overlapping structures. Another possibility is to extend an existing search engine, such as Lucene and Terrier. We may choose the second possibility and extend Lucene search engine to our purpose, since Lucene has well-organized source code and is used by a large number of academic publications.

To evaluate our work, we specify a set of queries and observe that if our methods are able to correctly answer these query types or not. We targeted these four types of XPath queries, which are used in XIRAF system [2]:

- /select-narrow: Return elements which are contained by another element
- /select-wide: Return elements which partially overlap another element
- /reject-narrow: Return all elements which are not contained by a context element (inverse of select-narrow)
- /reject-wide: Return only those elements which do not even partially overlap a context node (inverse of select-wide)

[1]http://www.wittgen-cam.ac.uk/cgi-bin/forms/home.cgi

[2]This archive uses techniques developed by the MLCD project. These techniques are GODDAG data structure and MECS and TexMECS markup languages.

Since there is no previous work on indexing of overlapping structures, we do not have a baseline system to compare our results with. However, after the first experiment, we will have a baseline that can be used for our next experiments.

6. CONCLUSION

Overlap is a common phenomenon in annotating digital data, either textual or non-textual files. Previous studies discussed encoding and modelling of overlapping structures and some of them addressed the querying of overlaps, where they add some axes to XPath/XQuery language. However, these querying techniques are tested for specific purpose applications and not for big data.

The aim of this work is to develop indexing and query processing approaches for overlapping structures. As a first step for indexing overlapping structures, we have chosen restricted GODDAG as an appropriate data structure for modelling overlaps. In the future, we will develop an indexing method for overlapping structures based on GODDAG data model.

7. ACKNOWLEDGMENTS

This work is supported by the iAd Centre and funded by the NTNU and the Research Council of Norway.

8. REFERENCES

[1] Fast search server 2010 for internet sites. http://sharepoint.microsoft.com/en-us/Pages/Videos.aspx?VideoID=26, 2010.

[2] W. Alink. Xiraf: An xml-ir approach to digital forensics, 2005.

[3] W. Alink, R. A. F. Bhoedjang, P. A. Boncz, and A. P. De Vries. Xiraf - xml-based indexing and querying for digital forensics. *Digit. Investig.*, 3:50–58, Sept. 2006.

[4] D. Barnard, R. Hayter, M. Karababa, G. Logan, and J. McFadden. Sgml-based markup for literary texts: Two problems and some solutions. *Computers and the Humanities*, 22:265–276, 1988. 10.1007/BF00118602.

[5] S. Bird and M. Liberman. A formal framework for linguistic annotation. *Speech Communication*, 33(1ââĆň“2):23 – 60, 2001. <ce:title>Speech Annotation and Corpus Tools</ce:title>.

[6] F. J. Burkowski. Retrieval activities in a database consisting of heterogeneous collections of structured text. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 112–125, New York, NY, USA, 1992. ACM.

[7] T. Consortium, L. Burnard, and S. Bauman. *TEI P5: Guidelines for electronic text encoding and interchange*. TEI Consortium, 2012.

[8] A. Dekhtyar and I. E. Iacob. A framework for management of concurrent xml markup. *Data and Knowledge Engineering*, 52(2):185 – 208, 2005.

[9] S. DeRose. Markup overlap: a review and a horse. *Interchange (UK)*, 11(1):16 – 29, 2005/03/. Markup Overlap;SGML;XML;TEI milestone markup;OSIS documents;.

[10] G. Gou and R. Chirkova. Efficiently querying large xml data repositories: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(10):1381 –1403, oct. 2007.

[11] F. Hasibi. Indexing and querying overlapping structures. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1144, 2013.

[12] F. Hasibi and S. E. Bratsberg. Non-hierarchical structures: How to model and index overlaps? *CoRR*, abs/1408.1, 2014.

[13] C. Huitfeldt and C. Sperberg-McQueen. Texmecs: An experimental markup meta-language for complex documents. http://www.hit.uib.no /claus/mlcd/papers/texmecs. html, 2001.

[14] I. Iacob and A. Dekhtyar. Queries over overlapping xml structures. Technical report, Technical Report TR 438-05, U. of Kentucky, CS Dept, 2005.

[15] R. Jin, Y. Xiang, N. Ruan, and D. Fuhry. 3-hop: a high-compression indexing scheme for reachability query. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 813–826, New York, NY, USA, 2009. ACM.

[16] P. Marinelli, F. Vitali, and S. Zacchiroli. Towards the unification of formats for overlapping markup. *New Rev. Hypermedia Multimedia*, 14(1):57–94, Jan. 2008.

[17] W. Piez. Lmnl in miniature, an introduction. http://www.piez.org/ wendell/LMNL/Amsterdam2008/presentation-slides.html, 2008.

[18] W. Piez. Luminescent: parsing lmnl by xslt upconversion. *Proceedings of Balisage: The Markup Conference 2012. Balisage Series on Markup Technologies*, 8, August 2012.

[19] P. Salembier and A. B. Benitez. Structure description tools. *Journal of the American Society for Information Science and Technology*, 58(9):1329–1337, 2007.

[20] C. Sperberg-McQueen and C. Huitfeldt. Concurrent document hierarchies in mecs and sgml. *Literary and Linguistic Computing*, 14(1):29–42, 1999.

[21] C. Sperberg-McQueen and C. Huitfeldt. Goddag: A data structure for overlapping hierarchies. In P. King and E. Munson, editors, *Digital Documents: Systems and Principles*, volume 2023 of *Lecture Notes in Computer Science*, pages 606–630. Springer Berlin-Heidelberg, 2004.

[22] I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang. Storing and querying ordered xml using a relational database system. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, SIGMOD '02, pages 204–215, New York, NY, USA, 2002. ACM.

[23] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman. On supporting containment queries in relational database management systems. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, SIGMOD '01, pages 425–436, New York, NY, USA, 2001. ACM.