jInfer XML Schema Inference Framework

Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, Matej Vitásek
Advisors: RNDr. Irena Mlýnková, Ph.D., Martin Nečaský, Ph.D.

Praha, 2011

Abstract

Today, many algorithms solving XML schema inference problem exist. None of them is widely used by public to practically solve the problem. This is because the lack of simple user interface to algorithms. We present NetBeans ([**?**]) based pluggable framework to fill this gap. Thus enabling people dealing with XML in practice to obtain schema for their documents by following simple steps, without need to read and understand theoretical aspects. We hope our solution will boost usage of XML schemas in practice, since creating schema for existing set of documents will be more effordable (one doesn't have to write one by hand).

Introduction

Although many algorithms ([Aho96, BNST06, BNV07, VMP08, Vyh]) try to solve problem of infering schema for existing set of XML documents, have you ever tried to sovle the problem in practice?

When the problem arises, one wants to find solution as quickly as possible, without need to investigate many scientific papers. We focus on this group of potential users - programmers, system administrators, xml coders, xml maintaners in private and scientific sector.

Framework is designed extensible and is implemented as a set of plug-ins for NetBeans platform. This brings second group of users (although much smaller in counts) of framework - scientific developers willing to experiment with their new algorithms may easily gain benefits of user interface provided, thus speeding up their development, easing them comparison with other algorithms already developed for framework.

But it is not all about user interface. We bring in XML/XSD/DTD import/export modules ready to use. We implement sample infering algorithm (see [Aho96]). We provide visualization tools to visualize input XML documents as a set of grammar rules, and tools to visualize non-deterministic finite automatons used in many of infering algorithms. Thus extending existing NFA base algorithm with user interaction is made simple for scientific developers.

Related work

Most of the algorithms named, doesn't have avalaible implementation, nor are made ready to use. Probably best solution nowadays is [BNV08]. It deals with schema infering, user interaction, schema visualization and user-aided refinement. Authors use their own algorithms for schema inference ([BNST06, BNV07]). Compared to our software, they are better in user interaction and schema visualization. There is no mention of extensibility, however, which we solve as our top priority.

Since both are trying to solve nearly same problem, we predict that we will approach some good ideas from other solution in further work.

Some XML editors also offers XML schema generation,

Short architecture overview

We divide the process of inference into three steps:

1. TODO steps, graph from architecture, short mention of internal structure = grammar

Extensibility

TODO short description of how easily can framework be extended ??? maybe not

More work/Open issues

There are still many issues to be worked on. First is set of supported input/output formats. On input, one can imagine XML Queries, on output Relax NG or Schematron. There is room for implementing schema visualization tool into jInfer, to enable user-aided schema refinement. Many algorithms published wait to be implemented as jInfer modules, maybe some of them will be tried in study of our master theses.

Conclusion

Framework presented aims to be the tool used to obtain schemas from XML/XSD/DTD files (and more inputs af-

ter extending). While maintaining distributable bundle for users, with best of module selections and algorithms implemented so far, we predict it will be used in future studies of new algorithms as a test environment. Best of them will probably migrate into jInfer source tree to aid common users to solve their document set quickly enough, to even bother with schema generation at all.

Weak side of the solution is the lack of good quality automatic or semi-automatic algorithms to be used by common user. Goal was to produce stable framework for future development and we hope to negate this issue by implementing master theses in this environment, providing it with bundle of different algorithms, solving different input and output formats and infering more concise schemas.

References

[Aho96] H. Ahonen. *Generating grammars for structured documents using grammatical inference methods*. PhD thesis, Department of Computer Science, University of Helsinki, Series of Publications A, Report A-1996-4, 1996.

[BNST06] Geert Jan Bex, Frank Neven, Thomas Schwentick, and Karl Tuyls. Inference of concise dtds from xml data. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 115–126. VLDB Endowment, 2006.

[BNV07] Geert Jan Bex, Frank Neven, and Stijn Vansummeren. Inferring xml schema definitions from xml data. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 998–1009. VLDB Endowment, 2007.

[BNV08] Geert Jan Bex, Frank Neven, and Stijn Vansummeren. Schemascope: a system for inferring and cleaning xml schemas. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1259–1262, New York, NY, USA, 2008. ACM.

[VMP08] Ondřej Vošta, Irena Mlýnková, and Jaroslav Pokorný. Even an ant can create an xsd. In *DASFAA'08: Proceedings of the 13th international conference on Database systems for advanced applications*, pages 35–50, Berlin, Heidelberg, 2008. Springer-Verlag.

[Vyh] Julie Vyhnanovská. Automatic construction of an xml schema for a given set of xml documents.