Ontology Integration in XML

Euna Jeong
Department of Computer Science and Information Engineering National Taiwan University
4F 11 Lane-9 Bai-Ren Jie Hsin-Tien, Taipei Taiwan
eajeong@agents.csie.ntu.edu.tw

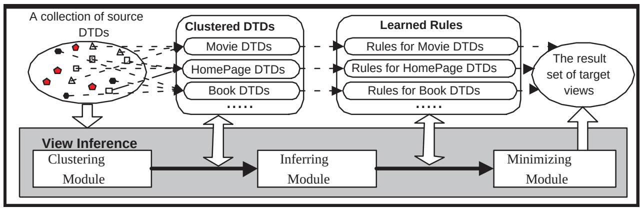
Chun-Nan Hsu
Institute of Informationi Science, Academia Sinica
chunnan@iis.sinica.edu.tw

Abstract

We study the problem of automatically generating an integrated schema for XML DTDs. Introducing a novel *view inference* approach, we shows that the set of views and source descriptions can be automatically derived.

Introduction The problem of information integration has become significant as the growing number of information sources on the Internet. Information integration systems provide users with an integrated schema of underlying sources. The integrated schema is designed by hand and a mapping between the integrated schema and the source schemas is needed for the system to answer queries. As XML (Bray, Paoli, & Sperberg-McQueen 1998) has become a new standard for representation and exchange of data on the Internet, in this article, we consider the problem of automatically generating an integrated schema for different XML DTDs with similar document types.

Architecture XML with a DTD is self-descriptive and provides a semistructured data model. These properties render that DTDs defining similar document type have structural and naming similarities. Given a collection of source DTDs, we propose a view inference approach which automatically derives the set of integrated views and source descriptions.



Step 1: Clustering DTDs takes a collection of source DTDs as an input. A DTD is modeled as an edge-labeled graph. The type of an object is defined by adjacent objects and the relation between them. Our strategy is to merge types by *top-down merge*. If two types have the same parent type, they will be merged into a new type. Based on merged types, DTDs are re-defined with them and clustered into one of DTD classes. Generic matching methods (Sanfeliu 1990) are used to know how similar two DTDs are.

Step 2: Learning Rules learns the general rules describing source DTDs in the same class. Our approach is based on tree grammar inference(Miclet 1990). The technique of the generalization of k-tails' concept is applied to learning tree grammars(Levine 1982). At first, the automaton contains one state for each subtree of the sample set T, if they have the same set of k-tails, then their states are made identical. This gives an automaton that recognizes a tree language containing T.

Step 3: Minimization of Tree Automata optimizes the learned rules. The learned rules are a set of states in a tree grammar and need to be transformed to an integrated view. The integrated views allow the user to formulate queries to XML documents. In this step, we use *bottom-up merge*. If two states have the same subtree, but different root labeling, then they will be merged. As a result, the minimized tree automaton will be more flexible than before.

Conclusions We have presented how to automatically generate integrated views for XML from source DTDs. Our approach allows for fully automatic generation of integrated views from a collection of DTDs.

References

Bray, T.; Paoli, J.; and Sperberg-McQueen, C. M. 1998. Extensible Markup Language(XML) 1.0. Technical report, W3C.

Levine, B. 1982. The use of tree derivatives and a sample support parameter for inferring tree systems. In *IEEE Transactions on Pattern Analysis and machine Intelligence*, 25–34.

Miclet, L. 1990. *Syntactic and structural pattern recognition*. World Scientific. chapter 9.

Sanfeliu, A. 1990. *Syntactic and structural pattern recognition*. World Scientific. chapter 6.