Dipartimento di Informatica
Università degli Studi di Verona

Rapporto di Ricerca
RR 21/2004, 14 June 2004

Towards association rules for XML documents

Carlo Combi
Barbara Oliboni
Rosalba Rossato

Dipartimento di Informatica
Università degli Studi di Verona

Rapporto di ricerca RR 21/2004
14 June 2004

Towards association rules for XML documents

Carlo Combi
Barbara Oliboni
Rosalba Rossato

Abstract

In this work we propose a flexible approach to extract and evaluate association rules on XML documents. We describe two kinds of association rules: *structural associations* and *value associations*. A *structural association* allows one to capture the similarity of an XML document with respect to a given structure, while a *value association* allows one to capture the similarity of the information contained in the XML document with respect to a given scenario. Moreover, we show how it possible to compose these associations in order to describe complex association rules on XML documents.

Keywords: flexible association rules, XML documents

1 Introduction

XML is spreading out as a standard for representing, exchanging, and publishing information ([1], [14]). XML is a markup language which is suitable for representing semistructured data, which are often described as "self-describing", i.e. no pre-imposed schema or type is needed for data interpretation itself. The need of describing association rules over XML documents has arisen in some work ([6], [13]). Indeed, even without documents with a fixed structure, it could be interesting to be able to identify in an XML document some recurrent situations: for example, in a document related to hospital patients, it could be useful to discover that a "patient" element usually contains "therapy" elements. Moreover, it could be of interest to discover that most "therapy" subelements of the element "patient" have the content "anti-hypertension therapy".

In this work we propose a flexible approach to evaluate association rules for XML documents. We propose two kinds of associations: *structural associations* which allow us to consider the structure of an XML document, and *value associations* which allow us to consider the contents of the document. The evaluation of *structural* and *value associations* returns similarity degrees taking into account the structure of the document and its contents, respectively. We show how it is possible to compose these kinds of associations in order to express more complex association rules, called *complex association rules*, composed by a structural part and a value part.

The structure of the paper is as follows. In Section 2 we discuss related work for the description of association rules for XML documents. In Section 3 we describe *structural* and *value* association rules for XML document and in Section 4 we show how it is possible to compose the previous mentioned associations to obtain *complex association rules*. In Section 5 we report the conclusion and draft some future research topics.

2 Related Work

In the context of semistructured data, a main issue is related to the description of functional dependencies over XML documents. In the relational context, a *functional dependency* over a relation r defined on the set Z of attributes, is a logical implication in the form $X \rightarrow Y$ where $X, Y \subseteq Z$. The functional dependency $X \rightarrow Y$ is satisfied if and only if for every tuple $s, t \in r$, if $s[X] = t[X]$ then $s[Y] = t[Y]$. For example, if we want to describe the fact that each university course (represented by the attribute *Course*) has an unique number of associated credits (represented by the attribute *Credits*), we can describe this relationship by means of a functional dependency $Course \rightarrow Credits$.

In the XML context, the problem of describing functional dependen-

cies is more complex than in the relational one. The information contained in an XML document could be partial and incomplete, and moreover the document could be without a Document Type Definition (DTD). It means the possibility of missing information in the XML document which can involve the violation of the required dependencies. Though the problem of describing functional dependencies for XML is still an open problem, in the literature there are some proposals which deal with this topic ([5], [10], [11]).

The first definition of functional dependencies for XML data is in [5]; in this work the authors propose also a normal form, based on the proposed dependencies for XML documents. In [11] the authors try to overcome the problems due to the nature of XML data and give a precise definition of functional dependencies without assuming the existence of a DTD. In [10] the authors propose an XML-based language to define functional dependencies for XML documents. In these approaches functional dependencies for XML are described in term of implication between paths (starting from the root) and their satisfaction is evaluated taken into account the reachable values (w.r.t. the considered paths).

Another interesting research topic is the problem of expressing integrity constraints for semistructured data and XML documents. In [8], the authors highlight the need of a formal definition of integrity constraints and define the most important categories of constraint for XML. In [7], the authors study absolute and relative keys for XML, and investigate their associated decision problems. They also propose a new key constraint language for XML which can handle keys with a complex structure. In general, a key is described by means of a path on (sub)tree with a specific root. In [9], the authors adopt the formal definition of keys described in [7] and propose a technique to obtain a compact set of keys from an XML document.

Another recent research direction in the context of XML data is related to the extraction of association rules from XML documents ([6], [12], [13]). In general, association rules describe the co-occurrence of data items in a large amount of collected data [4].

The formal definition of an *association rule*, introduced by Agrawal et al. [3] for the problem of mining association rules between sets of items in a large database of customer transactions, is reported in the follow.

Let $\mathcal{I} = I_1, I_2, \ldots, I_m$ be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented by means of a binary vector, with $t[k] = 1$ if t bought the item I_k, and $t[k] = 0$ otherwise. There is a tuple in the database for each transaction. Let X be a set of some items in $\mathcal{I}$. A transaction t *satisfies* X if for all items I_k in X, $t[k] = 1$.

Definition 1 *An association rule is an implication in the form $X \Longrightarrow I_j$, where X is a set of items in $\mathcal{I}$, and I_j is a single item in $\mathcal{I}$ that is no present in X.*

The rule $X \Longrightarrow I_j$ is satisfied in the set of transactions T with a confi-

dence factor $0 \leq c \leq 1$ *if and only if at least* $c\%$ *of transactions in* T *that satisfy* X *also satisfy* I_j.

The quality of an association rule is described by means of two parameters *support* and *confidence*. Support corresponds to the frequency of the set $X \cup Y$ in the dataset, while confidence corresponds to the conditional probability $p(Y|X)$, i.e. the probability of finding Y having found X. Several works deal with the problem of mining association rules in large databases ([2], [3], [4]).

In [13], the authors show a technique which allows one to extract association rules, by using XQuery, from XML documents. In [6], the authors describe association rules from XML documents by introducing an XML-specific operator, called XMINE RULE, which is based on the use of XML query languages.

Our approach is included in this research direction but, unlike to the previous mentioned proposals, allows one to describe flexible association rules for XML. We propose a technique which associates a similarity degree to each considered rule. The similarity degree is quite different from the standard parameters used for association rules, such as support and confidence, but we think it is interesting to evaluate, in a flexible way, the satisfiability of an association rule in an XML document. The introduced flexibility is related to the usage of different evaluation techniques.

3 Flexible association rules

In this work we suppose to have a set of association rules to evaluate on a set of XML data. In this Section we briefly describe a flexible approach to evaluate different kinds of association rules on XML documents. In Figure 1 is reported a well-formed XML document which contains the information about PhD students.

In this work we represent XML documents as *XML graphs* (see Figure 2 for the graphical form of the XML document reported in Figure 1). We choose to represent XML elements by means of nodes, and their containment relationships by means of non-labeled edges. We consider general XML documents, thus we allow the presence of mixed XML elements. For this reason, we have *mixed XML graphs* composed by three kinds of nodes: *simple*, *complex*, and *mixed* nodes. A *simple* node is a leaf and has a specific value (see node Address in Figure 2), while a *complex* node has at least one outgoing edge and it could have a specific value. When a complex node has a value, then it is called *mixed* (*mixed complex* node). In Figure 2 the nodes PhDStudent are mixed nodes, while the node Person is a *complex* node.

In our approach it is possible to describe two interesting classes of association rules over a (set of) XML document(s): *structural* associations and *value* associations.

```
<?xml version=”1.0” encoding=”UTF-8" ?>
<Person>
  <PhDStudent>  Jennifer Brown
    <PhDInfo>  2nd year </PhDInfo>
  </PhDStudent>
  <PhDStudent>  Marc Steven
    <PhDInfo>  3th year
       <Salary>    800  </Salary>
    </PhDInfo>
  </PhDStudent>
  < ResearchGroup> DB
    <PhDStudent>  Roger Moore
      <Salary>  800  </Salary>
    </PhDStudent>
  </ResearchGroup>
  <PhDStudent>  Paul O’ Connor
    <PhDInfo> 1st year </PhDInfo>
    <Salary>  800  </Salary>
  </PhDStudent>
  <ResearchGroup> AI
    <PhDStudent> Elly Bawer
      <BioInfo>
        <Address> Blossom Ave </Address>
        <PhDInfo>   2nd year
          <Salary>  900 </Salary>
        </PhDInfo>
      </BioInfo>
    </PhDStudent>
  </ResearchGroup>
</Person>
```

Figure 1: A well-formed XML document.

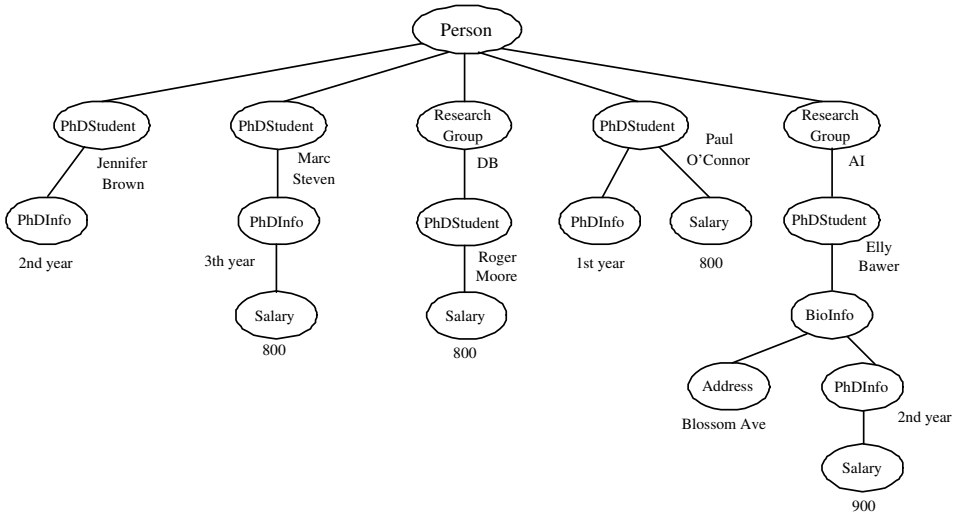

Figure 2: The XML graph representing the XML document of Figure 1.

A *structural* association allows one to evaluate the similarity of an XML document with respect to a given structure. A *value* association allows one to check the similarity of the information contained in an XML document with respect to a given scenario.

As we will describe in the follow, when we want to describe an association over an XML document we can use the logical notation $Element_S \rightarrow Element_D$, where $Element_S$ is called *starting element* and $Element_D$ is called *destination element*, but in order to evaluate an association on an XML graph we use its graphical representation.

The graphical representation of an association is a direct graph which shows the fact that the destination node (representing the destination element) can be reached by the starting node (representing the starting element). In the case of description of value associations, the values associated to the nodes are also reported in the graphical representation.

For example, Figure 3 shows the graphical representation of the association rule PhDStudent $\rightarrow$ Salary.

3.1 Structural association rules

Structural associations allow one to evaluate the similarity of an XML document with respect to a given structure. For example, with respect to the XML document of Figure 1, a structural association could be PhDStudent $\rightarrow$ Salary. This association means that a PhD student has to have a salary.

The graphical representation of this association, evaluated on the XML graph of Figure 2, is reported in Figure 3 and describes the fact that starting

from a PhDStudent node, a Salary node can be reached.

(a)

Figure 3: A structural association.

In order to establish the similarity degree of the XML graph with respect to a given structural rule, we propose the following steps:

- **subgraph extraction**: in this step it is possible to count and extract the subgraphs which satisfy the required structural association. The subgraphs extraction can be realized by means of a suitable graph-search algorithm finding all the subgraphs having as root the starting node and containing, at some level, the destination node. The extracted subgraphs can satisfy the required structure in a more or less precise way, i.e. the destination node could be directly connected to the starting node or between them could be a path composed by more than one edge.

- **subgraph weight**: in this step it is possible to associate a *subgraph weight* to each subgraph found in the previous step. The weight takes into account the correspondence of the subgraph with the required structure and it can be calculated by using different *evaluation techniques*. In the case of a structural association rule, a reasonable evaluation technique can take into account the number of edges needed to reach the destination node (in the example the node Salary), starting from the starting node (in the example the node PhDStudent).

  The subgraph weight related to the i-th subgraph is denoted as ssd_i and it can be calculated in this way:

  $$ssd_i = \frac{1}{NrEdge(Node_{S_i}, Node_{D_i})}$$

  where $Node_{S_i}$ and $Node_{D_i}$ represent the starting and destination nodes respectively. ssd_i represents the weight associated to the i-th subgraph having as root $Node_{S_i}$. $NrEdge(Node_{S_i}, Node_{D_i})$ is a function which returns the number of edges between $Node_{S_i}$ and $Node_{D_i}$. The proposed evaluation technique is an example of approach to evaluate association rules considering the subgraphs structure. Other suitable techiniques can be studied and used.

- **structural satisfiability degree:** in this step it is possible to evaluate the *structural satisfiability degree* (ssd) of the XML graph with

respect to the required structural association. The *ssd* value is a value in [0,1] and it assumes value 1 whether all the extracted subgraphs respect exactly the required structure. The *ssd* can be calculated with the formula:

$$ssd = \sum_{i=1}^{n} ssd_i \cdot \frac{1}{n}$$

where n is the number of the subgraphs extracted in the first step.

For example, evaluating the constraint of Figure 3 on the graph in Figure 2, the graph-search returns the four subgraphs having as root the node PhDStudent and containing the node Salary. In Figure 4 we show these subgraphs included in dashed region.

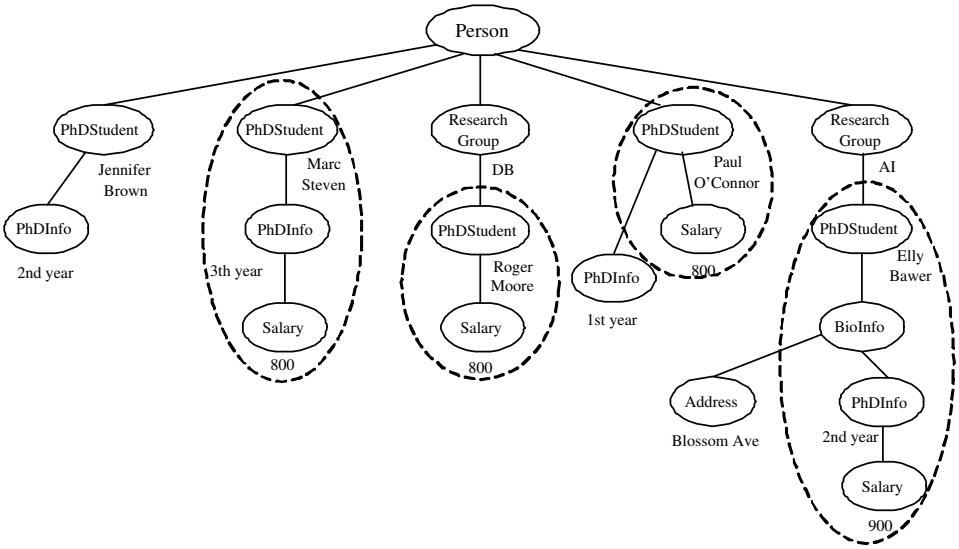

Figure 4: The subgraphs which satisfy the structural constraint PhDStudent → Salary.

The first retrieved subgraph, having as root the node PhDStudent with value *Marc Steven*, has weight 1/2 because between the nodes PhDStudent and Salary there are two edges, the second and third subgraphs have weight 1 because there is only one edge between the nodes PhDStudent and Salary, while the fourth subgraph has weight 1/3 because the path between PhDStudent and Salary is composed by three edges.

Thus, the *ssd* has value:

$$\frac{1}{2} \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + \frac{1}{3} \cdot \frac{1}{4} = \frac{17}{24} = 0.708$$

The value 0.708 represents the structural satisfiability degree of the structural association rule PhDStudent → Salary. This value describes the fact that the extracted subgraphs do not respect in an exact way the required structure, i.e. in some subgraphs the path between the considered nodes has length greater than one. In general, the *ssd* value is a satisfiability degree which represents likeness degree and has a different meaning with respect to the support.

3.2 Value associations

A *value association* allows one to check the similarity of the information contained in an XML document with respect to a given scenario. For example, a value association on the XML document shown in Figure 1 could be the association PhDStudent → Salary(800) describing that PhD students have a salary with value 800. In general, the logical formula which describes a value association is $Element_S \rightarrow Element_D(val)$.

The graphical representation of the proposed value association is shown in Figure 5 and describes the fact that the node Salary, with value 800, must be reached from the node PhDStudent. The node Salary is the destination node ($Node_D$), while the node PhDStudent is the starting node ($Node_S$).

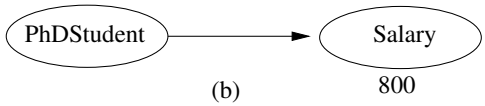

Figure 5: A value association.

In order to evaluate the *value satisfiability degree* (*vsd*) of this kind of association, it is possible to choose several evaluation techniques. A simple way is to count how many subgraphs satisfy the required value association, i.e. in how many subgraphs the destination node, with value 800, is reached from the starting node. In general, the logical formula, which allows one to evaluate the *vsd* is:

$$vsd = \sum_{i=1}^{n} Value(Node_{S_i}, Node_{D_i}, val) \cdot \frac{1}{n}$$

where $Node_{S_i}$ and $Node_{D_i}$ represent respectively the starting and the destination nodes of the i-th subgraph.

Also in this case, the value n represents the number of subgraphs having as root the starting node $Node_S$ and containing the destination node $Node_D$ (as described above). In the evaluation of this association, the function $Value(Node_{S_i}$,
$Node_{D_i}, val)$ returns the value 1 if in the i-th extracted subgraph, the destination node $Node_{D_i}$ has value val, and returns the value 0 otherwise. When

we evaluate the value association rule of Figure 5, on the XML graph of Figure 2, the function $Value(PhDStudent, Salary, 800)$ checks whether in the extracted subgraphs (shown in Figure 4), the node Salary has value 800. Note that, if the destination node of the i-th subgraph is a complex node (without a specific value), the function $Value$ returns 0. With respect to the XML graph shown in Figure 2 the value satisfiability degree associated to the rule represented in Figure 5 has value:

$$1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + 0 \cdot \frac{1}{4} = \frac{3}{4} = 0.75$$

The first, second and third subgraph have degree 1, while fourth subgraph (having value *Elly Bawer* for the node PhDStudent) has weight 0 because the value of the Salary node is 900. The final result describes the fact that three subgraphs (on four) satisfy the proposed value association.

A second evaluation technique can take into account not only the value of the destination node but also its distance from the starting node. In this case, the weight associated to each subgraph is also based on the structure of the subgraph itself. The general formula which allows to calculate the value satisfiability degree with this second technique is:

$$vsd = \sum_{i=1}^{n} \frac{Value(Node_{S_i}, Node_{D_i}, val)}{n \cdot NrEdge(Node_{S_i}, Node_{D_i})}$$

The application of this second kind of evaluation technique, to the XML graph shown in Figure 2, for the association rule of Figure 5, returns the following satisfiability degree:

$$1 \cdot \frac{1}{4 \cdot 2} + 1 \cdot \frac{1}{4 \cdot 1} + 1 \cdot \frac{1}{4 \cdot 1} + 0 \cdot \frac{1}{4 \cdot 3} = \frac{5}{8} = 0.625$$

In this case the contribution of a subgraph in the final *svd* is in inverse proportion to the number of edges between the starting and destination nodes.

We are investigating other evaluation techniques to apply in order to evaluate value association rules. These techniques could consider also the difference between the information contained in the XML document and the researched information.

For example, in the considered example, PhDStudent $\rightarrow$ Salary(800), a suitable metric could associate different satisfiability degrees to the subgraphs having value 900 for the node Salary.

4 Combining association rules on XML documents

In this Section we show how the flexible association rules introduced in Section 3 can be combined to discover complex association rules on XML data.

An association rule allows one to capture a relationship between an antecedent and a consequent and returns a value which describes the percentage of satisfaction of the relationship with respect to the considered data. For example, in [3] the authors have proposed an efficient algorithm that generates all significant association rules between items in the context of a large database of customer transactions. An example of the extracted association rules is the statement that 90% of transactions that purchase bread and butter also purchase milk. Note that, in this kind of association rules, both the antecedent and the consequent take into account the value of the information, but not its structure.

Our work consider the semistructured (XML) data context, in which the same information can be represented in different ways. Thus, it could be useful and interesting to evaluate relationships, by means of particular association rules, between the structure of the XML document and its contents.

For this reason, in our approach, the antecedent of the rule is composed by a structural association, while the consequent is a value association. In this way we can define *complex association rules* taking into account both the structure of the information and its contents.

The formal definition of a *complex association rule* is:

Definition 2 *Let* s *and* v *be the logical notation of a structural association and a value association, respectively. A complex association rule is an implication in the form* $\mathsf{s} \Longrightarrow \mathsf{v}$.

The complex association rule $\mathsf{s} \Longrightarrow \mathsf{v}$ *is satisfied in the XML document* D *with a confidence factor* $0 \leq c \leq 1$ *if and only if at least* $c\%$ *of the subparts of* D *which satisfy, with a degree greater than a user defined threshold, the structural association* s *also satisfy the value association* v.

In the follow, we assume that the given user threshold for the structural association is 0. An example of *complex association rule* is

$$(\mathsf{ResearchGroup} \rightarrow \mathsf{PhDStudent}) \Longrightarrow (\mathsf{PhDStudent} \rightarrow \mathsf{Salary}(900))$$

which can be extracted on the XML graph on Figure 2. It has a confidence factor 0.5, i.e. in the 50% of the subgraphs where the node PhDStudent can be reached by the ResearchGroup node, the value of the Salary node is 900. In Figure 6, the extracted subgraphs, which satisfy the antecedent ResearchGroup $\rightarrow$ PhDStudent of the complex association rule, are included in dashed regions.

The confidence factor can be obtained by evaluating the *vsd* degree for the subgraphs satisfying the structural association, described in the antecedent of the rule, and having a satisfiability degree greater than the user threshold. For example, in this case we have to evaluate if a node Salary with value 900 can be reached starting from a PhDStudent node contained in the extracted subgraphs.

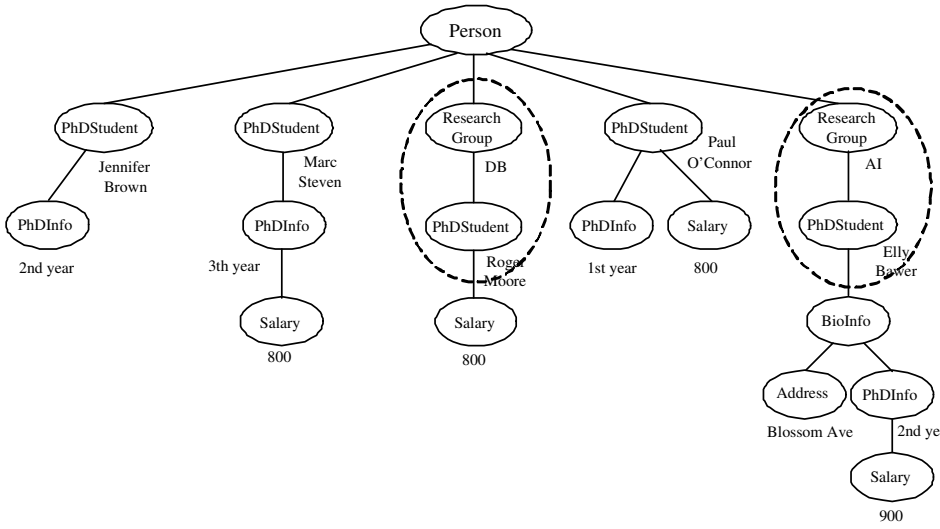

Figure 6: The subgraphs which satisfy the structural constraint ResearchGruop → PhDStudent.

In general, with our approach, given an XML graph we can obtain the satisfiability degree of the structural and value associations rules defined on it and also compose these kinds of rules in order to describe complex associations. In particular, given a structural association rule, we can extract the subgraphs satisfying it and analyze the value association rules on them.

5 Conclusions and Future Work

In this work we have proposed a flexible approach to describe association rules on XML document. We have described *structural* and *value* associations and we have proposed some evaluation techniques in order to establish the similarity degree. Structural and value associations can be composed to describe more complex association rules. The notion of association rule could be extended to study more complex situations. For example, it could be interesting to extend the notion of complex association to consider a set of structural associations in the antecedent of the rule itself and study the mixed cases in which the antecedent is composed by set of structural and value associations. Another interesting situation is the case where both the antecedent and the consequent are value associations. In this way, the complex rule could represent a kind of *flexible functional dependency*, i.e. the satisfaction of the dependency is described by means of a parameter which reveals its satisfiability instead of a boolean answer. As future work, we aim to analyze flexible functional dependencies, described by means of asso-

ciation rules between value associations and investigate suitable evaluation techniques to apply both to structural and value associations.

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: from relations to semistructured data and XML*. Morgan Kaufman, 2000.

[2] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transcations on Knowledge and Data Engineering*, pages 914–925, 1993.

[3] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proceedings of the 20th International Confernece on Very Large Data Bases*, pages 478–499. Morgan Kaufmann, 1994.

[5] M. Arenas and L. Libkin. A normal form for XML documents. In *Proceedings of the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 85–96. ACM Press, 2002.

[6] D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi. Discovering Interesting Information in XML Data with Association Rules. In *SAC 2003*, pages 450–454. ACM Press, 2003.

[7] P. Buneman, S.B. Davidson, W. Fan, C.S. Hara, and W.C. Tan. Reasoning about keys for XML. *Information Systems*, 28(8):1037–1063, 2003.

[8] P. Buneman, W. Fan, J. Simèon, and S. Weinstein. Constraints for Semistructured Data and XML. *SIGMOD Record*, 30(1):47–55, 2001.

[9] G. Grahne and J. Zhu. Discovering approximate keys in XML data. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 453–460. ACM Press, 2002.

[10] M. Li Lee, T.W. Ling, and W.L. Low. Designing Functional Dependencies for XML. In *Extending Database Technology*, pages 124–141, 2002.

[11] M.W. Vincent and J. Liu. Functional dependencies for XML. In Y.Zhang X. Zhou and M.E. Orlowska, editors, *APWeb 2003*, volume 2642 of *LNCS*, pages 22–34, 2003.

[12] J. W.W. Wan and G. Dobbie. Extracting association rules from XML documents using XQuery. In *DASFAA 2004*, pages 110–112, 2003.

[13] J. W.W. Wan and G. Dobbie. Mining association rules form XML data using XQuery. In *DMWI 2004. To appear*, 2004.

[14] J. Widom. Data management for XML-research directions. *IEEE Data Engineering Bulletin*, 22(3):44–52, 1999.

University of Verona
Department of Computer Science
strada le grazie, 15
37134 Verona
Italy

http://www.di.univr.it

University of Verona
- **Department of Computer Science**