

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	------------------------------	---------------	----------------------------------	--------------

## MARSDEN FUND FULL RESEARCH PROPOSAL (STANDARD)

### 1. TITLE OF RESEARCH PROPOSAL

Cardinality Constraints for XML:

Challenging the Trade-off between Expressiveness and Tractability

### 2. IDENTIFICATION

#### Contact Principal Investigator

Name (with title):	Associate Professor Sebastian Link
Full Address:	School of Information Management Victoria University of Wellington PO Box 600 Wellington 6015
Telephone:	(04)463 6813
Mobile telephone:	(021)236 3602
Fax:	(04)463 5446
Email:	sebastian.link@vuw.ac.nz

#### Other Principal Investigator(s)

Name (with title)	Institution	Country
Prof. Dr. Sven Hartmann	Clausthal University of Technology	Germany

### 3. SUMMARY

Describe in up to 200 words the nature of your proposed research in plain English for a general audience. This summary should be able to be used for publicity purposes if the proposal is offered funding.

The eXtensible Markup Language (XML) has evolved to be the “lingua franca” for data integration and data exchange on the Internet and elsewhere. This development has led to a dramatic increase of XML data that must be stored, managed and processed in its native format. The syntactic flexibility of XML together with the tree-like nested structure of its data impose new challenges for database researchers. It is particularly difficult to develop data management tools that are both efficient and cater for a large class of properties that are naturally exhibited by XML data.

Our project is based on the somewhat surprising observation that cardinality constraints have not been exploited in the context of XML. Cardinality constraints can express many properties of XML data that cannot be captured by other classes of XML constraints. They have a direct impact on many XML recommendations such as schema specification languages, query languages or data manipulation and transformation languages. We aim at developing a well-founded theory that gives original insight into the characteristics of XML data, enhances the semantic capabilities of XML, allows designers to make an informed choice about which classes of cardinality constraints to incorporate into XML recommendations, and provides efficient algorithms to advance XML data processing.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

#### 4. BACKGROUND

The eXtensible Markup Language (XML,[46]) is now the de-facto standard for information/data representation and exchange on the Internet and elsewhere. This development is due to the high syntactic flexibility XML provides. Consequently, an increasing number of users find it very simple to create data in XML or an XML dialect, e.g., the familiar HTML, NITF (in the news industry), WeatherML (for weather information), CellML (in bioinformatics), WML (for mobile phones) and XML Pay (for Internet-based payments) [1]. There has been wide consensus that XML documents/data should obtain the same type of management functionalities as conventional data received from relational database management systems. The essential difference between XML data and traditional data (e.g. relational data) is the extra structural, tree-like relationships between the various elements of an XML data source. This complex structure makes it rather difficult for the database community to meet the increasingly important challenge of providing full-fledged tools that can store, manage and process this data in its native format. In particular, XML has not been primarily designed to serve as a data model, and thus does not carry the semantics of its data. Integrity constraints can enhance the semantic capabilities of XML in the sense that they permit only those XML databases that are considered meaningful for the application at hand. Consequently, the study of integrity constraints has been recognised as one of the most important, yet involved and demanding areas of XML research [12, 15, 19, 42, 52, 54]. For almost all previously studied classes of XML constraints [4, 8, 10, 11, 17, 20, 21, 29, 30, 31, 51, 53] the intricate structure of XML data results in decision problems that are, in general, intractable. It remains a major challenge to find expressive classes of XML constraints that can be efficiently reasoned about [15, 19, 20, 21, 42, 52].

Our proposal is based on our observation that **cardinality constraints have not been investigated in XML** despite their evident significance [18, 25, 37, 41, 44, 45]. For instance, these constraints permit the *efficient deduction of invaluable knowledge about the database without actually processing any data*. Since querying large databases is time-consuming this can save considerable computational resources especially for requests from mobile devices. Imagine a new personalised Web-service for investors who enquire (e.g. by cell-phones) about current investment options for certain investment categories. On one hand, investors will only use this service if its costs are reasonable. On the other hand, the service provider prefers to utilise its resources only if they are paid for. Luckily, the designer specifies cardinality constraints that represent business rules of the provider: e.g. to supply between 5 and 10 options for each investor per category. The investor profiles are sources of additional constraints, e.g. that there are up to 5 categories an investor is interested in. If the underlying database system can handle the interaction of these constraints it can approximate the maximal costs of the service without querying the database itself. This minimizes the costs for unpaid services and gives investors an informed choice, too. However, **current database systems cannot reason about cardinality constraints**, e.g. decide that “there are between 25 and 50 recommendations available” is an implied constraint for investors with such a profile. With the current absence of reasoning facilities this Web-service is prohibitively expensive to implement.

Cardinality constraints are naturally exhibited by XML data since they represent restrictions that occur in everyday practice. They can express more properties than other XML constraints [4, 5, 8, 9, 20, 21, 53]. Moreover, cardinality constraints have a direct impact on various applications including schema design, query optimization/rewriting, efficient storing/updating, data exchange/integration, and consistent query answering. Commercial pressure and insufficient time resources force many database researchers to narrow their investigations to notions that have only a limited expressiveness and applicability. Hence, many characteristics of XML data remain unaddressed and most applications underexploited. Cardinality constraints have the potential to make a long-lasting contribution to database technologies. Such a contribution, however, requires **tractable reasoning capabilities for expressive constraints**.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

## 5. OVERALL AIM OF THE RESEARCH

**General Goal.** We intend to advance the current endeavours of the database community to provide commensurate management solutions that do justice to the increasing amount of native XML data. While the structural relationships of XML data render high complexity for XML data modelling, management and processing they are also sources of invaluable semantic knowledge that is often overlooked or inadequately exploited. Our critical observation is that cardinality constraints can express important, yet unaccounted, information that is naturally exhibited by XML data. Our experiences [24, 25, 32, 33, 34] will help us to introduce classes of cardinality constraints for XML and to develop a well-founded and robust theory that:

- a.) gives original insight into the properties of XML data,
- b.) enhances XML's semantic capabilities and its potential to become a first-class data model,
- c.) allows developers to make an informed choice about the classes of cardinality constraints that should be incorporated into XML modelling and XML recommendations,
- d.) provides efficient algorithms that reason about these constraints, and
- e.) therefore unlocks many application domains that have remained unexplored so far.

**Research problems.** Our major objective is the identification of classes of cardinality constraints that express as much relevant information about XML data as possible **and** yet allow us to develop efficient algorithms that decide the following problems: **1.)** implied by a given set of constraints, **2.)** in conflict with a given set of constraints, **3.)** satisfied by an XML database, and **4.)** interacting with schema definitions or other types of constraints.

**Knowledge Increase.** Findings to these questions provide insight about the *limits of XML's database management capabilities*, point us towards those characteristics of XML data that can be expressed and managed automatically, and allow us to *chart the tractability frontier* of such management tools. We will provide database developers with *unambiguous guidelines* about the benefits and drawbacks of incorporating classes of cardinality constraints into XML recommendations. Our efficient algorithms will directly find their way into various XML recommendations and therefore *enhance XML's database management facilities*.

**Impact.** The project will reveal new links between and new approaches to various areas, e.g.:

- *Databases and Discrete Maths.* Our first results show surprising links between the implication of certain cardinality constraints and constructive graph properties [32, 33]. We will fully explore these bonds and tie seemingly disparate areas, based on our previous experience [34].
- *XML database design.* The majority of XML data is created by users with minimal background in database design. With massive numbers of Web databases being created it will become increasingly difficult to change their poor organization. Cardinality constraints can be applied to ban data redundancies, processing difficulties with the database, and inefficient query processing through a database transformation known as pivoting [28].
- *Inconsistent XML databases.* In practice, most databases are inconsistent. Cardinality constraints can be used to avoid uncertain data, and we will explore two approaches towards dealing with inconsistencies: the specification of soft constraints, and consistent query answering.
- *Query optimization.* Cardinality constraints can optimize XPath [47] and XQuery [49] queries. We can develop efficient algorithms that decide if queries of some query language fragments can be optimized; and perform this optimization automatically. Moreover, cardinality constraints provide effective means to predict the number of i) query answers, ii) updates, and iii) en/decryptions. This has important repercussions for physical database design.

**Novelty.** No other research has addressed the invaluable semantic knowledge about XML data that is expressed by our types of cardinality constraints. Despite the time-consuming difficulty, we aim at pushing the expressiveness of these constraints while maintaining the tractability of their associated decision problems. Through our prior success with our approach [32, 33] we are confident to meet this challenge given sufficient time resources. We believe that the increasing amount of XML data and the vast potential outcomes justify such a thorough study.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

## 6. PROPOSED RESEARCH

**Cardinality constraints: Expressiveness vs. Tractability.** In the first phase we study cardinality constraints (CCs) independently of any specification such as a DTD [46] or XSD [48]. Our approach is based on the common representation of XML data as trees, as done by DOM [16], XSLT [50], and XML Schema [48]. Figure 1 shows such a representation in which nodes are annotated by their type: *E* for element, *A* for attribute, and *S* for text (PCDATA).

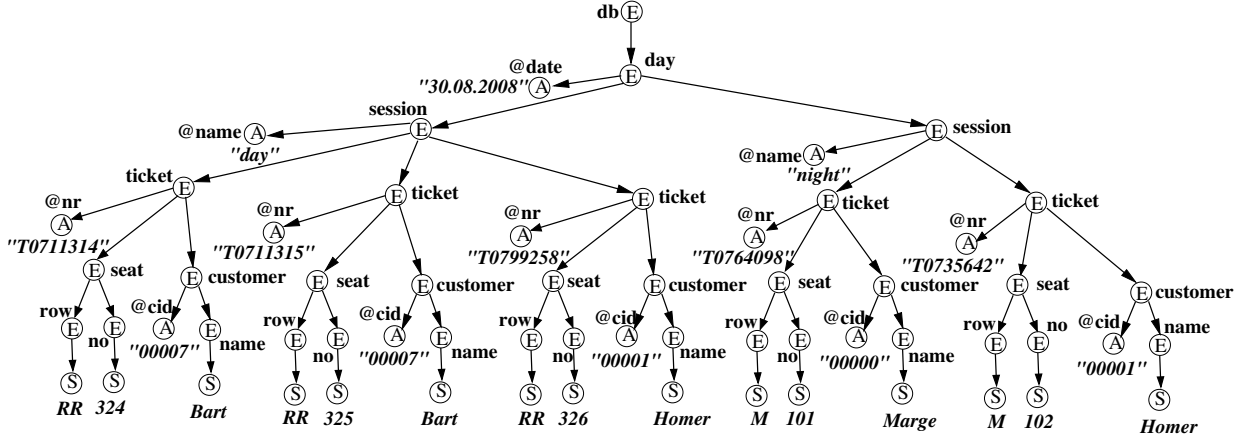


Figure 1: The XML tree (fragment)  $T$

The XML tree  $T$  in Figure 1 keeps information about ticket sales for some event. This data is subject to various business rules that can be expressed by CCs. These constraints restrict the number of nodes that have the same (complex) values on some selected subnodes. Such restrictions can be specified either for the entire tree or relative to selected subtrees. The following CC addresses every subtree rooted at a *session*-node: each *customer*/*@cid*-value that occurs under some *ticket*-node may occur as a *customer*/*@cid*-value under up to eight *ticket*-nodes in the same *session*-subtree. That is, each customer may purchase up to 8 tickets for the same session (we denote this CC by  $C1$ ). While  $T$  satisfies  $C1$  it does not satisfy the cardinality constraint that each customer may purchase only 1 ticket per session ( $C2$ ).

When defining cardinality constraints there are at least three factors that determine their expressiveness: i) the restrictions on the number of nodes (e.g. lower, upper bounds), ii) the navigational operators for accessing nodes (e.g. path expressions for descendants, ancestors) and iii) the notion of value-equal nodes (e.g. string equality on leaves, isomorphic subtrees). Previous work has focused on keys where nodes are accessed via child and descendant expressions, only an upper bound of fixed value 1 is permitted, and value-equality refers to isomorphic subtrees [8, 9, 32]. For  $T$  a key may express that the *seat*-values identify the *ticket*-nodes in each *session*-subtree. Our CCs include XML keys [32] and the *minOccurs*- and *maxOccurs*-attributes of XML Schema [23, 39, 48] as special cases. Those attributes restrict the number of a node's children independently of any present data (i.e. express purely structural knowledge). In addition, our CCs primarily restrict the number of nodes in XML subtrees based on the data on selected subnodes. We have only started to explore this semantic knowledge so far [33].

However, many (combinations of) sources of expressiveness are also sources of the undecidability or intractability of associated decision problems. The (finite) satisfiability problem asks for an arbitrary finite set  $\Sigma$  of constraints if there is a (finite) XML tree that satisfies all constraints in  $\Sigma$ . The (finite) implication problem asks for an arbitrary finite set  $\Sigma \cup \{\varphi\}$  of constraints whether every (finite) tree that satisfies all constraints in  $\Sigma$  also satisfies  $\varphi$ . If  $\Sigma$  consists of  $C1$  and the constraint that there are precisely two sessions per day ( $C3$ ), then  $\Sigma$  implies the constraint that each customer can purchase up to 16 tickets per day. However, the constraint that each customer can purchase at most 10 tickets per day is not implied by  $\Sigma$ , and needs to be specified additionally if it represents a desired business rule.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

A practically relevant and robust theory ought to balance expressiveness and tractability. Thus, we will *define expressive notions of CCs that also permit efficient solutions to their associated decision problems*. First, we adopt the path-based approach towards defining CCs, and then investigate which (combinations of) restrictions, navigational operators and notions of value-equality are *sources of intractability*. For instance, with isomorphic subtrees as the notion for value-equality the specification of both lower and upper bounds leads to *coNP*-hardness of the implication problem, even if nodes are accessed by *child*-navigation only [33]. An important step towards identifying manageable subclasses is the study of sound inference rules (preferably *axiomatisations*) for CCs. Such rule systems can reveal invaluable properties of CCs and can assist us in *designing efficient algorithms for deciding the implication problems*. We have applied path algebras [35] and methods from extremal graph theory [7] to XML keys [8, 9]. These techniques enabled us to characterise XML key implication in terms of reachability in a digraph [32], and to repair a flaw within the proposed XML key axiomatisation [9]. We also developed a quadratic time algorithm to decide implication [32], improving the so far best known heptic bound [9]. Some preliminary results show that these *techniques can be extended to achieve similar results for CCs* [33]. However, we depend on sufficient time resources to push their expressiveness while maintaining the tractability of the associated decision problems. We will guide our PhD student to *investigate these decision problems for cardinality constraints defined by **tree patterns*** instead of path expressions. So far, these patterns have only been applied to XML query processing [38] and functional dependencies for XML [36].

**Interactions with other classes of constraints.** A schema specification offers many advantages in processing, translating, querying, and storing XML data. We will extend our research on the satisfiability and implication problems for classes of cardinality constraints to the presence of common schema specifications, such as DTDs [2] and XSDs [48]. There are intricate interactions between the class of keys and foreign keys, and DTDs which lead to the undecidability of the satisfiability problem, and to intractability in the case of unary keys and foreign keys [3, 20]. Moreover, there are also interactions between CCs and DTDs. For instance, consider the two element declarations  $\langle \text{ELEMENT day}(\text{session}, \text{session}) \rangle$  and  $\langle \text{ELEMENT day}(\text{session}) \rangle$ , respectively. The first declaration implies *C3* (and vice versa), however, the second declaration contradicts *C3*, i.e., there is no XML tree that conforms to both the DTD and *C3*. Similarly, the use of *minOccurs*- and *maxOccurs*-attributes in an XSD such as

$$\langle \text{xs:element name="day"} \rangle \langle \text{xs:complexType} \rangle \dots \langle \text{xs:element name="session"} \rangle \text{type="sessionType"} \text{maxOccurs="1"} / \rangle \dots \langle / \text{xs:complexType} \rangle \langle / \text{xs:element} \rangle$$

may lead to non-trivial interactions. In fact, there is no XML tree that conforms to both the XSD fragment and *C3*. The objective is to *identify practically relevant classes of CCs and classes of DTDs (XSDs) whose associated decision problems are still tractable* (see [36] for interactions between DTDs and functional dependencies, FDs). Similarly, we want to *investigate the interaction between classes of cardinality constraints and other types of constraints*, such as FDs and inclusion constraints. We will benefit from our experiences in studying the interactions between CCs and FDs in Entity-Relationship models [25].

**XML database design.** Cardinality constraints can reveal important semantic information on the quality of the database layout with respect to update policies **and** query preferences. We can *develop efficient algorithms that apply database transformations known as **pivoting** [28] based on a given set of update policies (for some expressive class of updates) **and** a given set of query preferences (for some expressive class of queries)*. These mappings transform poorly-organised database structures into well-designed ones in order to avoid data redundancies, update anomalies, and inefficient query processing. A strategy that combines both update policies and query preferences has not been addressed yet in previous research, and is even a long-standing open problem in relational database design.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

For example, consider the cardinality constraint that on each day of an event there are precisely two sessions that allocate tickets to the same seat. Hence, the seat information is stored redundantly (twice a day). Consequently, the layout of the XML tree  $T$  in Figure 1 is not a particularly good choice if updates on the seat information are likely to occur frequently nor if queries are based on the seat information. Instead, one could store the information on each seat once together with the allocation of the seat to tickets per day. This makes updates of this particular kind easier, and processing of such specific queries more efficient. The small example illustrates that cardinality constraints with lower bounds are very important in this line of research since they point us towards sources of redundant XML data.

**Inconsistent XML databases.** In reality, most databases are inconsistent with respect to their schema or with respect to the constraints specified. For this reason, we will investigate two approaches towards dealing with inconsistencies based on CCs. In the first approach we will define *soft cardinality constraints* which also permit XML databases that violate a constraint [26, 27]. It is our goal to *acquire guidelines for dealing with the violation of CCs*. Such heuristics can range from merely detecting a violation and reporting it to the data administrator, to automatic repair actions. For example, the XML tree  $T$  in Figure 1 violates the CC  $C2$ . A possible repair is the removal of the subtree rooted at the second *ticket*-node.

Consistent query answering is an approach to querying inconsistent databases without repairing them first. When querying the XML database, only *certain answers* are retrieved. That is, answers that are present in all repairs of the XML database. For example, the following XPath query retrieves the name of all ticket holders for the night session of *30.08.2008*

```
/day[date="30.08.2008"]/session[name="night"]//name.
```

If the query is evaluated against  $T$ , then both *Bart* and *Homer* are returned. However,  $T$  is inconsistent with respect to the CC  $C2$ . For example, a repair could mean the removal of either offending *ticket*-subtree or the replacement of any offending occurrence of a customer name by a new name not occurring in the tree. In the first case there are two repairs for  $T$  and consistent query answering returns only *Homer* as a certain answer. In the second case there are infinitely many repairs for  $T$  and consistent query answering returns both *Bart* and *Homer*. We plan to *study different notions of repairs*, and based on these notions we want to *find expressive classes of cardinality constraints and expressive classes of query languages that still guarantee the tractability of consistent query answering* [13].

**XML query optimization.** Cardinality constraints provide additional semantic knowledge that can also be utilized for query optimization. For example, the following XQuery query retrieves the number of all those tickets purchased by the customer with *cid 00007* for either the *30.08.2008* or *01.09.2008* and which are for any seat in row *RR*.

```
for $t in doc("tickets.xml")/day[date="30.08.2008" or date="01.09.2008"]//ticket
where $t/seat/row="RR" and $t/customer/id="00007"
return <ticketnr>{$t/nr}</ticketnr>
```

The CCs may tell us that less tickets are selected based on the *cid*-information than based on the *seat*-information. Hence, the condition on the right of the **where**-clause should be processed before the condition on the left of the **where**-clause. We will *investigate the problem of optimizing a query (update, encryption) of some query (update, encryption) language fragment with respect to a class of CCs*. Efficient solutions to the implication problem enable us to apply not just explicitly but also implicitly specified constraints for optimization strategies. We aim at *developing efficient algorithms that compute bounds on the number of query answers, updates and encryptions/decryptions (of certain languages) without processing the underlying XML database*. In addition, we want to establish *new indexing techniques based on the knowledge represented by the semantics of cardinality constraints* [55].

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

### 6a. PROPOSED RESEARCH FOR EXTENDED PERIOD (OPTIONAL)

If the applicant wishes the proposal to be considered for funding for an additional one or two years after the initial period, please indicate briefly how the research programme might be extended, and how long for. It is acknowledged that this may change as the research progresses. PLEASE USE A MAXIMUM OF HALF A PAGE.

The first three years of funding lay the foundations for reasoning about cardinality constraints, and apply them effectively to enhance efficient XML data processing, and to design high-quality XML databases. An **additional year of funding** enables us to address further significant research questions:

**Consistency Checking and Data Mining.** The specification of (cardinality) constraints restricts valid XML documents to those that are considered meaningful to the application at hand. We can *design algorithms that efficiently decide the consistency problem for cardinality constraints*, i.e., decide whether a given XML document satisfies a given cardinality constraint. This would refine and extend previous research on XML type checking [40] and XML keys [6]. It seems natural to *apply tree automata [14] to verify the consistency with respect to both a cardinality constraint and a schema specification*.

In database design it is challenging to identify the meaningful (cardinality) constraints that ought to be specified. An approximation of such constraint sets can be obtained by applying data mining algorithms for discovering those constraints that are satisfied by typical XML documents. We will therefore *design algorithms that compute a **cover** for the cardinality constraints that are satisfied by a given XML tree*, i.e., a set of cardinality constraints that implies precisely those cardinality constraints satisfied by the tree. The findings that address this problem for XML keys [22] present a good starting point for this research direction.

**Propagation of cardinality constraints.** XML data is frequently mapped into database instances of different data models, or vice versa. This is the case, for instance, in data exchange/integration or in the definition of database views. Since cardinality constraints reveal important semantic knowledge on the properties of XML data it is desirable that this knowledge is preserved during such transformations. We will *investigate how database (schema) transformations between XML and other data models can be refined with the support of cardinality constraints*. This would substantiate the fact that our results do not only apply to XML databases, but also to databases of different data models.

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

## 7. REFERENCES

- [1] XML Applications and Initiatives - Contents Listing for XML Applications and Industry Initiatives, <http://xml.coverpages.org/xmlApplications.html>, retrieved January 11, 2006.
- [2] S. Abiteboul, P. Buneman, D. Suciu: *Data on the Web - From Relations to Semistructured Data and XML*, Morgan Kaufmann Publishers, 2000.
- [3] M. Arenas, W. Fan, L. Libkin: What's hard about XML schema constraints?, *DEXA*, pp. 269-278, 2002.
- [4] M. Arenas, L. Libkin: A normal form for XML documents, *PODS*, pp. 85-96, 2002.
- [5] M. Arenas, L. Libkin: An information-theoretic approach to normal forms for relational and XML data. *Journal of the ACM* 52, pp. 246-283, 2005.
- [6] B. Bouchou, M. Halfeld Ferrari Alves, M.A. Musicante: Tree Automata to Verify XML Key Constraints. *WebDB*, pp. 37-42 2003.
- [7] B. Bollobas: *Extremal graph theory*. Dover Publications, 2004.
- [8] P. Buneman, S. Davidson, W. Fan, C. Hara, W. Tan: Keys for XML. *Computer Networks* 39, pp. 473-487, 2002.
- [9] P. Buneman, S. Davidson, W. Fan, C. Hara, W. Tan: Reasoning about keys for XML. *Information Systems* 28, pp. 1037-1063, 2003.
- [10] P. Buneman, W. Fan, J. Siméon, S. Weinstein: Constraints for semi-structured data and XML. *SIGMOD Record* 30, pp. 47-54, 2001.
- [11] P. Buneman, W. Fan, S. Weinstein: Path constraints in semistructured databases. *Journal of Computer and System Sciences* 61, pp. 146-193, 2000.
- [12] S. Ceri, P. Fraternali, S. Paraboschi. XML: Current developments and future challenges for the database community. *EDBT*, 3-17, 2000.
- [13] J. Chomicki: Consistent Query Answering: Five Easy Pieces. *ICDT*, pp. 1-17, 2007.
- [14] H. Comon et al.: *Tree Automata Techniques and Applications*, <http://www.grappa.univ-lille3.fr/tata/>, 2002.
- [15] A. Deutsch, V. Tannen: Containment and integrity constraints for XPath. *KRDB*, 2001.
- [16] V. Apparao et al.: *Document object model (DOM) Level 1 specification*, *W3C Recommendation*. <http://www.w3.org/TR/REC-DOM-Level-1/>, Oct. 1998.
- [17] D. Embley, W. Y. Mok: Developing XML documents with guaranteed good properties, *ER*, pp. 426-441, Springer, 2001.
- [18] D. Embley, S. Liddle, S. Woodfield: Cardinality constraints in semantic data models. *Data and Knowledge Engineering* 11, pp. 235-270, 1993.
- [19] W. Fan: XML constraints. In *DEXA Workshops*, pp. 805-809, 2005.
- [20] W. Fan, L. Libkin: On XML integrity constraints in the presence of DTDs, *Journal of the ACM* 49, pp. 368-406, 2002.
- [21] W. Fan, J. Siméon: Integrity constraints for XML. *Journal Computer and System Sciences* 66, pp. 254-291, 2003.
- [22] G. Grahne, J. Zhu: Discovering approximate keys in XML data. *CIKM*, pp. 453-460, 2002.
- [23] W. Gelade, W. Martens, F. Neven: Optimizing Schema Languages for XML: Numerical constraints and Interleaving. *ICDT*, pp. 269-283, Springer, 2007.
- [24] S. Hartmann. On the consistency of int-cardinality constraints. *ER*, pp. 150-163, Springer, 1998.
- [25] S. Hartmann. On the implication problem for cardinality constraints and functional dependencies. *Annals of Mathematics and Artificial Intelligence* 33, pp. 253-307, 2001.
- [26] S. Hartmann. Coping with inconsistent constraint specifications. *ER*, pp. 241-255, Springer, 2001.
- [27] S. Hartmann. Soft Constraints and Heuristic Constraint Correction in Entity-Relationship Modelling. *Semantics in Databases 2001* pp. 82-99 Springer 2001

Standard Proposal	Contact PI's surname Link	Initials S	Application Number 08-MAU-095	Panel MIS
-------------------	---------------------------	------------	-------------------------------	-----------

- [28] S. Hartmann: Decomposing relationship types by pivoting and schema equivalence. *Data Knowl. Eng.* 39(1), pp. 75-99, 2001.
- [29] S. Hartmann, S. Link: More functional dependencies for XML, *ADBIS*, pp. 355-369, Springer, 2003.
- [30] S. Hartmann, S. Link, K.-D. Schewe: Functional dependencies over XML Documents with DTDs, *Acta Cybernetica* 17, 2005.
- [31] S. Hartmann, T. Trinh: Axiomatising functional dependencies for XML with frequencies. *FoIKS*, pp. 159-178, Springer, 2006.
- [32] S. Hartmann, S. Link: Unlocking keys for XML trees. *ICDT*, pp. 104-118, Springer, 2007.
- [33] S. Hartmann, S. Link: Numerical keys for XML. *WoLLIC*, pp. 203-217, Springer, 2007.
- [34] S. Hartmann, S. Link: Correspondences between nested database dependencies and fragments of propositional logic. *Annals of Pure and Applied Logic* 152(1-3), pp. 84-106, 2008.
- [35] D. Jungnickel: *Graphs, Networks and Algorithms*, Springer, 2002.
- [36] L. Kot, W. White: Characterization of the Interaction of XML Functional Dependencies with DTDs. *ICDT*, pp. 119-133, Springer, 2007.
- [37] M. Lenzerini, P. Nobili: On the satisfiability of dependency constraints in entity-relationship schemata. *Information Systems* 15, pp. 453-461, 1990.
- [38] G. Miklau and D. Suciu. Containment and equivalence for a fragment of XPath. *Journal of the ACM* 51, pp. 2-45, 2004.
- [39] F. Reuter, N. Luttenberger, *Cardinality Constraint Automata: A Core Technology for Efficient XML Schema-Aware Parsers*. <http://www.swarms.de/publications/cca.pdf>, 2003.
- [40] T. Schwentick: Automata for XML—A survey. *Journal of Computer and System Sciences* 73(3), pp. 289-315, 2007.
- [41] D. Seipel, U. Geske: Cardinality Constraints in Disjunctive Deductive Databases. *INAP*, 2000.
- [42] D. Suciu: On database theory and XML. *SIGMOD Record* 30, pp. 39-45, 2001.
- [43] B. Thalheim: *Dependencies in Relational Databases*. Teubner, 1991.
- [44] B. Thalheim: Fundamentals of cardinality constraints. *ER*, pp. 7-23, Springer, 1992.
- [45] B. Thalheim: *Entity-Relationship Modelling*. Springer, 2000.
- [46] T. Bray et al.: *Extensible Markup Language (XML) 1.0 (Fourth Edition) W3C Recommendation*. <http://www.w3.org/TR/xml/>, Aug. 2006.
- [47] J. Clark, S. DeRose: *XML Path Language (XPath) Version 1.0, W3C Recommendation*. <http://www.w3.org/TR/xpath>, Nov. 1999.
- [48] H. Thompson, D. Beech, M. Maloney, N. Mendelsohn: *XML Schema part 1: Structures second edition, W3C Recommendation*. <http://www.w3.org/TR/xmlschema-1>, Oct. 2004.
- [49] S. Boag et al.: *XQuery 1.0: An XML Query Language, W3C Recommendation*. <http://www.w3.org/TR/xquery/>, Jan. 2007.
- [50] M. Kay: *XSL Transformations (XSLT) Version 2.0 W3C Recommendation*. <http://www.w3.org/TR/xslt20/>, Jan. 2007.
- [51] T. Trinh: Functional dependencies for XML - Axiomatisation and normal form in the presence of frequencies and identifiers, MSc thesis, Massey University, 2005.
- [52] V. Vianu: A web odyssey: From Codd to XML. *SIGMOD Record* 32, pp. 68-77, 2003.
- [53] M. Vincent, J. Liu, C. Liu. Strong functional dependencies and their application to normal forms in XML. *ACM Transactions of Database Systems* 29, pp. 445-462, 2004.
- [54] J. Widom. Data management for XML: Research directions. *Data Engineering Bulletin* 22, pp. 44-52, 1999.
- [55] K.F. Wong, J. Xu Yu, N. Tang: Answering XML Queries Using Path-Based Indexes: A Survey. *World Wide Web* 9(3), pp. 277-299, Springer, 2006.