

# Multidimensional Mapping and Indexing of XML

Michael G. Bauer, Frank Ramsak und Rudolf Bayer

TU München

## **Gliederung:**

1. Einleitung
2. Mehrdimensionales Mapping
3. Messungen
4. Zusammenfassung

Teile der Arbeit gefördert im DFG-Schwerpunktprogramm “Verteilte Verarbeitung und Vermittlung digitaler Dokumente (V3D2)”

# Einleitung

## Allgemeines Vorgehen bei XML-Speicherung in RDBMS:

- **Schreddern:** Zerteilen der XML-Dokumente in Fragmente vorgegebener Granularität
- **Speichern:** Speicherung der Fragmente in RDBMS mit entsprechendem Schema
- **Umschreiben:** Rewriting der XML-Queries nach SQL

## Der Klassiker: Edge Mapping

- Zwei Tabellen

*roots(docid, label)* und *edge(parentid, childid, label/value)*

- `<paper>`

`<title>Query Processing</title>`

`</paper>`

*roots:* (0, paper)

*edge:* (0,1, title)

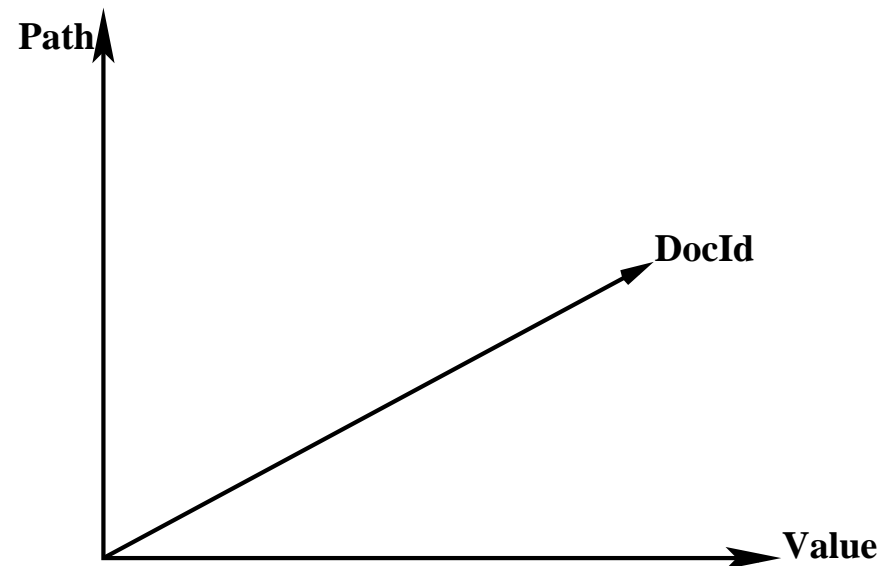
(1, NULL, 'Query Processing')

- Anfragen werden durch (u.U.) viele Self-Joins realisiert.

# Modell zum XML-Mapping

## Multidimensionaler Ansatz:

- Auftrennung eines XML-Dokuments in Pfade, Werte und DokumentenIds
- Betrachtung als Dimensionen mit jeweils einer eigenen Ordnung



## Ordnung und Reihenfolge

Bei der Speicherung:

- Reihenfolge bereits in DTD festgelegt
- Problem mehrfach gleicher Pfade  
Beispiel: <author><LN>
- Dokument muß korrekt restauriert werden

Bei den Anfragen: Festlegung des Kontexts

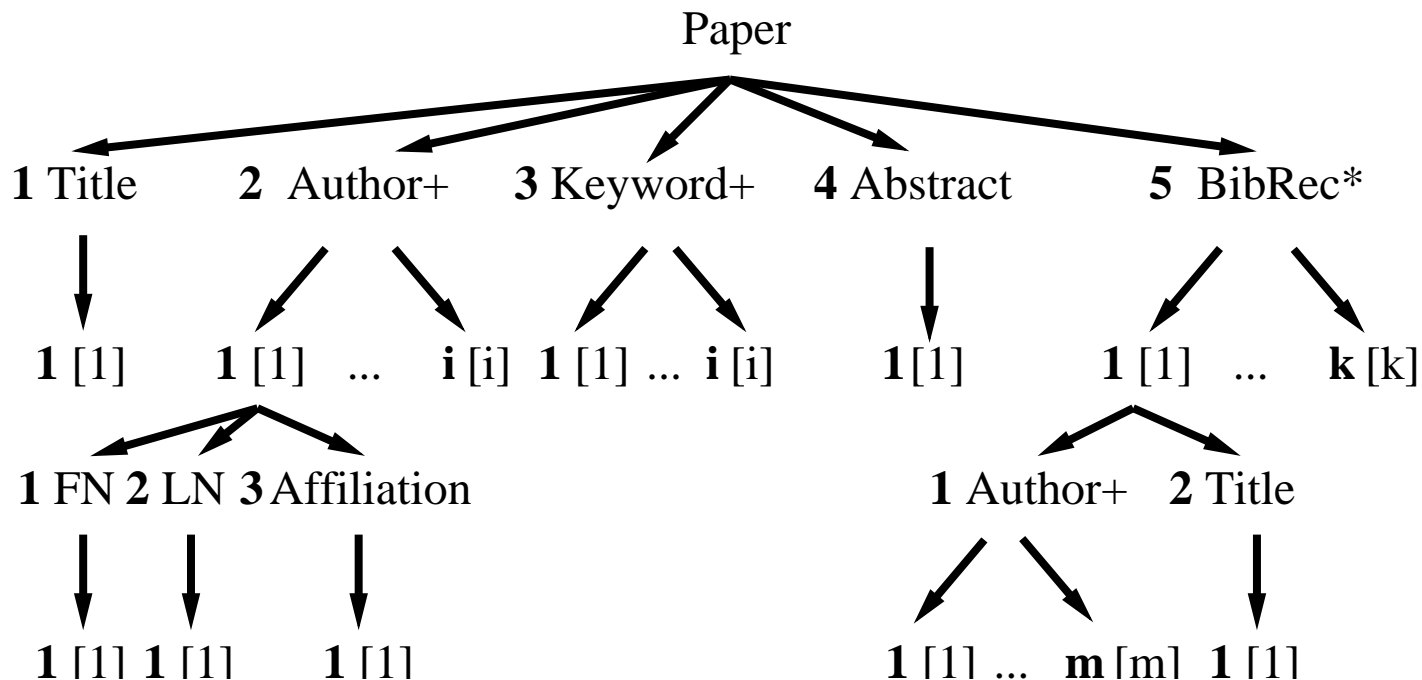
**Beispiel:** <author><FN>**Michael**  
                  <LN>**Bauer**   </author>  
          <author><FN>**Frank**  
                  <LN>**Ramsak** </author>

XPath: author[FN = "**Michael**" and LN = "**Bauer**"]

# Implementierung mittels Nummerierungsschema

Bitstrings (Surrogate) zur Repräsentation der Pfade

(Multidimensionales Hierarchisches Clustering, MHC)

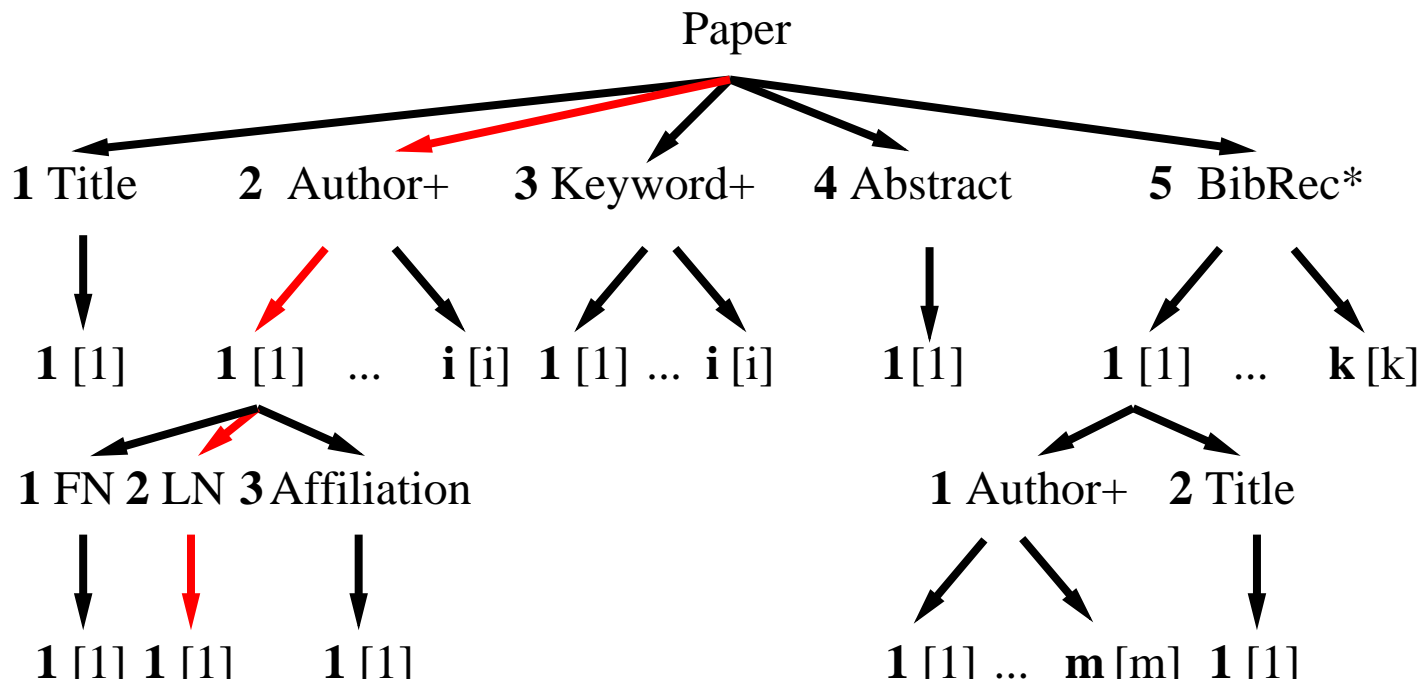


Beispiel: <author> [ 1 ] <LN> [ 1 ] entspricht: 2.1.2.1

# Implementierung mittels Nummerierungsschema

Bitstrings (Surrogate) zur Repräsentation der Pfade

(Multidimensionales Hierarchisches Clustering, MHC)



Beispiel: <author> [ 1 ] <LN> [ 1 ] entspricht: 2.1.2.1

# Implementierung

Typedim: Pfadtabelle, vollständige Pfade, sehr klein

Path	Surrogat
author[1]/FN[1]	00010 00001 00001 00001 00000 00000 00000 00000
author[1]/LN[1]	00010 00001 00010 00001 00000 00000 00000 00000

XMLtriple: Kerntabelle, trägt Information der XML-Dokumente

DocId	Surrogat	Value
1	00010 00001 00001 00001 00000 00000 00000 00000	Michael
1	00010 00001 00010 00001 00000 00000 00000 00000	Bauer

## Modell zur XML-Speicherung

```
for $b in document ( ' 'http://www3.in.tum.de' ' ) //paper
  where $b/author[FN=' 'Michael' ' and LN=' 'Bauer' ' ]
  return <result> $b/title $b/keyword </result>
```

### Definitionen:

1. **Selektion:** Ergebnis der Selektion sind Ids der Vaterknoten (DokumententIds) für die ein XPath-Prädikat in einem Dokument wahr evaluiert wird.
2. **Projektion:** Ergebnis der Projektion sind Teile von Dokumenten, die das XPath-Prädikat der Selektion erfüllen (referenziert über DokumententIds).

# Detaillierte Messungen

Datenbasis: 10000 XML-Dokumente, 400 verschiedene Autoren (max. 8 Autoren pro Papier), 50 MB Rohdaten.

System: Transbase Hypercube

4 verschiedene Indexe auf XMLtriple:

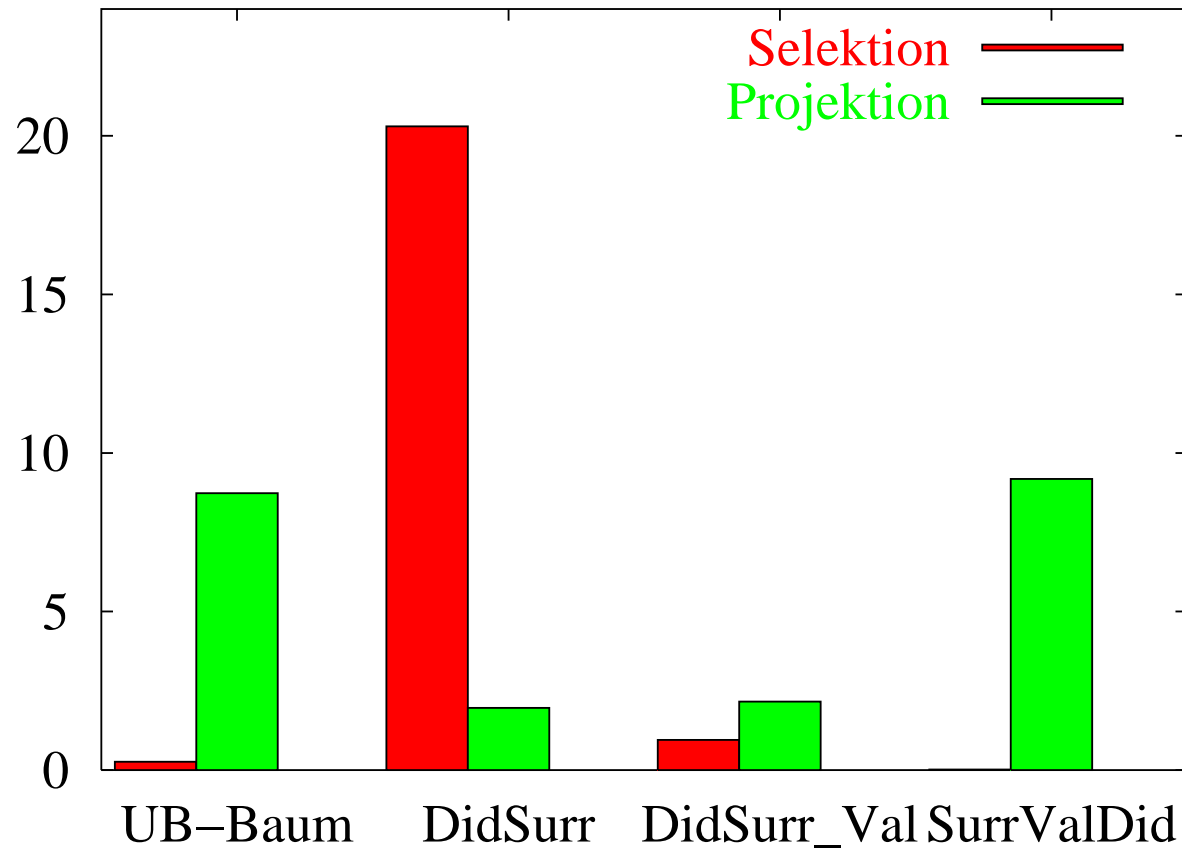
- UB-Baum: mehrdimensionaler, clusternder Index
- DidSurr, SurrValDid: Compound B-Bäume
- DidSurr\_Val: Compound B-Baum und Sekundärindex auf Value

Folgende XQuery-Anfrage:

```
for $b in document("http://www3.in.tum.de")//paper
where $b/author[FN="Michael" and LN="Bauer"]
return <result> $b/title $b/keyword </result>
```

# Detaillierte Messungen

Laufzeiten:



## Detaillierte Messungen

Seitenzugriffe: Selektion, Projektion, Seitengröße 2KB

Index	Selektion		Projektion		DB-Größe
	log. Dats.	phys. Seiten	log.Dats.	phys. Seiten	
UB-Tree	782	693	31043	6441	40 MB

## Detaillierte Messungen

Seitenzugriffe: Selektion, Projektion, Seitengröße 2KB

Index	Selektion		Projektion		DB-Größe
	log. Dats.	phys. Seiten	log.Dats.	phys. Seiten	
UB-Tree	782	693	31043	6441	40 MB
DidSurr	11946	11957	436	305	29 MB

## Detailierte Messungen

Seitenzugriffe: Selektion, Projektion, Seitengröße 2KB

Index	Selektion		Projektion		DB-Größe
	log. Dats.	phys. Seiten	log.Dats.	phys. Seiten	
UB-Tree	782	693	31043	6441	40 MB
DidSurr	11946	11957	436	305	29 MB
DidSurr_Val	705	1208	436	305	37 MB

## Detaillierte Messungen

Seitenzugriffe: Selektion, Projektion, Seitengröße 2KB

Index	Selektion		Projektion		DB-Größe
	log. Dats.	phys. Seiten	log.Dats.	phys. Seiten	
UB-Tree	782	693	31043	6441	40 MB
DidSurr	11946	11957	436	305	29 MB
DidSurr_Val	705	1208	436	305	37 MB
SurrValDid	22	36	463	468	24 MB

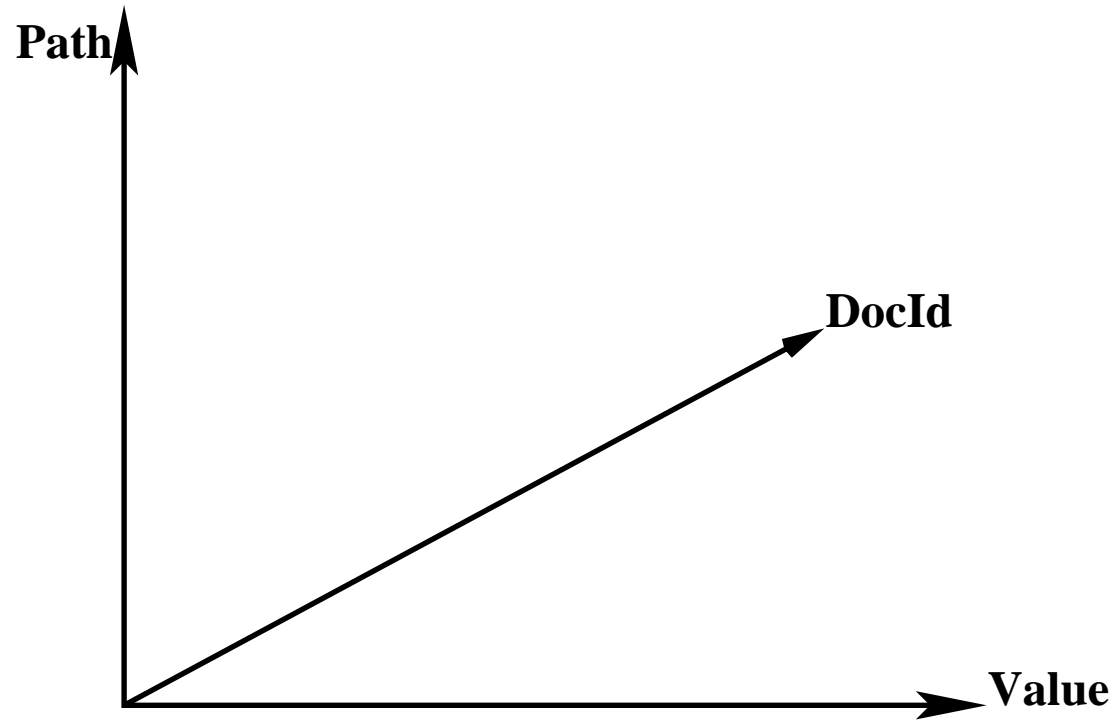
## Detaillierte Messungen

Seitenzugriffe: Selektion, Projektion, Seitengröße 2KB

Index	Selektion		Projektion		DB-Größe
	log. Dats.	phys. Seiten	log.Dats.	phys. Seiten	
UB-Tree	782	693	31043	6441	40 MB
DidSurr	11946	11957	436	305	29 MB
DidSurr_Val	705	1208	436	305	37 MB
SurrValDid	22	36	463	468	24 MB
Edge_parentid	241118	49776	517	508	83 MB
Edge_sec.	7473	6526	2488	1353	188 MB

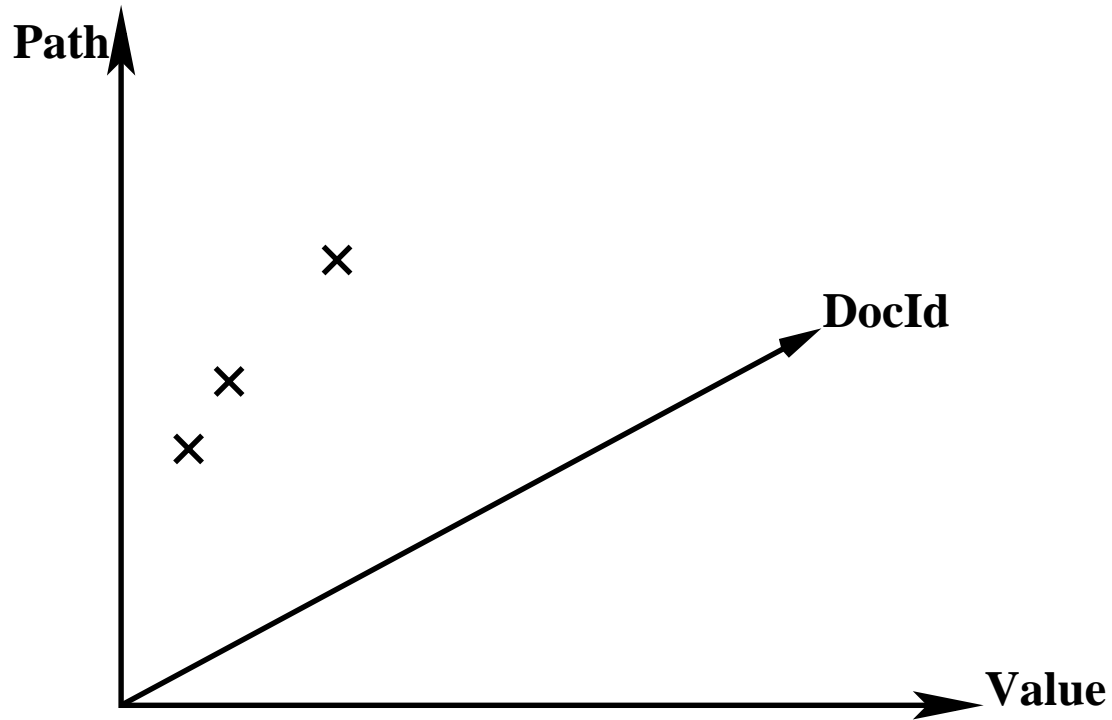
# Projektion

Vorgehensweise: Einschränkung auf Pfad und DocId,  
Durchstich durch Value-Dimension



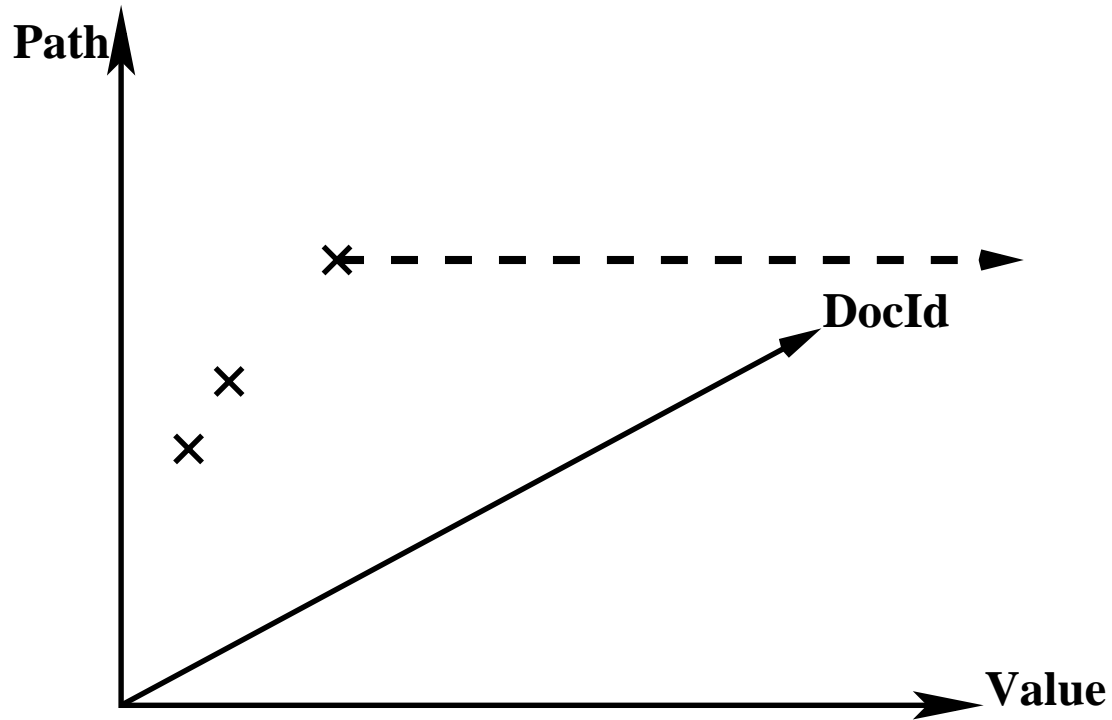
# Projektion

Vorgehensweise: Einschränkung auf Pfad und DocId,  
Durchstich durch Value-Dimension



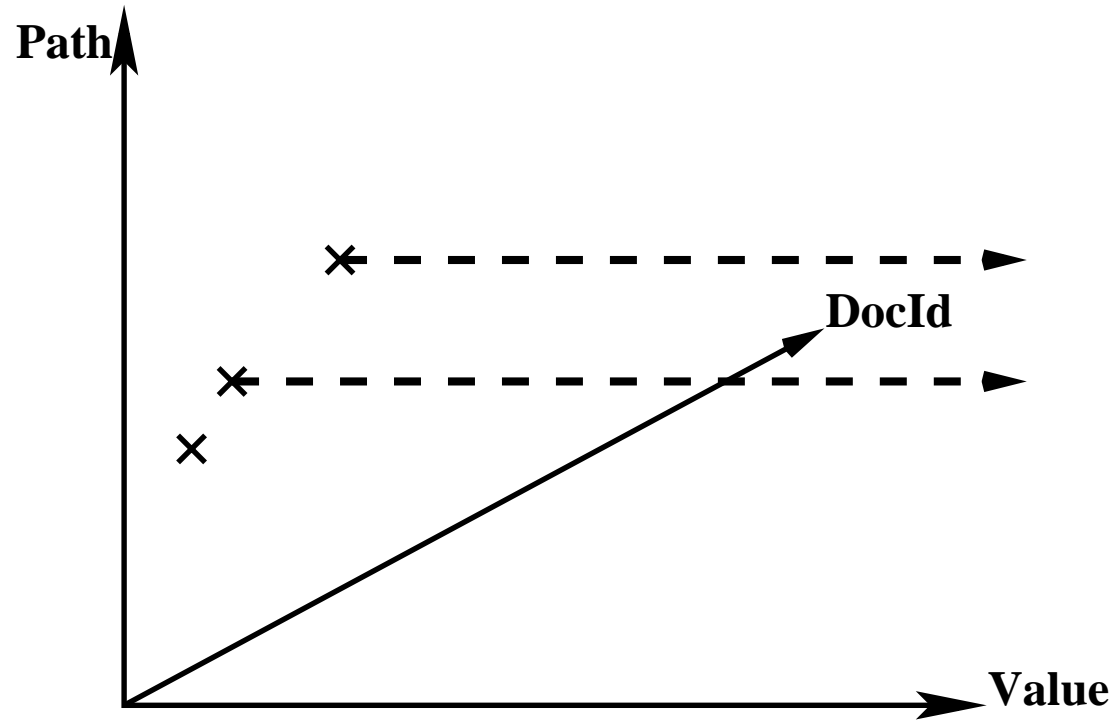
# Projektion

Vorgehensweise: Einschränkung auf Pfad und DocId,  
Durchstich durch Value-Dimension



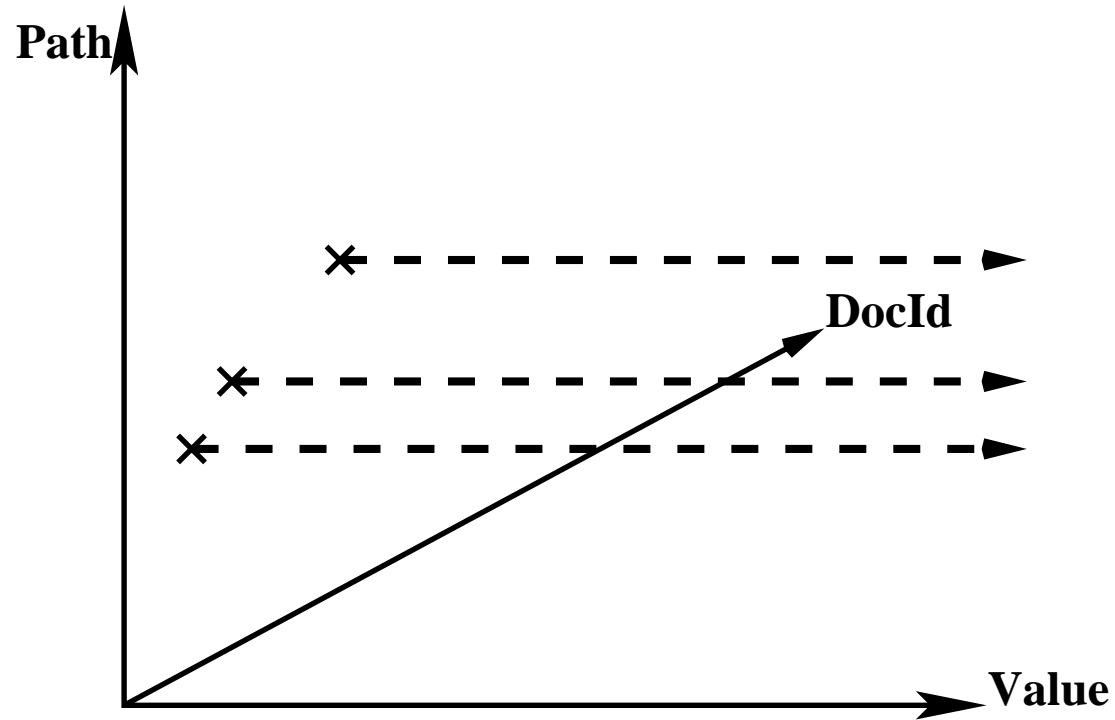
# Projektion

Vorgehensweise: Einschränkung auf Pfad und DocId,  
Durchstich durch Value-Dimension



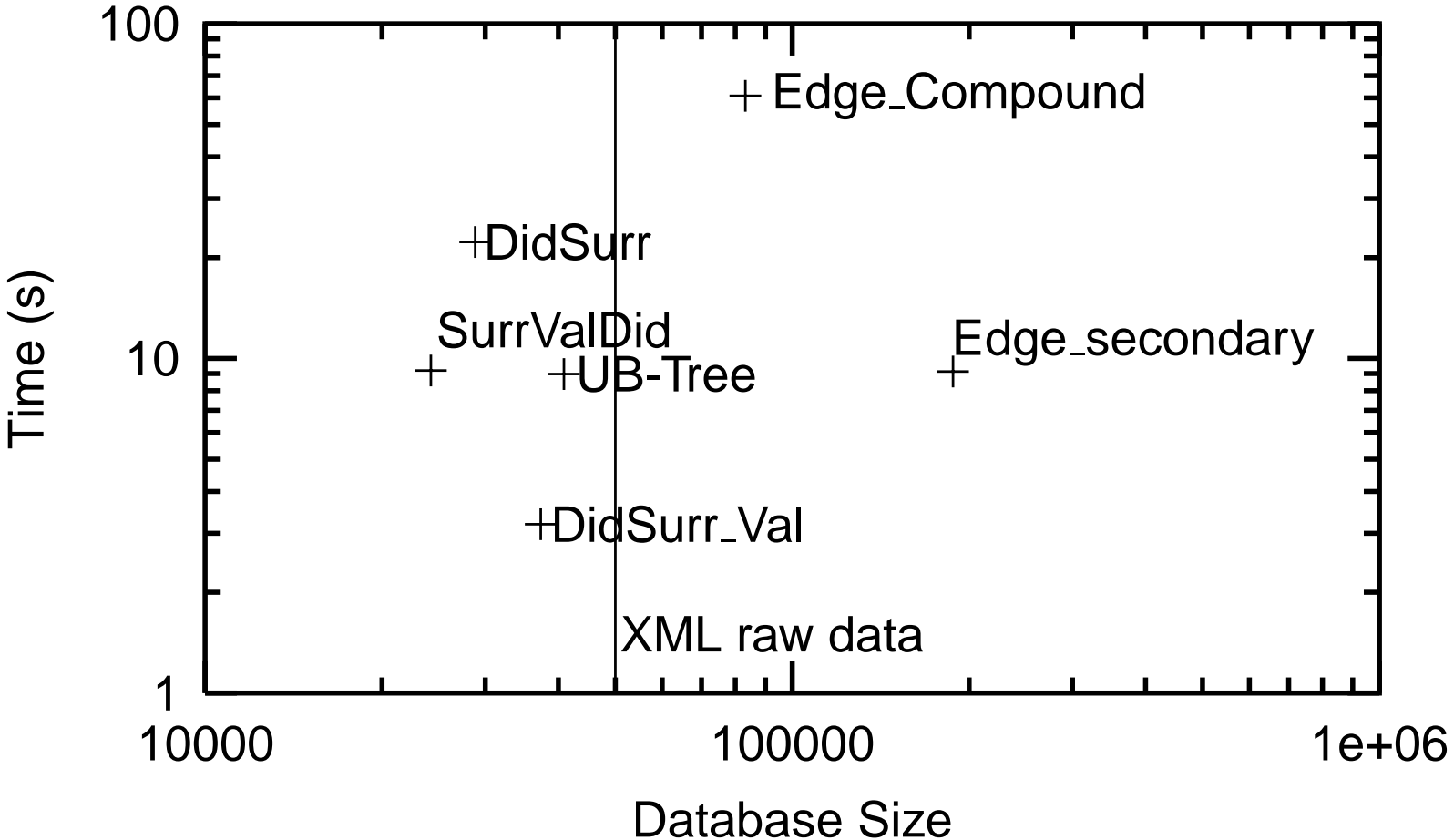
# Projektion

Vorgehensweise: Einschränkung auf Pfad und DocId,  
Durchstich durch Value-Dimension



# Gesamtübersicht Messungen

Gegenüberstellung Laufzeit / Datenbankgröße:



# Zusammenfassung

- XML-Mapping und Indexierung als multidimensionales Problem
- MHC als Nummerierungsschema für XML-Dokumente (kompakt und ordnungserhaltend)
- Interessante Kandidaten DidSurr\_Val und SurrValDid:
  - Primärindex auf Struktur, Sekundärindex auf Werte
  - Nur Primärindex mit Bevorzugung der Pfade
- Starke Dualität Struktur / Werte
- Ausblick: Abgrenzung zu Speziallösungen