

Dependable Adaptive Information Systems – The Big Picture

Theo Härder

www.haerder.de

Layer model for DBMS architecture and its evolution

DAIS – the layer stack

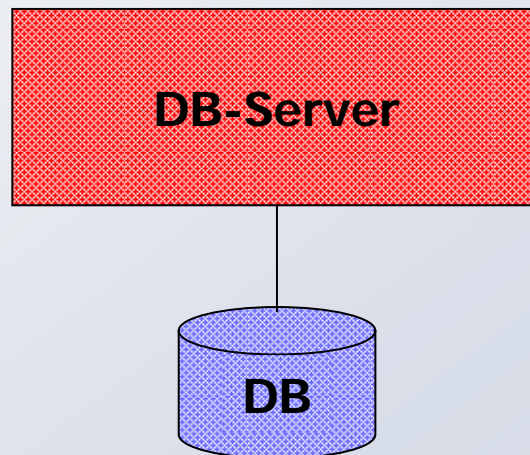
Labeling and indexing of XML tree nodes

Path processing algorithms

Architecture of DAIS

Data Independence and Evolution – the Grand Challenges for the DBMS Architecture

- Relational model
 - declarative set-oriented access and data independence
 - how can the postulated independence at the language and implementation levels be transformed into a system?
- Monolitical approach?
 - Q1: Select B.Title, P.Year, A.Name
From Books B, Authors A
Where B.Author = A.Author
And A.Name = „S*“ And B.Topic = „DBMS“



- interface to external memory: read and write of pages
(DB is a very long bit string!)

Database Systems in Business, Technology and Web, March 1985, Karlsruhe

Layer Model for DBMSs

DAIS – the layer stack

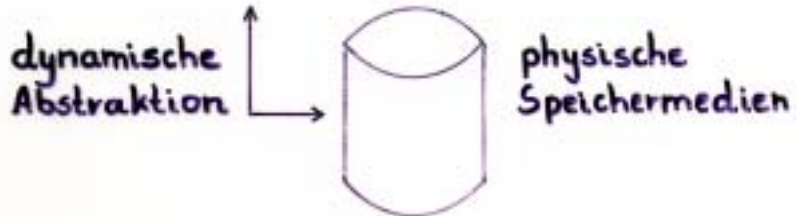
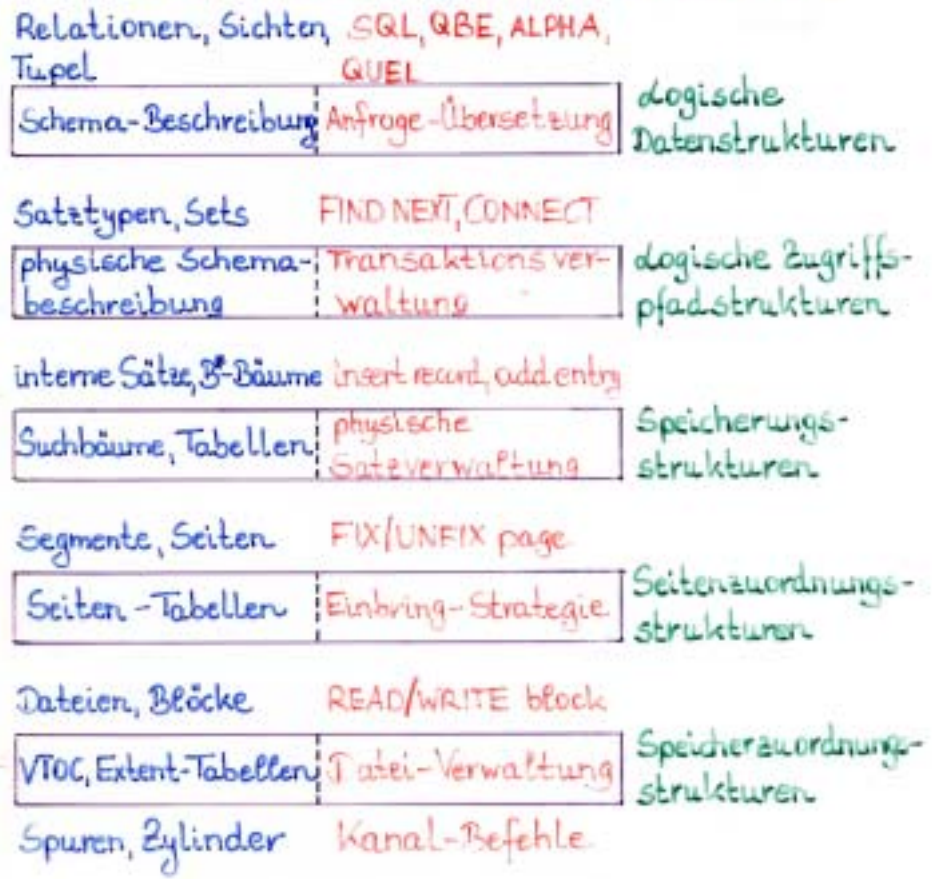
Labeling of XML Tree Nodes

XML Indexing

Path Processing Algorithms

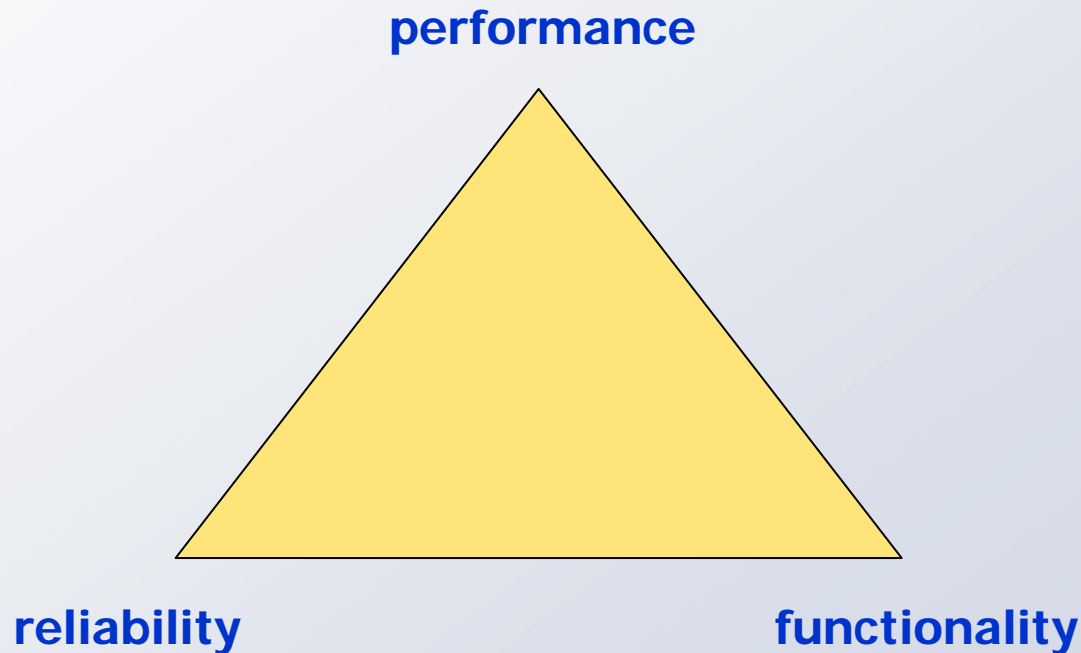
Architecture of DAIS

Schichtenarchitektur eines DBMS



The Magic Triangle

- “Classical” key requirements for a DBMS



Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

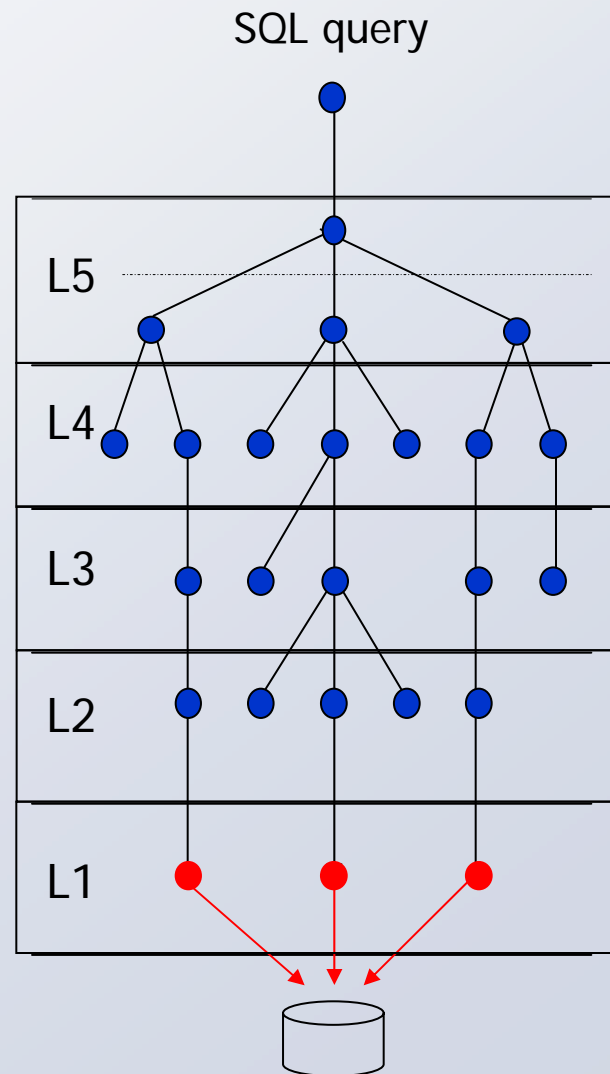
Path Processing
Algorithms

Architecture of
DAIS



Extensions and Optimizations in the Past 20 Years

- New storage types, LOBs
- Optimization by Moore's law
 - factor 10^4 , page size from 2K to 8–32K
 - LRU + reference density (LRU-K)
- New access paths?
 - adequate and integrated access paths in addition to the ubiquitous B-tree?
 - most important performance improvement by fine-grained locking
- New algorithms
 - hash joins, arbitrary predicates, ...
 - shared use of scans, reuse of results
 - adaptivity to resource unavailability
- Compilation and Optimization (SQL4)
 - cost-based optimizers (histograms)
 - dynamic QEPs
 - "gambling"



Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

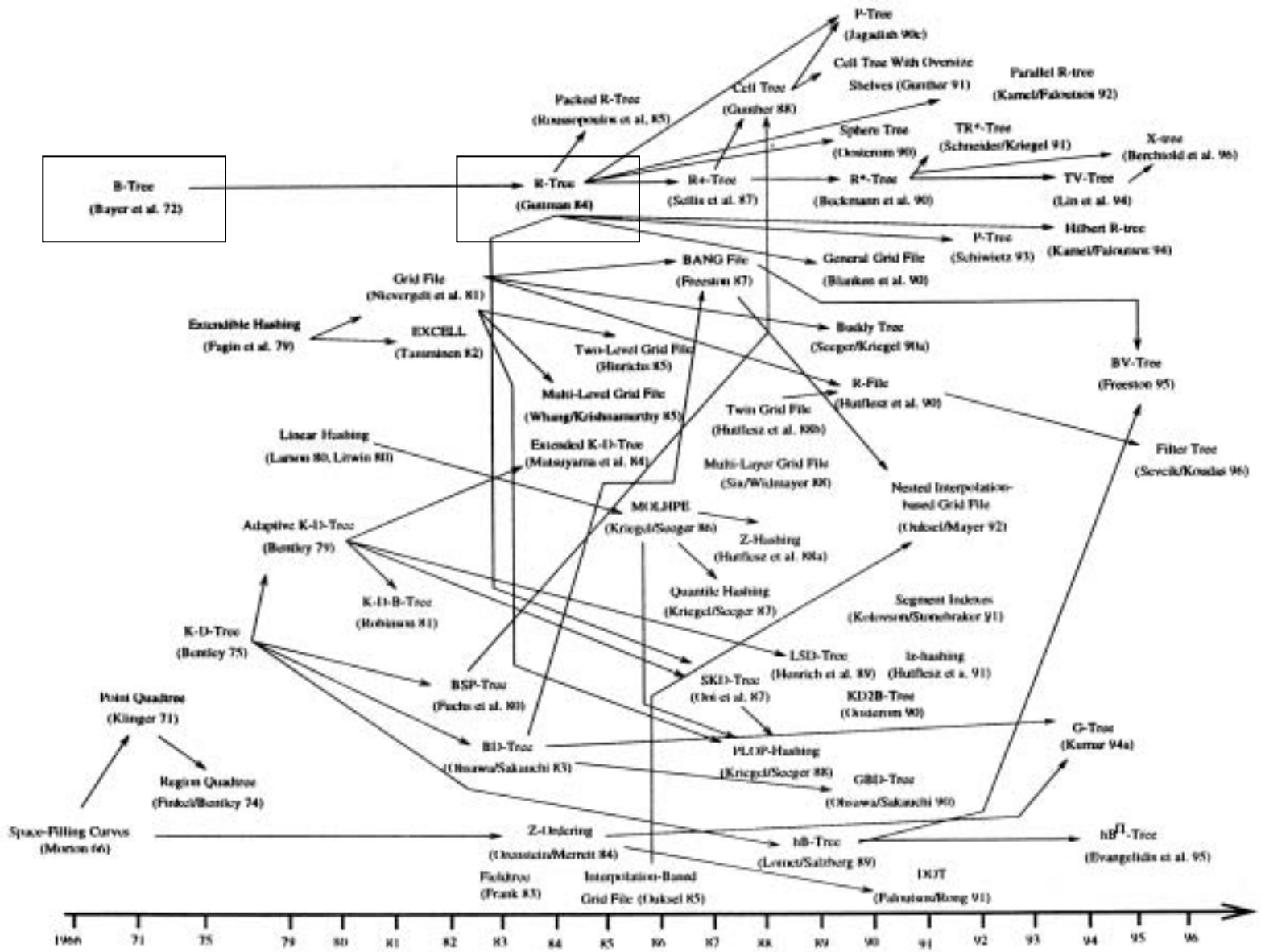
Path Processing
Algorithms

Architecture of
DAIS



Genealogy of Access Paths

- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



Transaction as Dynamic Control Structure

- **Atomicity**
“all or nothing” property of any DBMS action
- **Consistency and semantic integrity**
of the DB is endangered by erroneous data and operations of an application program
- **Isolated execution**
means “logical single-user mode”
- **Durability**
requires that modified data of successful transactions must survive any type of failure

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

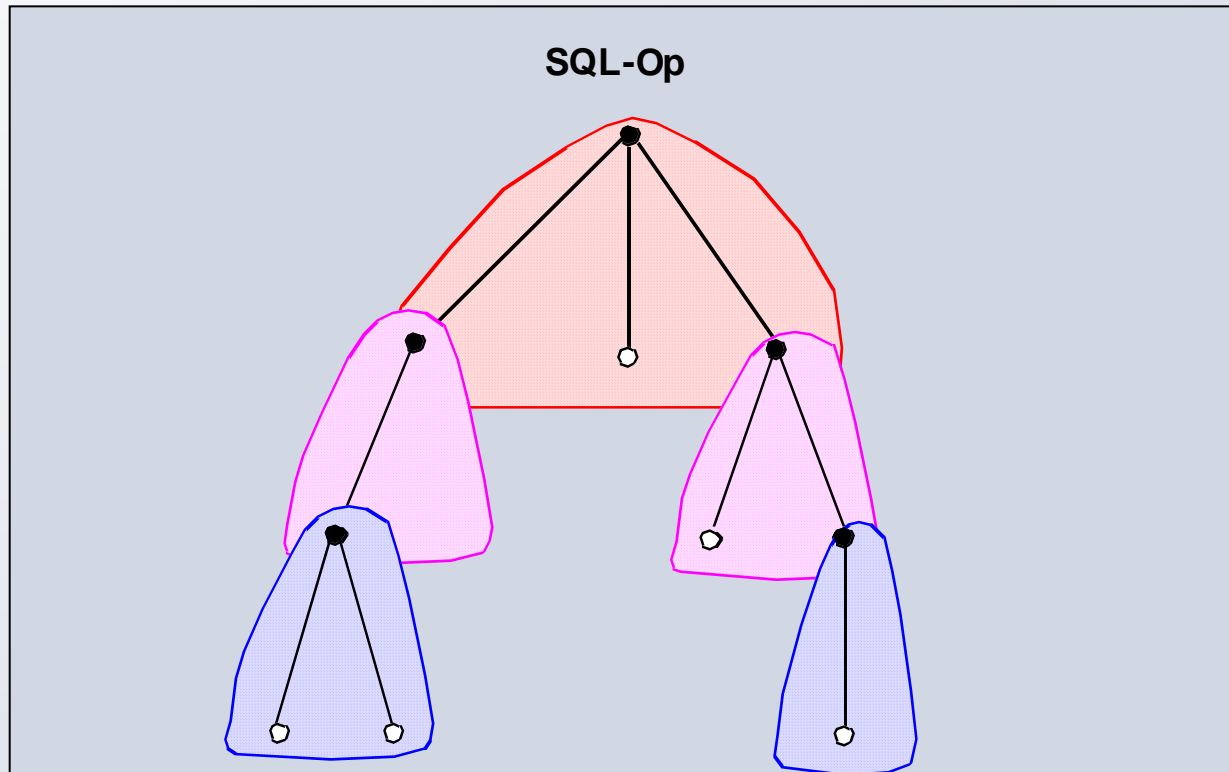
Path Processing
Algorithms

Architecture of
DAIS



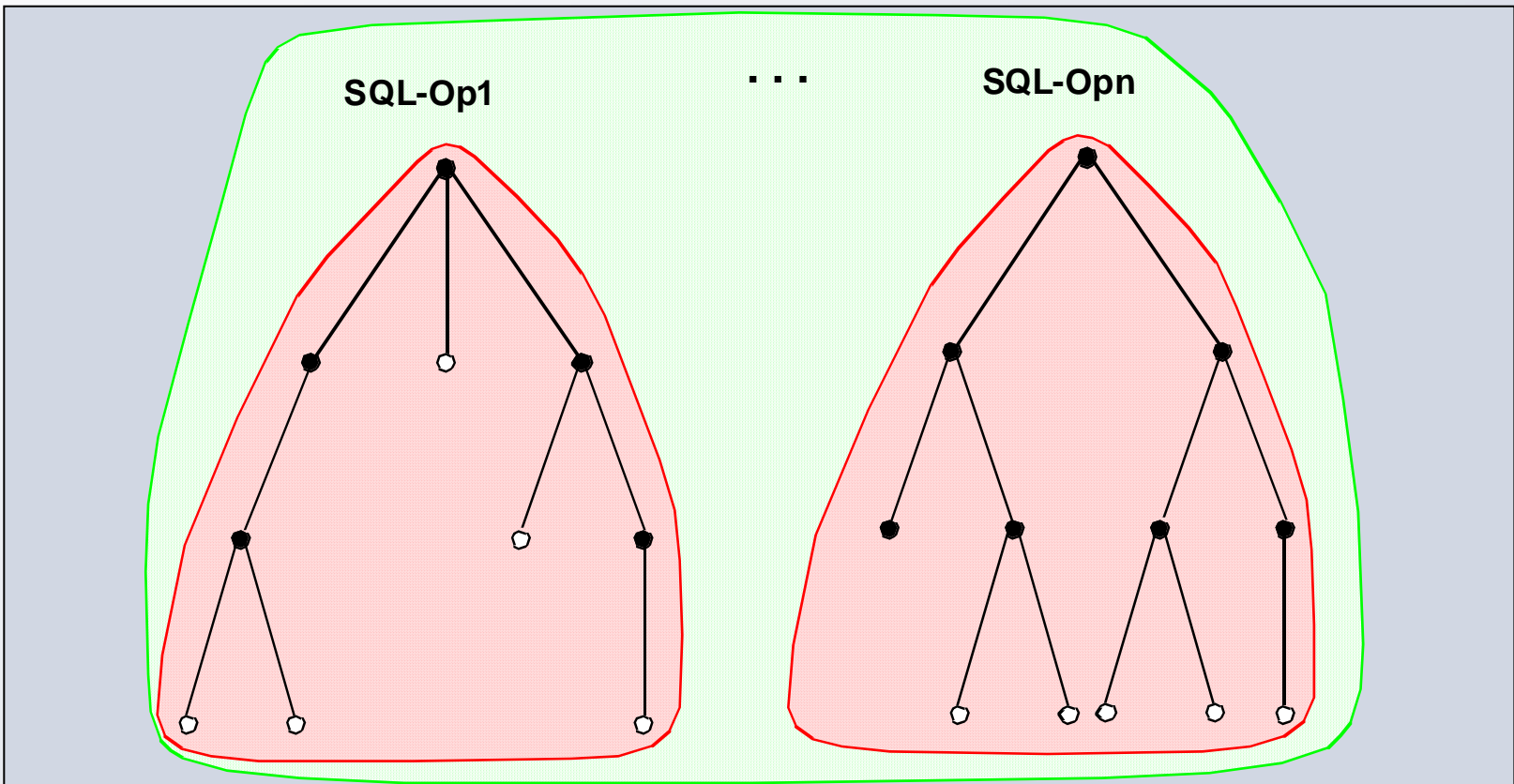
Atomicity is not a Natural Property of Computers

- Transactions are abstractions
 - building blocks: atomic actions (AAs)
 - AAs are abstractions, too
 - even if AAs would be implementable in an atomic way, hierarchy of AAs would not!



Statement and Transaction Atomicity

- Key mechanisms
 - concurrency control
 - logging and recovery

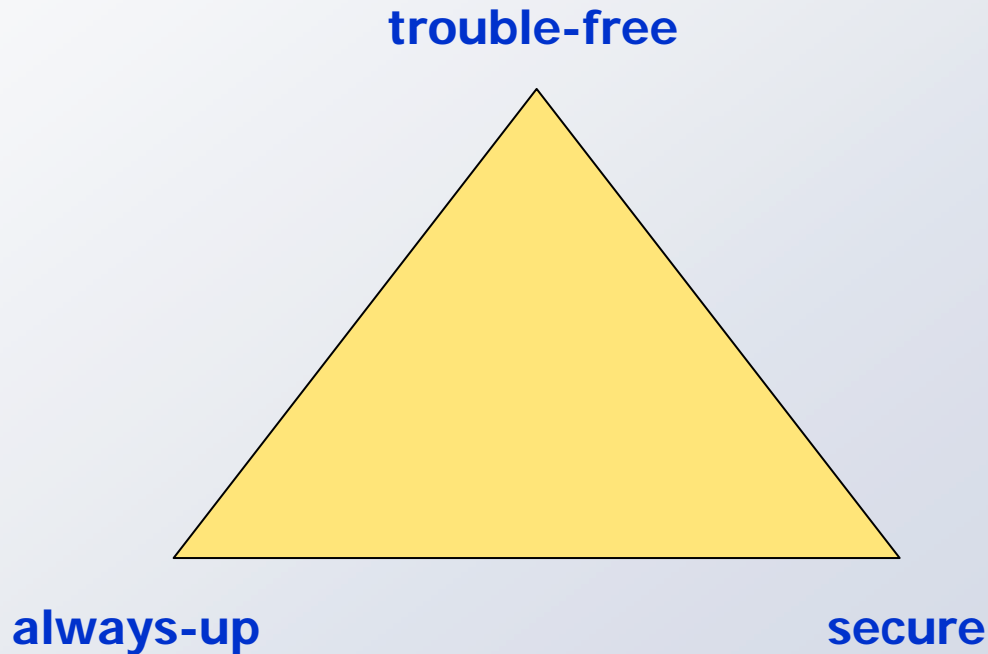


- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



Dependable Adaptive Information Systems

- Additional properties needed (Jim Gray)



Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



Example of an XML Document

- <bib>

```
<book year="1994" id="1">
  <title>TCP/IP Illustrated</title>
  <author>
    <last>Stevens</last>
    <first>W.</first>
  </author>
  <price>65.95</price>
</book>
<book year="2000" id="2">
  <title>Data on the Web</title>
  <author>
    <last>Abiteboul</last>
    <first>Serge</first>
  </author>
  <author>
    <last>Buneman</last>
    <first>Peter</first>
  </author>
  <author>
    <last>Suciu</last>
    <first>Dan</first>
  </author>
  <price>39.95</price>
</book>
<book year="1999" id="3">
  <title>The Economics of . . . </title>
  <editor>
    <last>Gerbag</last>
    <first>Darcy</first>
    <affiliation>CITI</affiliation>
  </editor>
  <price>129.95</price>
</book>
```

</bib>

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



Is the Layer Model Suitable for other Data Types?

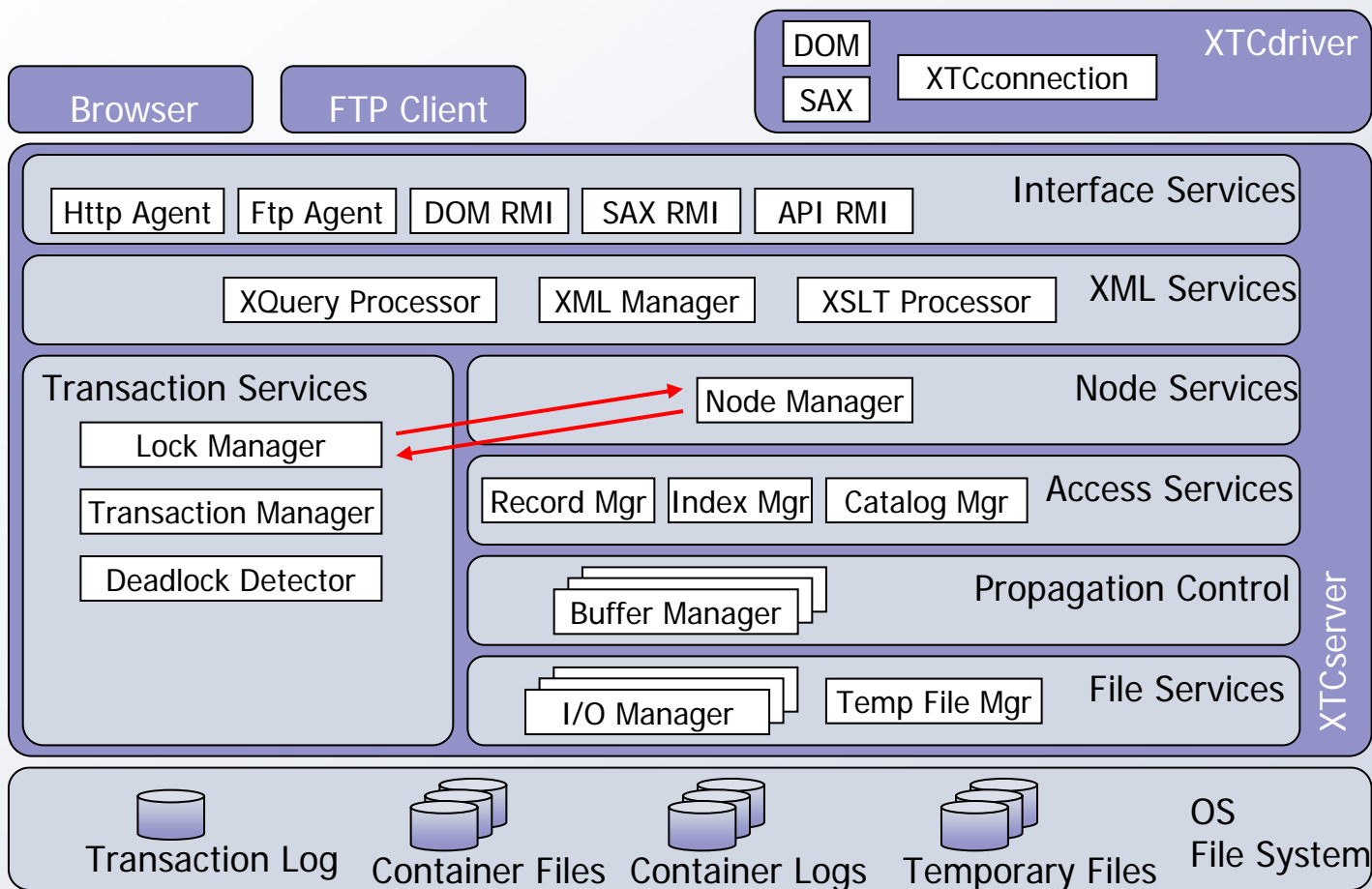
- Document order must be preserved:
 - round-trip property!
 - **node order matters!**
- Layer model
 - designed for declarative and set-oriented evaluation of records
 - hierarchical layers and information hiding guaranteed **long-term system evolution**
 - “self-^{*}” properties require **much more information flow across system layers**

Important observation:

The invariants in database management determine the mapping steps of the supporting architecture



A Native XDBMS Architecture



■ XTC – architectural overview

- reuse of the layer model is possible, but needs substantial adjustments and, in particular, new functionality in the higher layers



DAIS – The Layer Stack

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



Agreement Protocols	Business Logic	Application Layer
Extended	Control Flow Data Flow	WfMS
TA-Models	Distribution Caching	Middleware

Transaction Mgmt

Query Processing

C

Consistency Control	Compilation Optimization	Data System
---------------------	-----------------------------	----------------

I

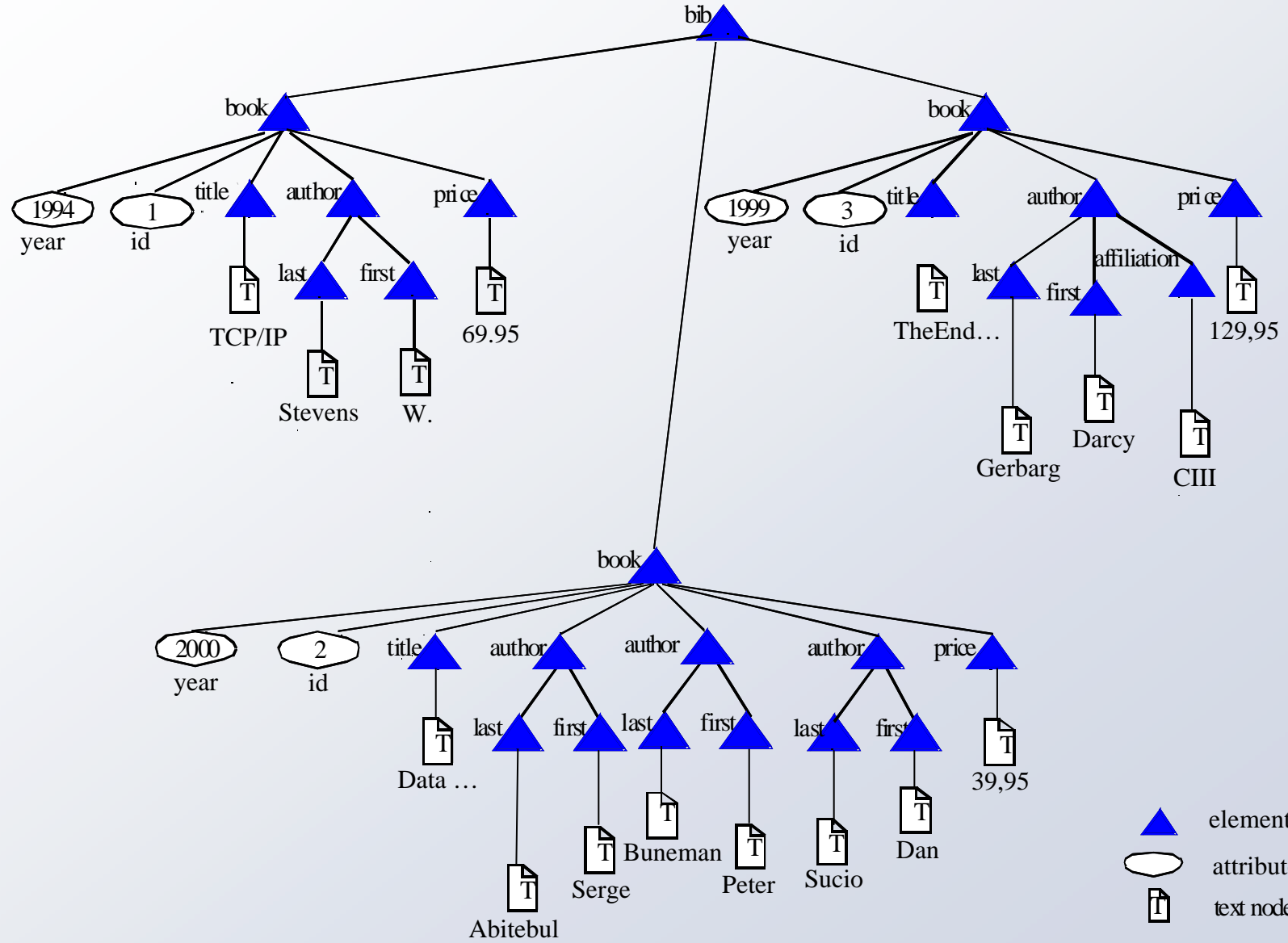
Transaction Services	Path Processing Algorithms	Access System
Concurrency Control	Document Representation Labeling, Indexing	

A

Logging / Recovery	Buffer Mgmt Propagation Control File Services	Storage System
-----------------------	---	-------------------

D

Example of a DOM Tree: Conceptual Representation



- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



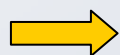
Holistic Support of all Internal XDBMS Operations

■ Node Labeling

- representation of an XML document: ordered, labeled tree with nodes of type element, attribute, text

■ Support of

- **declarative query processing**
 - all core operations
 - indexing support
- **navigational processing**
 - in combination with XML document representation and
 - additional access path structures
- **concurrency control**
 - most operations jump into the document tree
 - intention locks up to the document root required



without accessing the XML document on disk

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

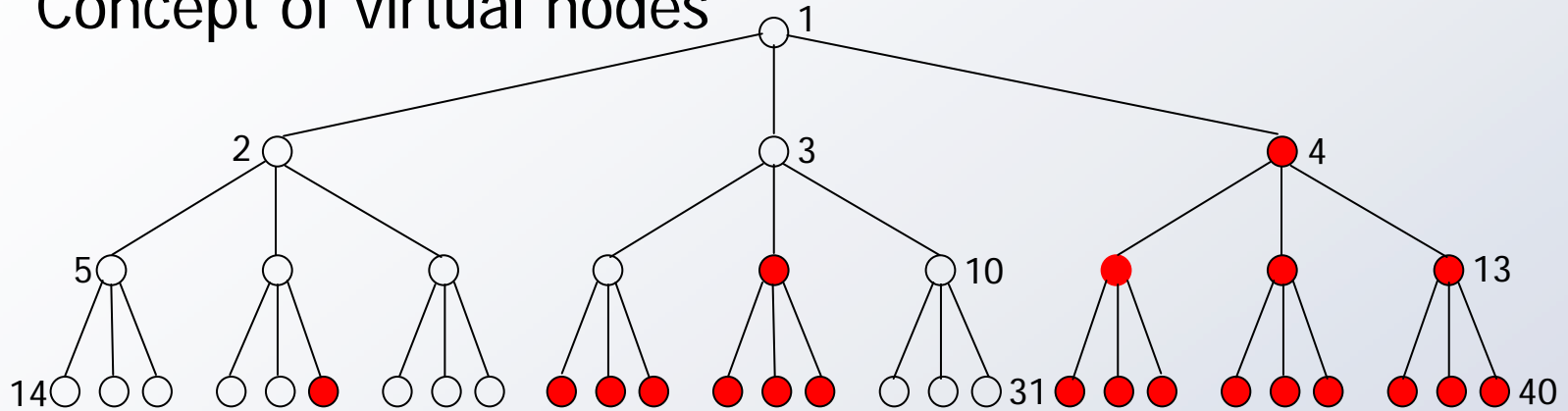
Path Processing
Algorithms

Architecture of
DAIS



Node Labeling – Use of k-ary Trees

■ Concept of virtual nodes



- $\text{parent}(cn, k) = \text{ceil}((cn - 1)/k)$
- $\text{child}(cn, k) = cn * k - (k-1) + 1, cn * k - (k-2) + 1, \dots, cn * k - 1 + 1, cn * k + 1$
- $\text{ancestor}(cn, k) = \text{parent}(cn, k), \text{parent}(\text{parent}(cn, k), k), \dots$
- $\text{descendant}(cn, k) = \text{child}(cn, k), \text{child}(\text{child}(cn, k), k), \dots$
- $\text{sibling}(cn, k) = \text{child}(\text{parent}(cn, k), k), \dots$
- previous/following ...

■ KO criterion

- any computed label may correspond to a virtual node
- tree representation has to be accessed to check whether a node is real or virtual



a document may have a very large k and very many levels

DeweyIDs Embody a Special Prefix Labeling Scheme

- Labels **must**
 - be **immutable** for the lifetime of the nodes
 - **preserve the document order**, when inserting new nodes
 - easily reveal the **level and the ID** for all ancestor nodes
- DeweyID consists of **several divisions** separated by dots

- overflow mechanism: **even** division values

$$d_1 = 1.3.17.2.2.3.4.9 \quad d_2 = 1.3.17.2.3.7$$

- level determination

$$d_1 = 1.3.17.2.2.3.4.9$$

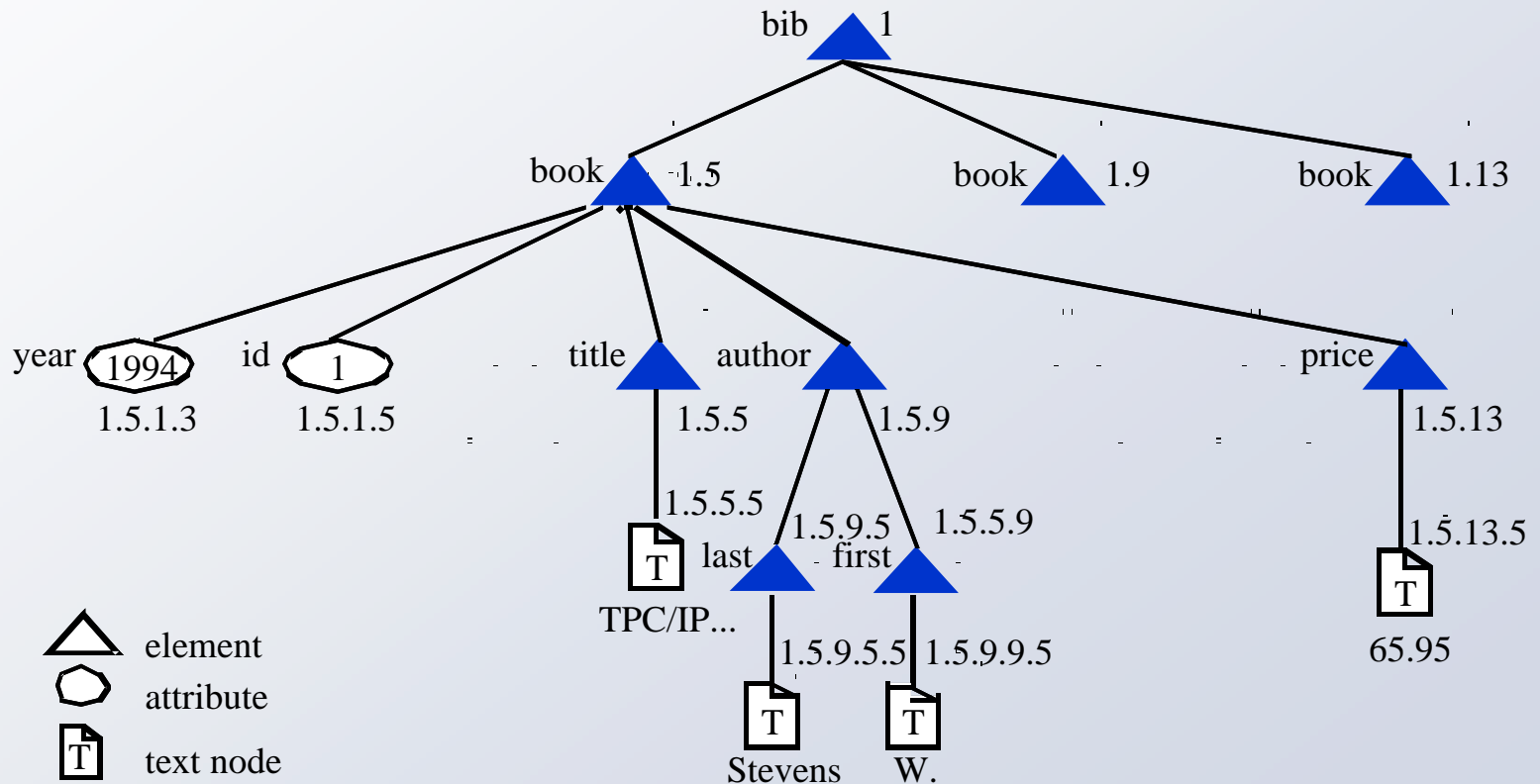
- ancestor IDs: $a_0 = 1$; $a_1 = 1.3$; $a_2 = 1.3.17$; $a_3 = 1.3.17.2.2.3$

- ordering d_2 ? d_1

$$d_1 < d_2 : 1.3.17.2.2.3.4.9 < 1.3.17.2.3.7$$

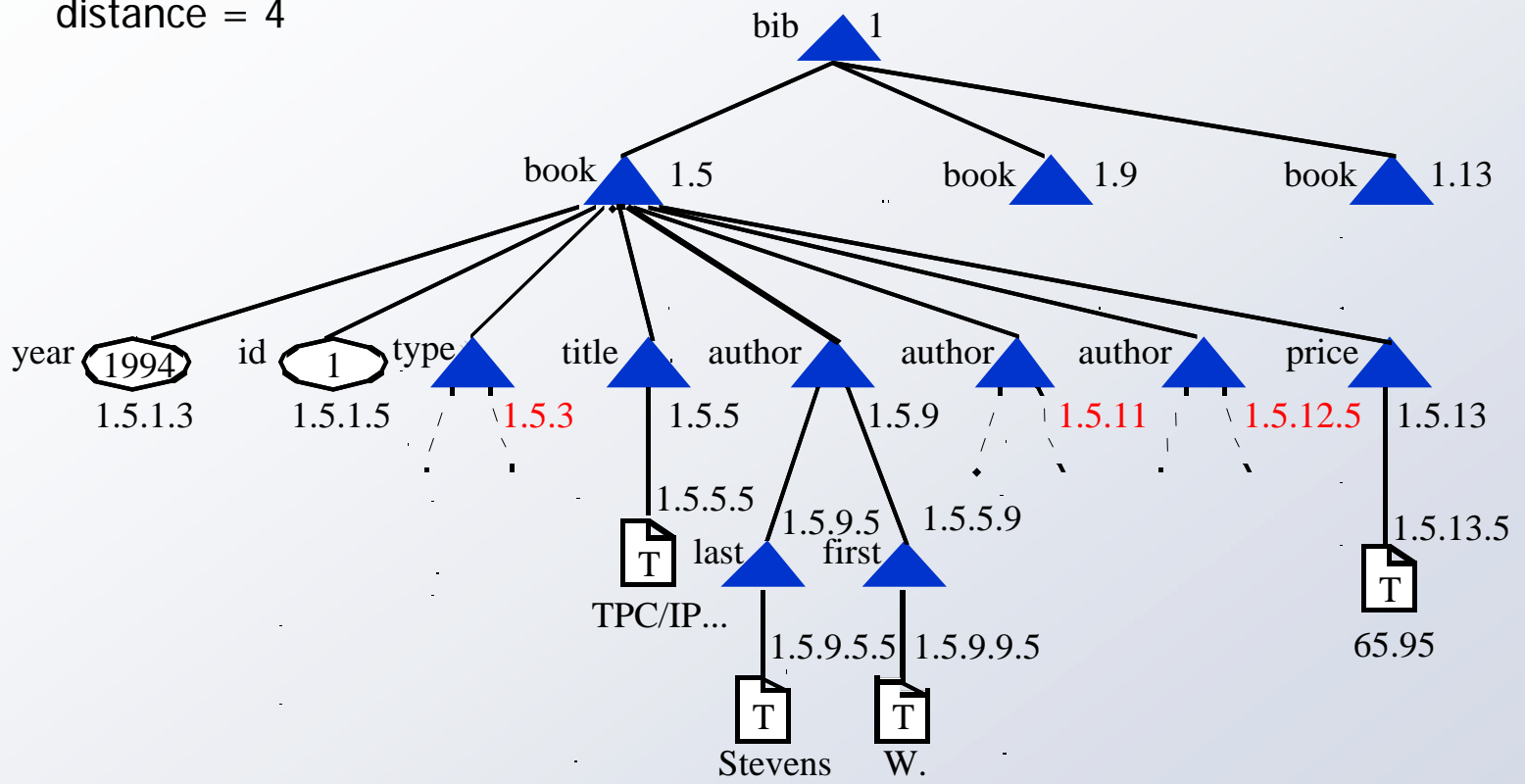
Initial Assignment of DeweyIDs

- assignment of division values is affected by parameter *distance* (= 4)
- on initial loading, only **odd** division values are assigned
- odd division value indicates **level transition**



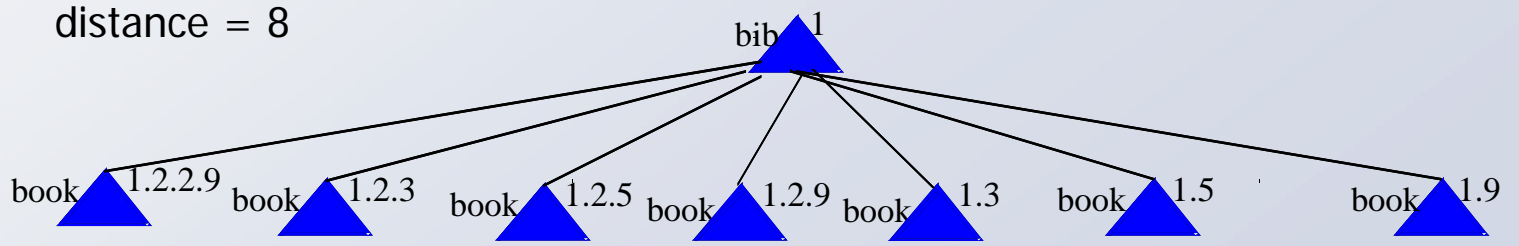
DeweyIDs: Insertion of Subtrees

distance = 4



worst-case considerations:

distance = 8



Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



Characteristics of XML Documents Considered

file name	description	size (bytes)	number of element nodes	number of text nodes	number of attributes	max. depth	∅-depth	max. fanout	∅-fanout of elements
1) treebank_e.xml	Encoded DB of English records of Wall Street Journal	86,082,517	2,437,666	1,391,845	1	37	8.44	56,385	1.58
2) nasa.xml	Astronomical data	25,050,288	476,646	303,676	56,317	9	6.08	2,435	1,76
3) psd7003.xml	DB of protein sequences	716,853,016	21,305,818	15,955,109	1,290,647	8	5.68	262,529	1.81
4) SwissProt.xml	DB of protein sequences	114,820,211	2,977,031	2,013,844	2,189,859	6	4.07	50,000	2.41
5) dblp.xml	Computer Science Index	284,994,162	6,662,623	6,013,355	1,375,832	7	3.39	649,080	2.11
6) customer.xml	Customers from TPC-H benchmark	515,660	13,501	12,000	1	4	3.41	1,501	1.89
7) ebay.xml	Ebay auction data	35,562	156	107	0	6	4.26	12	1.90
8) lineitem.xml	Line items from TPC-H benchmark	32,295,475	1,022,976	962,800	1	4	3.45	60,176	1.94
9) mondial-3.0.xml	Geographical DB of diverse sources	1,784,825	22,423	7,467	47,423	6	4.15	955	3.45
10) orders.xml	Orders from TPC-H Benchmark	5,378,845	150,001	135,000	1	4	3.42	15,001	1.90
11) uwm.xml	Courses of a University Website	2,337,522	66,729	40,234	6	6	4.37	2,112	1.91

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



DeweyIDs – Comparison of Avg. Sizes to Max. Sizes

Document	∅-size			max-size		
	dist(2)	dist(32)	dist(256)	dist(2)	dist(256)	dist(256)
1. tree-bank_e	6.67	11.57	15.94	22	46	72
2. nasa	5.19	8.54	11.30	8	13	18
3. psd7003	5.61	8.84	11.30	8	13	17
4. SwizzProt	5.10	7.04	8.14	8	11	13
5. dblp	4.58	6.12	7.16	7	10	13
6. customer	3.17	5.04	6.19	4	6	7

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS



XML Indexing Methods

■ Path indexing

- effective support for
 - query rewriting,
 - query pruning based on structural summaries, e.g., to check the existence of paths

■ Many types of path fragments in queries to be supported

- parent/child: `a/b/c`
- ancestor/descendants: `a//x//y`
- enhanced by predicates: `a/b[pred1]//x[pred2]`
- twigs: `a/b[.//c]//x[d]`

 What is the ubiquitous index – the silver bullet – for XML queries?

Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

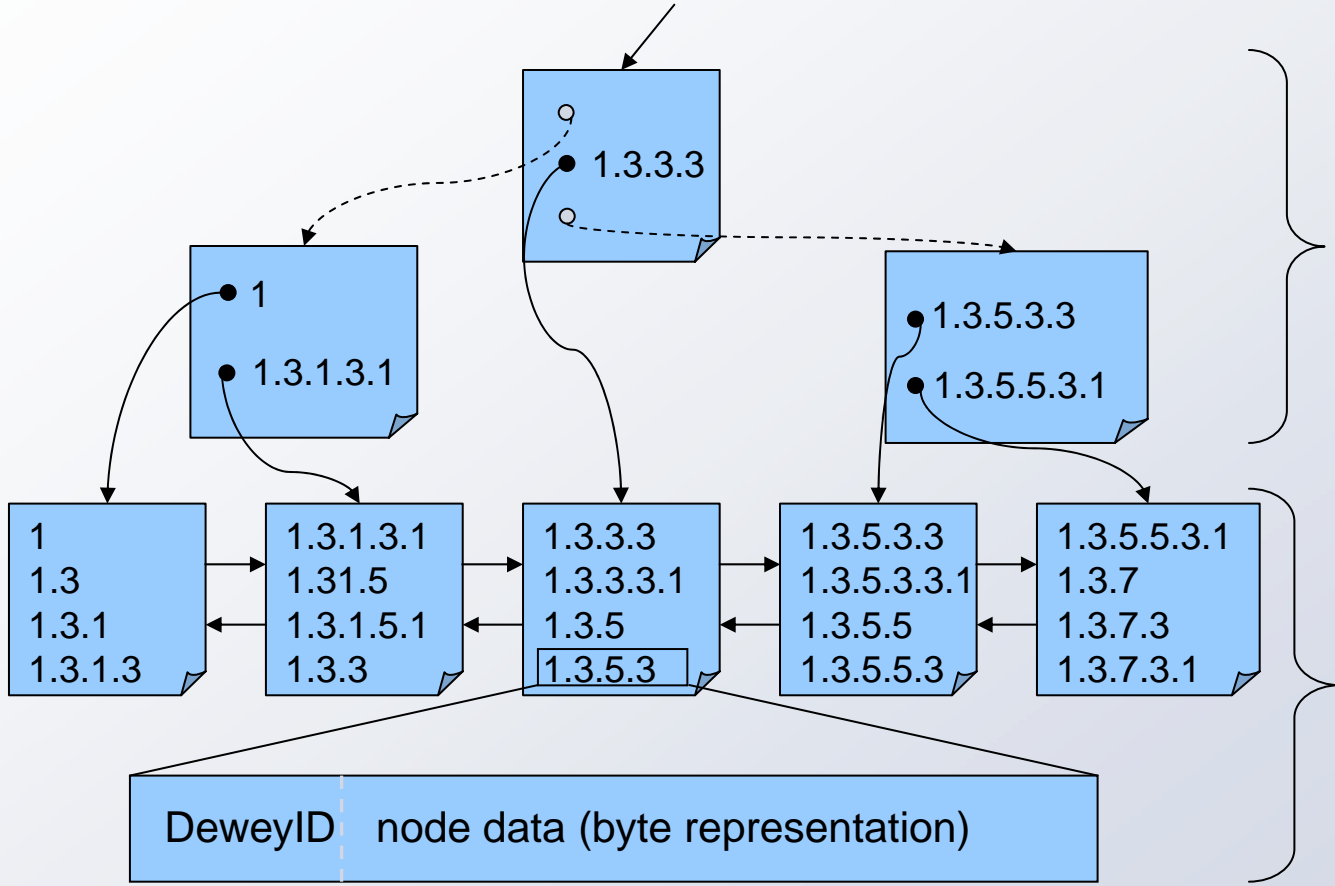
Path Processing
Algorithms

Architecture of
DAIS



XTC Document Index

- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



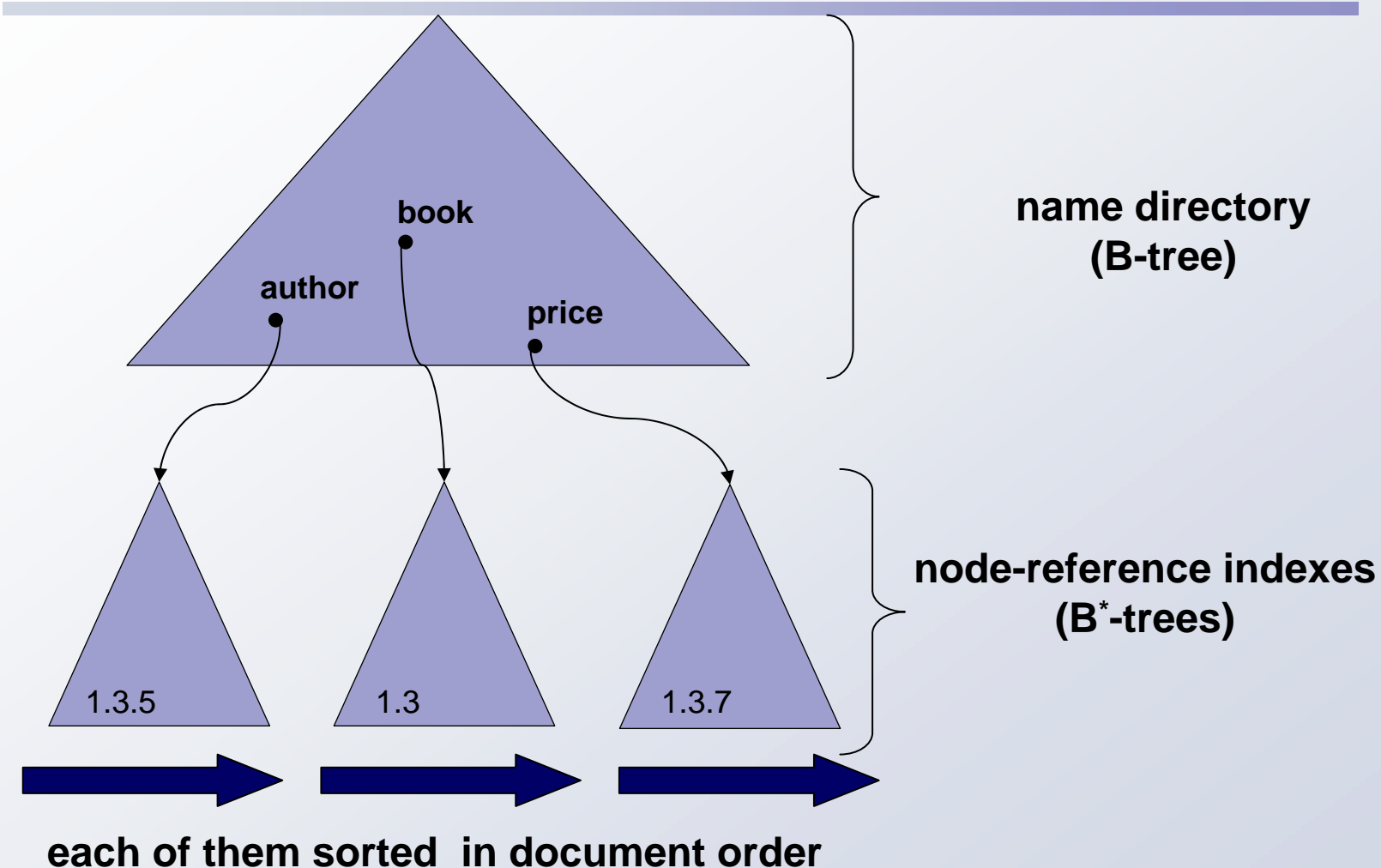
document index
document container

document index is a B-tree for the document(s) stored in the doubly-chained pages of the document container
text values exceeding a given threshold are stored in referenced mode

prefix compression works!



XTC Element Index



In combination with a data guide, the XTC element index can be used like a path index

- unique occurrence of element names: **DeweyIDs are path indexes**
- **homonyms** of element names: additional tests allow the **separation of the paths** (DeweyIDs)

- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



XQuery – Path Processing Steps

- Problem: XQuery is really complex
 - order-preserving joins, implicit grouping, result construction ...
- Example fragment

```
doc("sample.xml")//author  
[count  
  (parent::buch/preceding-sibling::  
  element())>3]/vname[../nname = "Adams"]
```

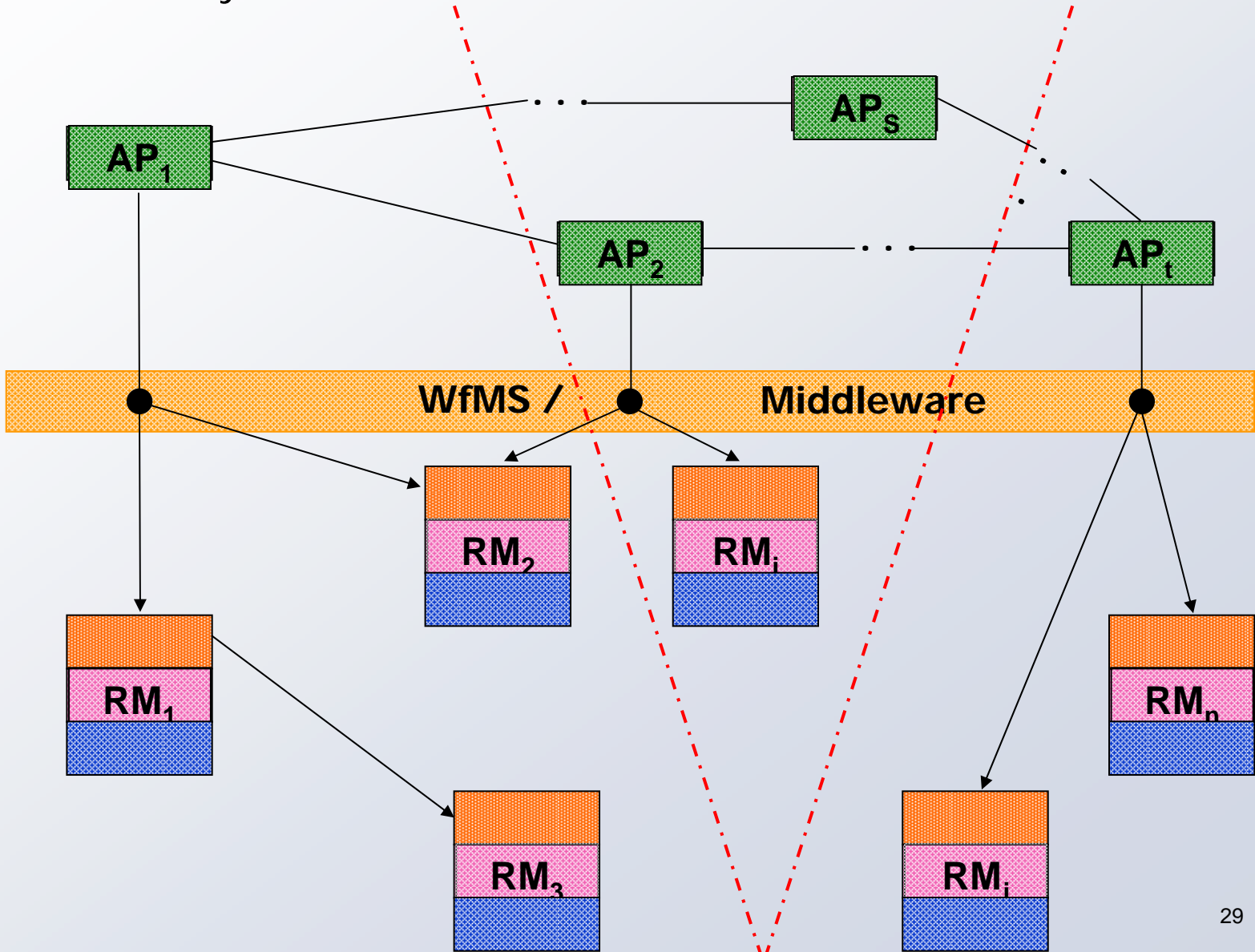
concentration on path processing steps

```
Axis::name_test
```

evaluation order: bottom-up, top-down, starting in the middle
algorithms: selective access via DeweyIDs, locking-aware

Adaptive Information Systems (2)

evolutionary on all levels: a continuous construction site



- Layer Model for DBMSs
- DAIS – the layer stack
- Labeling of XML Tree Nodes
- XML Indexing
- Path Processing Algorithms
- Architecture of DAIS



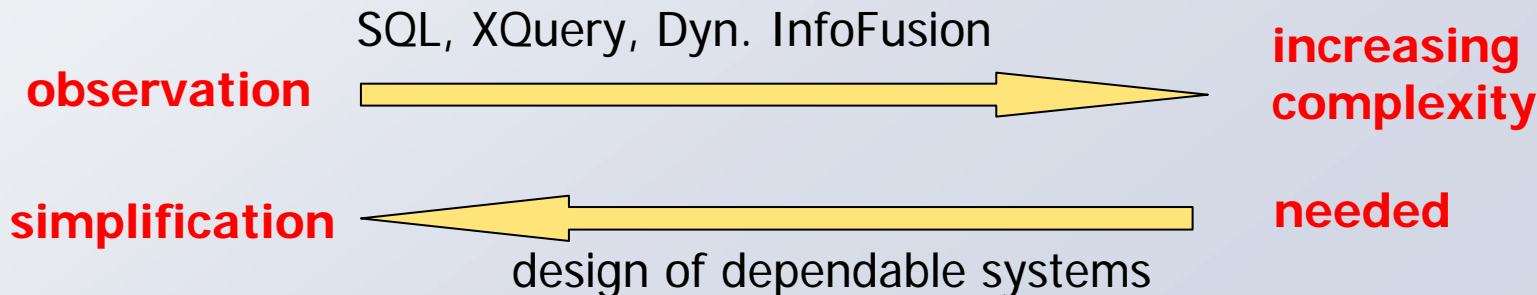
Dependable Adaptive Information Systems

- **Development goals (J. Gray: 1998 Turing Lecture)**
Build a system used by millions of people that is always available – out less than 1 second per 100 years = eight 9s of availability!

Assure that the system only services authorized users, services cannot be denied by unauthorized users, and information cannot be stolen (and prove it)!

“trouble-free + secure + always-up”

- However:



Layer Model
for DBMSs

DAIS –
the layer stack

Labeling of
XML Tree Nodes

XML
Indexing

Path Processing
Algorithms

Architecture of
DAIS

