Synchronisationsverfahren für XML-Datenbanksysteme

Felix Kling

TU Kaiserslautern

13. Februar 2008

Warum spezielle Synchronisationsverfahren?

Bisherige Situation

- Speicherung in RDBMS

Warum spezielle Synchronisationsverfahren?

Bisherige Situation

- Speicherung in RDBMS
  - Komplettes Dokument als ein String

Probleme

- Sperren eines Tupels sperrt komplettes Dokument

Warum spezielle Synchronisationsverfahren?

Bisherige Situation

- Speicherung in RDBMS
  - Komplettes Dokument als ein String
  - Mapping des Dokuments in ein generisches Tabellenschema

Probleme

- Sperren eines Tupels sperrt komplettes Dokument
- Wiederherstellung des Dokuments aufwendig

Warum spezielle Synchronisationsverfahren?

Bisherige Situation

- Speicherung in RDBMS
  - Komplettes Dokument als ein String
  - Mapping des Dokuments in ein generisches Tabellenschema
  - Mapping des Dokuments in ein eigenes Tabellenschema

Probleme

- Sperren eines Tupels sperrt komplettes Dokument
- Wiederherstellung des Dokuments aufwendig
- Schemadefinition notwendig

Warum spezielle Synchronisationsverfahren?

Bisherige Situation

- Speicherung in RDBMS
  - Komplettes Dokument als ein String
  - Mapping des Dokuments in ein generisches Tabellenschema
  - Mapping des Dokuments in ein eigenes Tabellenschema

Probleme

- Sperren eines Tupels sperrt komplettes Dokument
- Wiederherstellung des Dokuments aufwendig
- Schemadefinition notwendig

$\Rightarrow$ Speicherung in nativen XML-Datenbanksystemen

Wichtige Eigenschaften von Synchronisationsverfahren

- Verwendetes Daten- bzw. Sperrmodell
- Mögliche Zugriffsmethoden
- Möglicher Overhead
- Potentielle Nebenläufigkeit
- Erreichte Konsistenzebene

*2PL

Überblick

- Datenmodell: DOM
- Zugriffsmethoden: DOM
- Zugriff: über Dokumentwurzel
- Varianten
  - DOC2PL
  - Node2PL
  - NO2PL
  - OO2PL

Sperrmodi

Sperrmodi

Strukturspersperren

Inhaltssperren

Sperrmodi

Strukturspersperren

T: Navigation

M: Modifikation

IDS: Sprung per ID/IDREF - Beziehung

IDX: Löschen von Knoten mit *id*-Attribut

Inhaltssperren

Sperrmodi

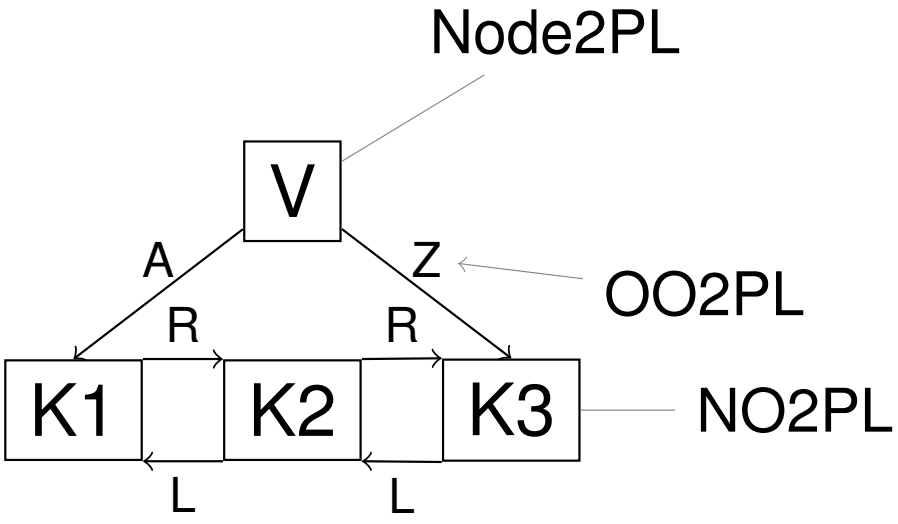
Strukturspersperren

- T: Navigation
- M: Modifikation
- IDS: Sprung per ID/IDREF - Beziehung
- IDX: Löschen von Knoten mit *id*-Attribut

Inhaltssperren

- S: Lesen
- X: Schreiben

Sperrobjekte



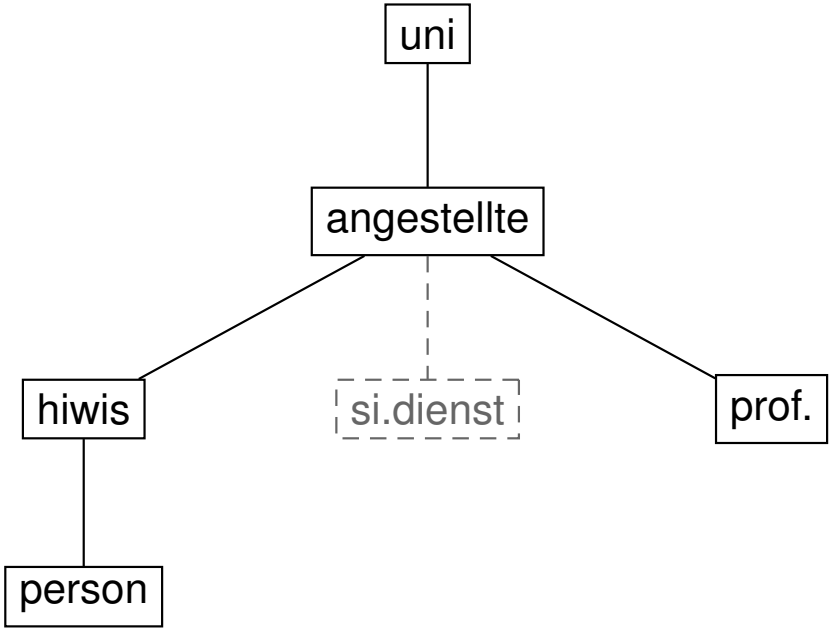
DOC2PL: Dokumentwurzel

Node2PL: Elternknoten

NO2PL: Knoten, deren Kanten traversiert werden

OO2PL: Kanten

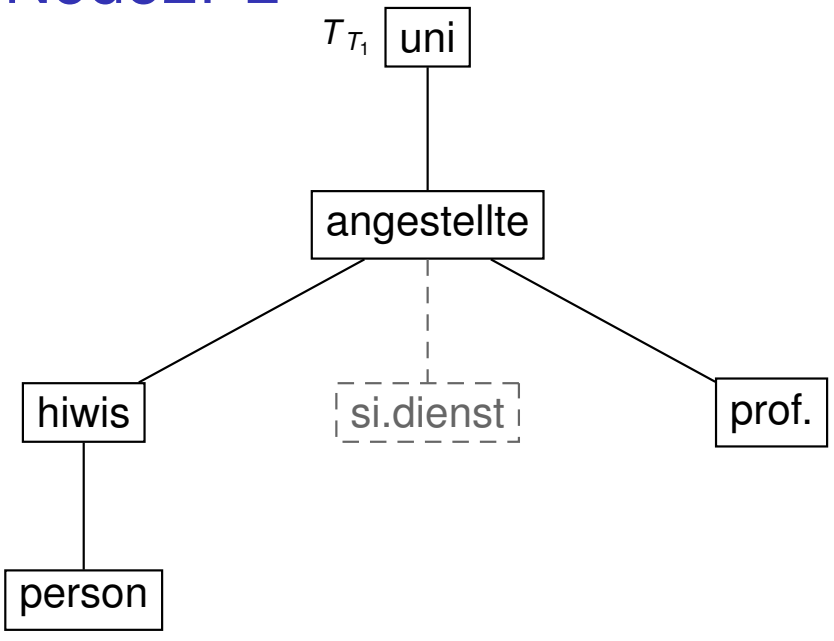
Beispiel



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

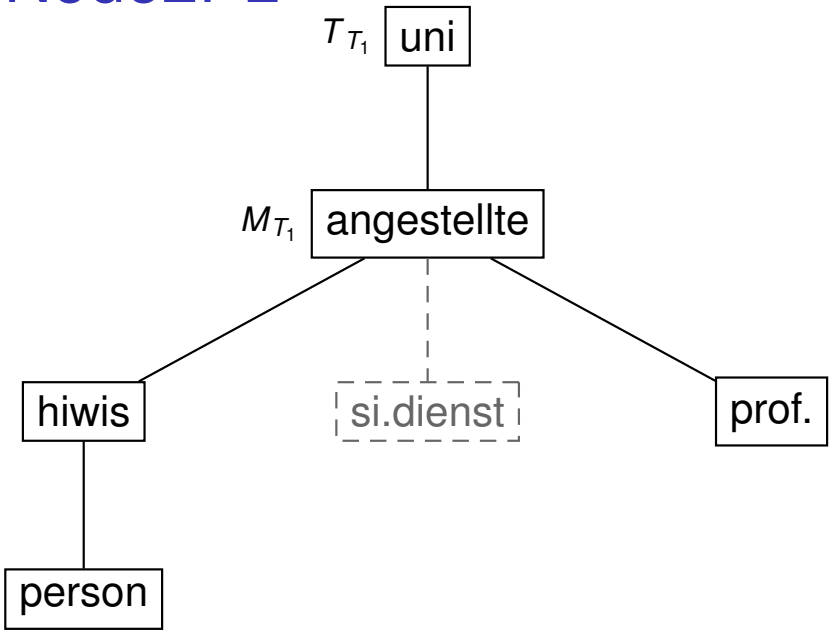
Node2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

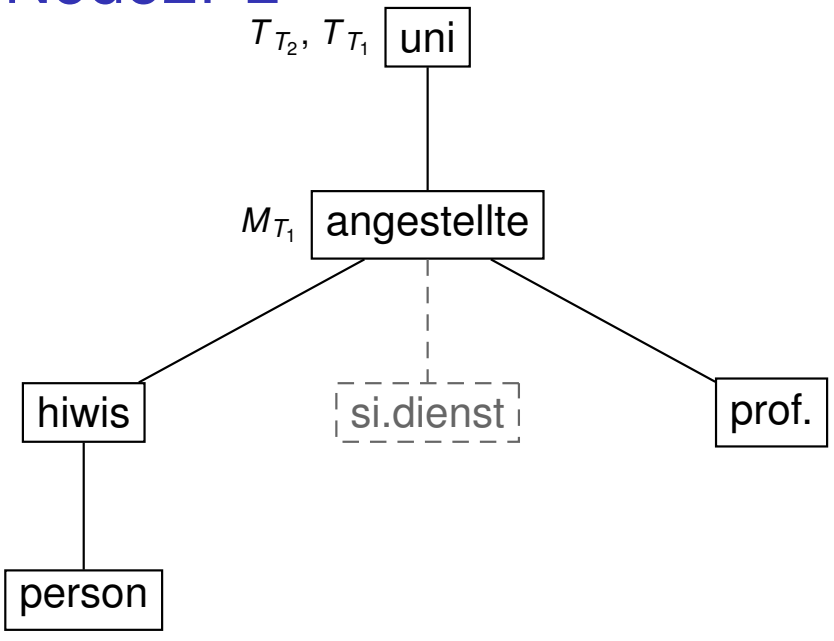
Node2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

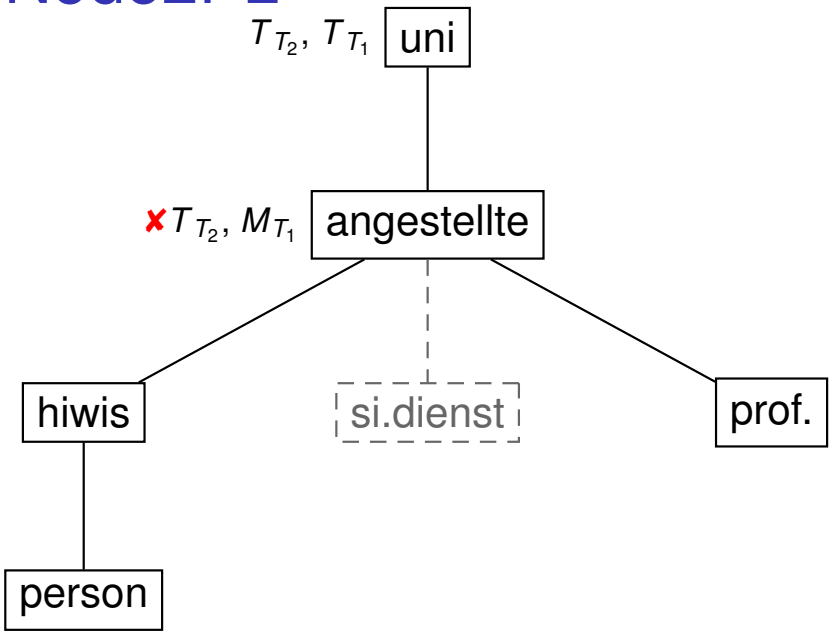
Node2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

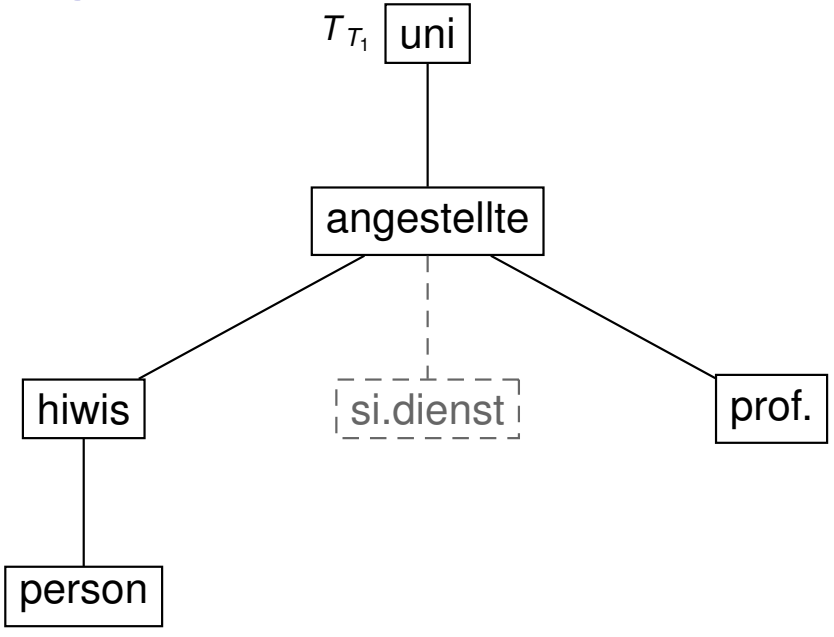
Node2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

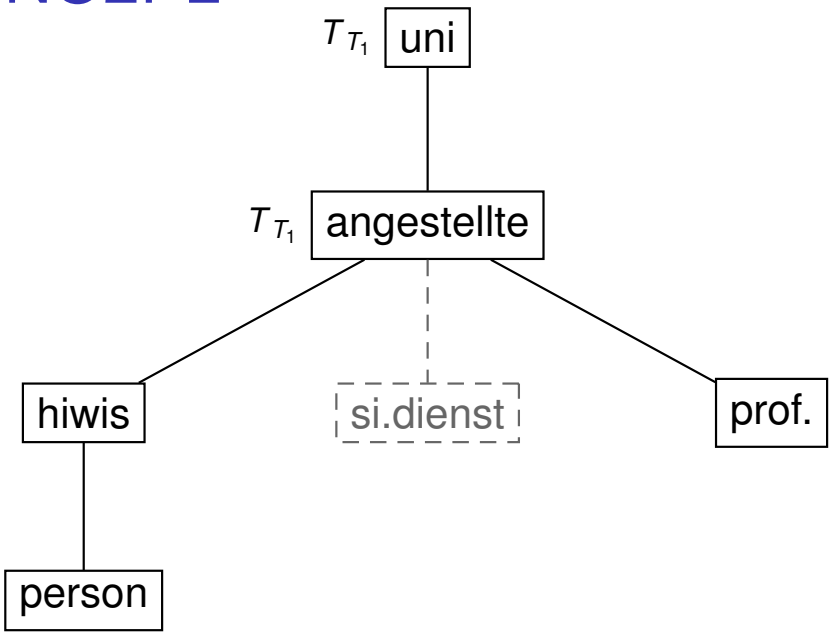
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

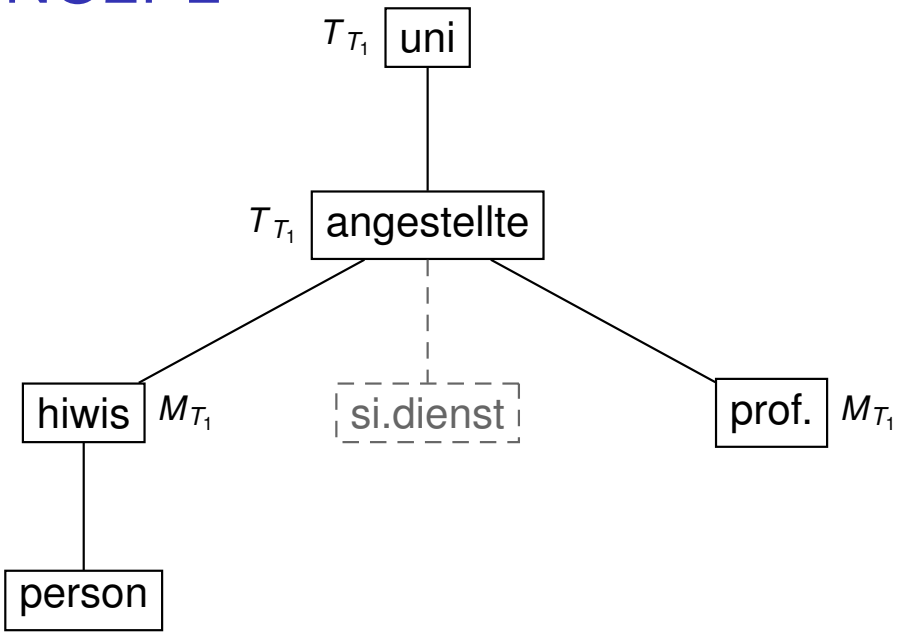
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

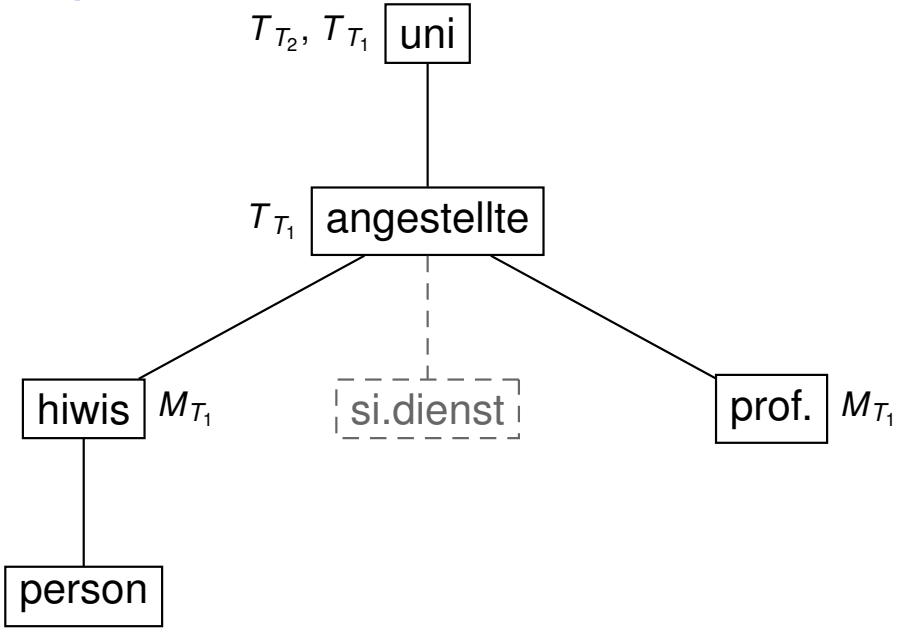
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

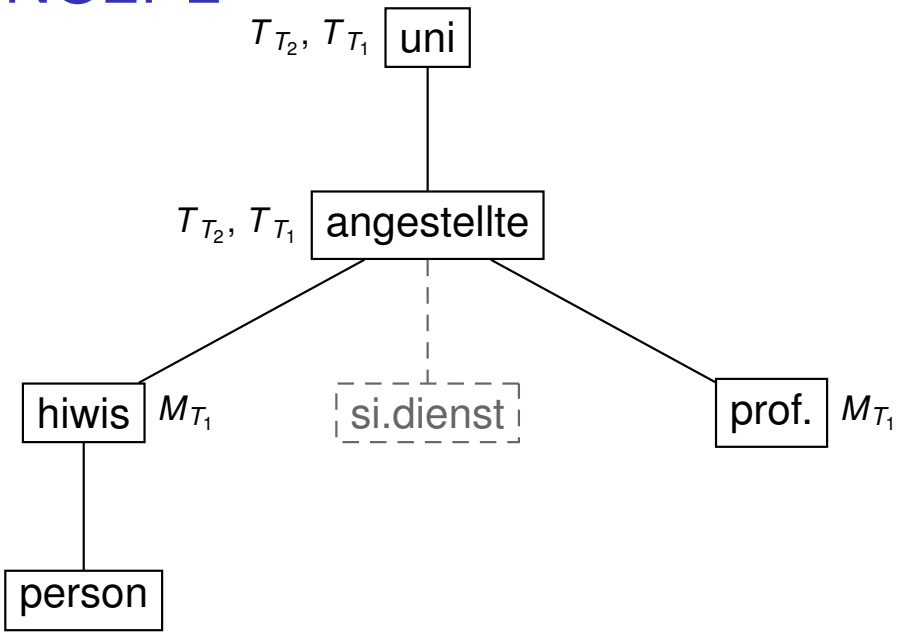
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

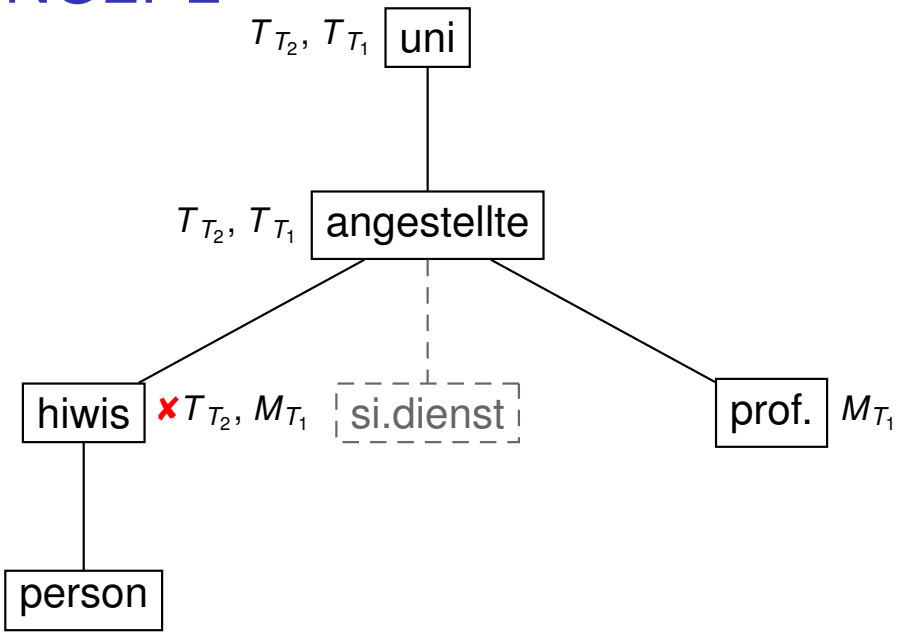
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

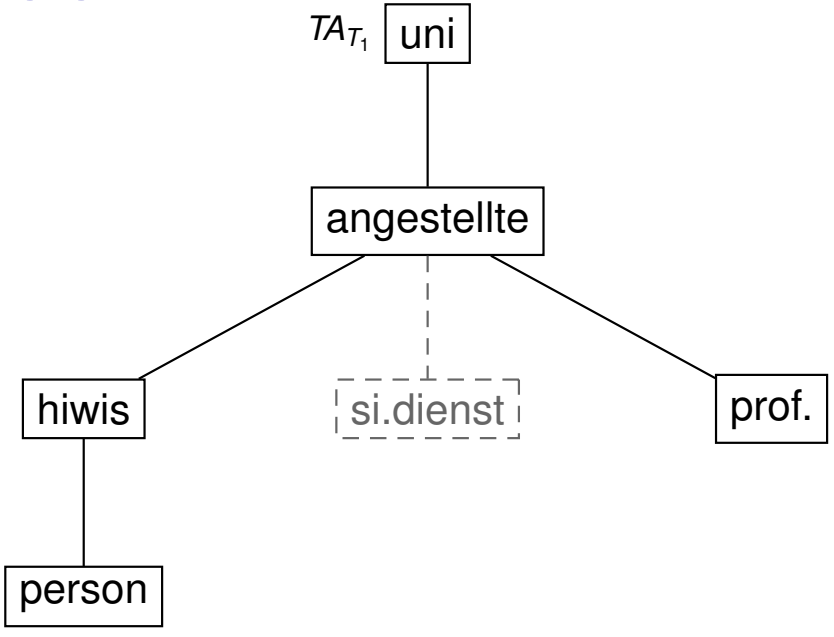
NO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

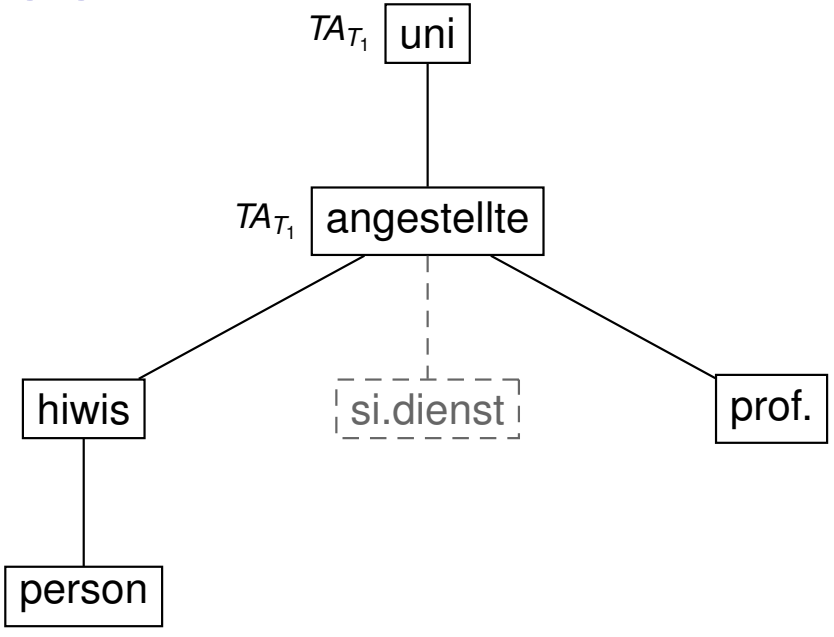
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

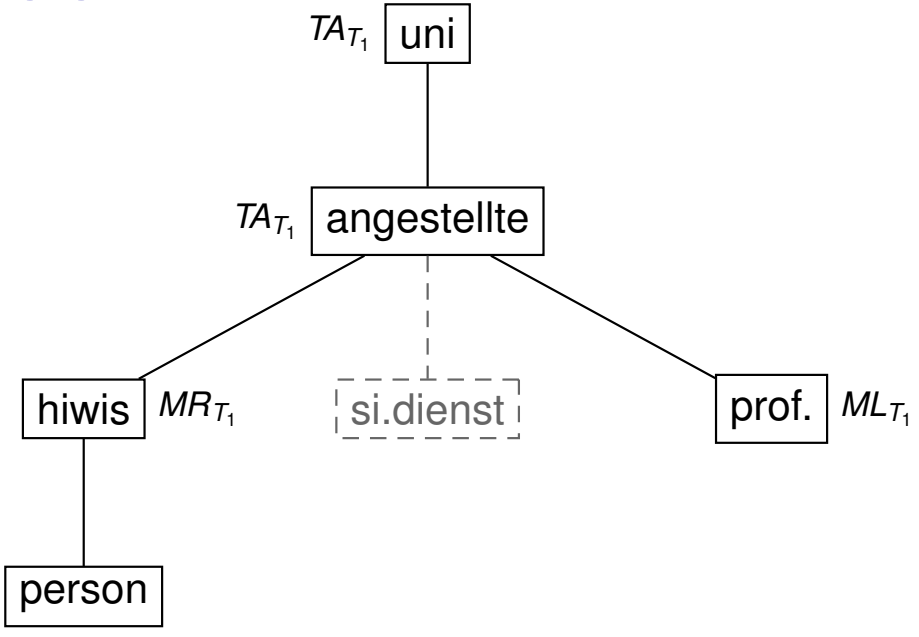
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

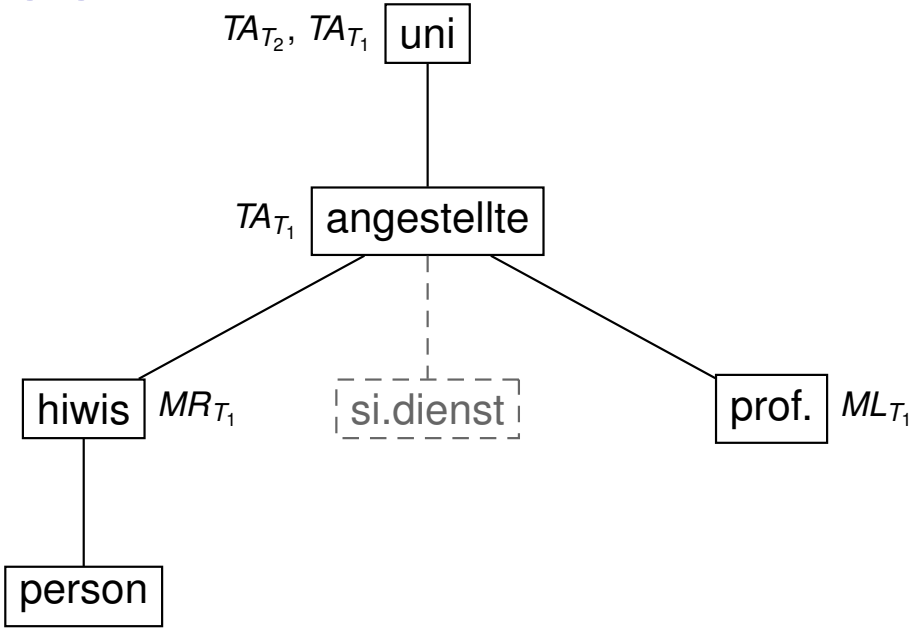
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

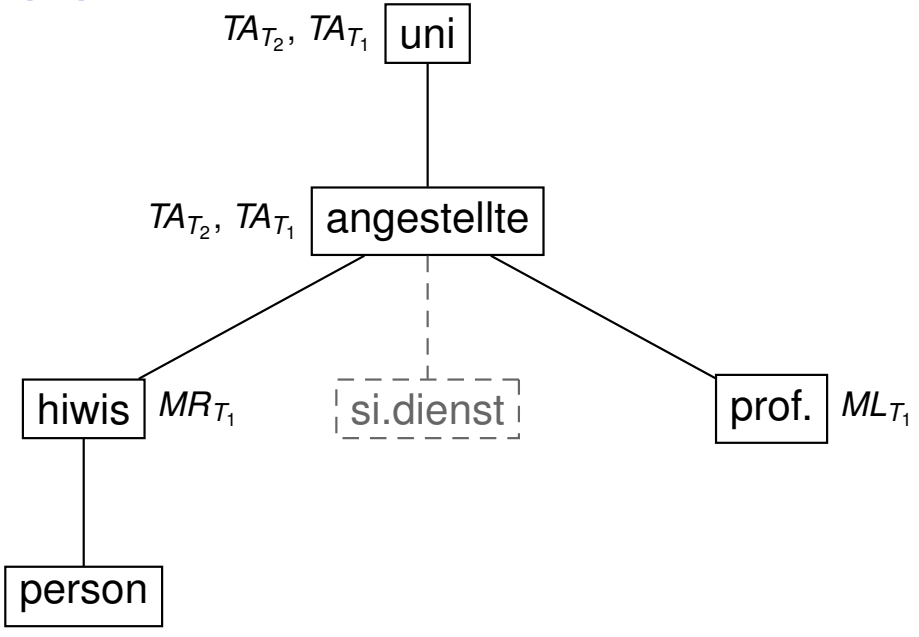
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

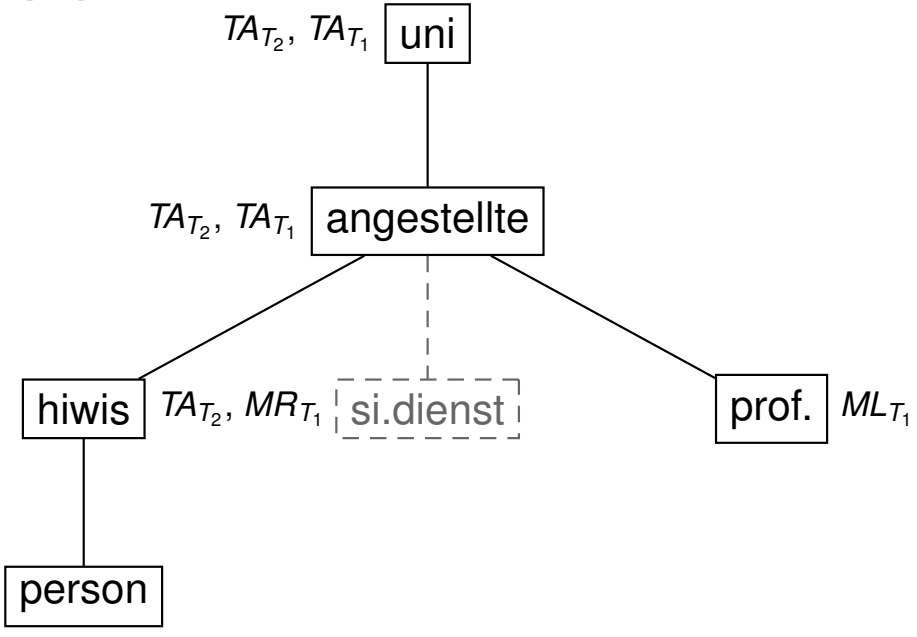
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

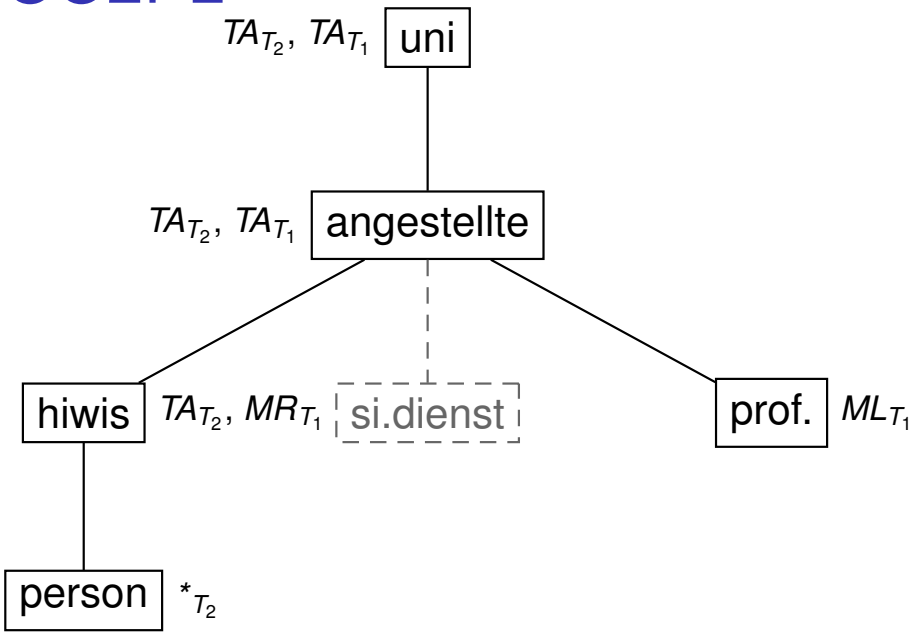
OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

Beispiel

OO2PL



- T_1 fügt einen neuen Knoten hinter *hiwis* ein (*sicherheitsdienst*).
- T_2 navigiert zu *person*.

XDGL

Überblick

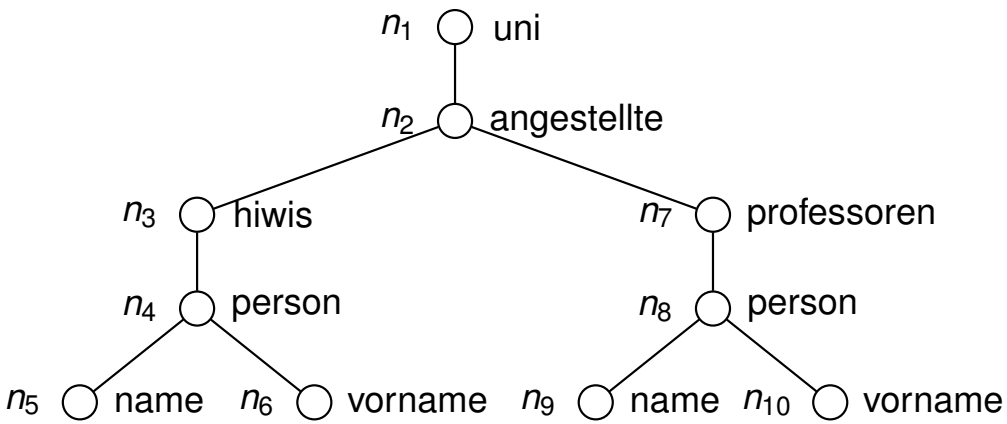
- Datenmodell: DataGuide
- Zugriffsmethoden: XPath

DataGuide

- Strukturzusammenfassung eines Dokumentes
- Jeder Pfad im Dokument ist genau einmal im DataGuide enthalten.
- Jeder Pfad im DataGuide existiert im Dokument.

DataGuide

```
<uni>
  <angestellte>
    <hiwis>
      <person>
        <name>Kling</name>
        <vorname>Felix</vorname>
      </person>
      <person>
        <name>Hiwi</name>
         <vorname>Muster</vorname>
      </person>
    </hiwis>
    <professoren>
     <person>
       <name>Professor</name>
       <vorname>Muster</vorname>
     </person>
    </professoren>
  </angestellte>
</uni>
```



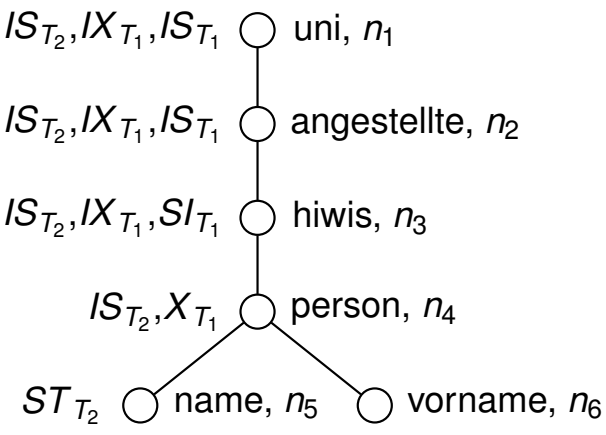
Zugriff

- vereinfachte Form von XPath
- keine Prädikattests
- *parent*-, *child*-, *descendant*- und *attribute*-Achse

Sperrmodi

- *SI*,*SA*,*SB* (*shared insert*, *shared after*, *shared befor*)
- *X* (*exclusive lock*)
- *ST* (*shared tree*)
- *XT* (*exclusive tree*)
- *IS*, *IX*

Beispiel



- T_1 fügt einen leeren *person*-Knoten unter *hiwis* (n_3) ein.
- T_2 liest n_5.

Phantomvermeidung

Logische Sperren

Vermeidung von Phantomen auf der *descendant*-Achse

Phantomvermeidung

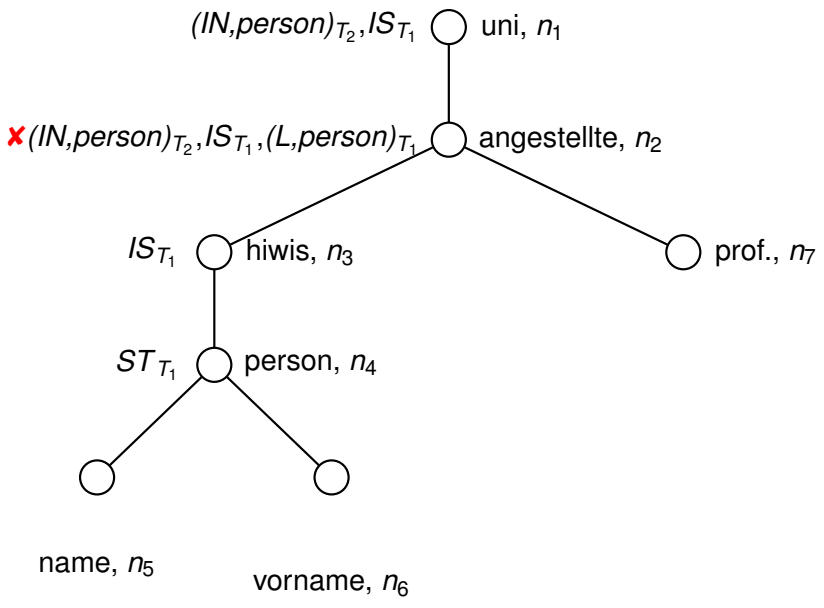
Logische Sperren

Vermeidung von Phantomen auf der *descendant*-Achse

Syntax

- Anfrage- und Löschoperationen: *(L, Knotenname)*
- Einfügeoperationen: *(IN, Knotenname)*

Beispiel



- T_1 liest alle *person*-Knoten (*/uni/angestellte//person*).
- T_2 fügt einen *person*-Knoten unter n_7 ein.

OptiX

Überblick

- Datenmodell: XPath
- Zugriffsmethoden: XPath
- Zugriff: über Dokumentwurzel

Versionsverwaltung

- Jede Transaktion T_i bekommt einen eindeutigen Zeitstempel $ID(T_i)$.
- Zwei Zeitstempel pro Knoten p: $V(p)$ und $IV(p)$
- $ID(T_j) \in EB(T_i)$
- Zugriff auf Knoten p, wenn $V(p) \in EB(T_i) \wedge IV(p) \notin EB(T_i)$.

Read- und Write-Sets

Read-Sets

RR(T_i): Enthält die Wurzelknoten der zurückgelieferten Teilbäume.

ER(T_i): Enthält alle Knoten, die explizit gelesen werden.

$ER_A(T_i)$: Enthält alle Knoten, hinter denen ein Knoten eingefügt wird.

Read- und Write-Sets

Write-Sets

$D(T_i)$: Enthält alle Knoten, die von T_i gelöscht oder ersetzt werden.

$R_n(T_i)$: Enthält alle Knoten, die umbenannt wurden.

$I(T_i)$: Enthält die Elternknoten der Knoten, die von T_i eingefügt worden sind.

$I_A(T_i)$: Enthält die gleichen Knoten wie $ER_A(T_i)$.

Beispiel

```
FOR $a IN /uni/angestellte
RETURN $a
```

- $RR(T_i)$ enthält Wurzelknoten des Ergebnisteilbaums.
- $ER(T_i)$ enthält $a und alle seine Vorgänger, da sie den Pfadausdruck */uni/angstellte* erfüllen

Beispiel

```
FOR $a IN /uni,
    $b IN $a/sonstige,
    $c IN $a/studenten
UPDATE $a {
DELETE $b,
INSERT <alumni /> AFTER $c }
```

- $ER(T_i)$ enthält $a, $b, $c und alle ihre Vorgänger
- $D(T_i)$ enthält $b
- $I(T_i)$ enthält $a
- $I_A(T_i)$ enthält $c

Validierung

T_i		T_j				
			n ∈			q ∈
		D	R_n	I	I_A	WS
n ∈	RR	-	-	-	√	-
	ER	-	-	√	√	√
	ER_A	-	-	√	-	√
q ∈	RS	-	√	√	√	

Tabelle: Konfliktmatrix für OptiX

3 Phasen

- *working phase*
- *validation phase*
- *update phase*

taDOM

Überblick

- Datenmodell: taDOM (Erweiterung von DOM)
- Zugriffsmethoden: DOM API, XPath
- Varianten:
  - taDOM2
  - taDOM2+
  - taDOM3
  - taDOM3+
- Besonderheit: Nutzung des Nummeriungsschemas (DeweyIDs) zur Anfrageauswertung

Varianten

taDOM2:	Unterstützt DOM-Level-2-Operatoren
taDOM2+:	Erweiterung von taDOM2, um Zugriff auf Dokument zu vermeiden
taDOM3:	Unterstützt DOM-Level-3-Operatoren
taDOM3+:	Erweiterung von taDOM3, um Zugriff auf Dokument zu vermeiden

DOM-Erweiterung

- Attributwurzel
- String-Knoten
- Virtuelle Navigationskanten

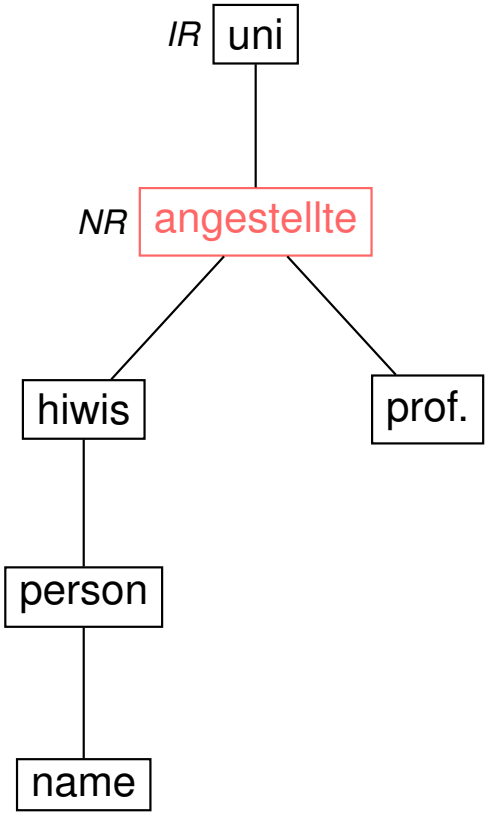
Sperrmodi von taDOM2

- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2

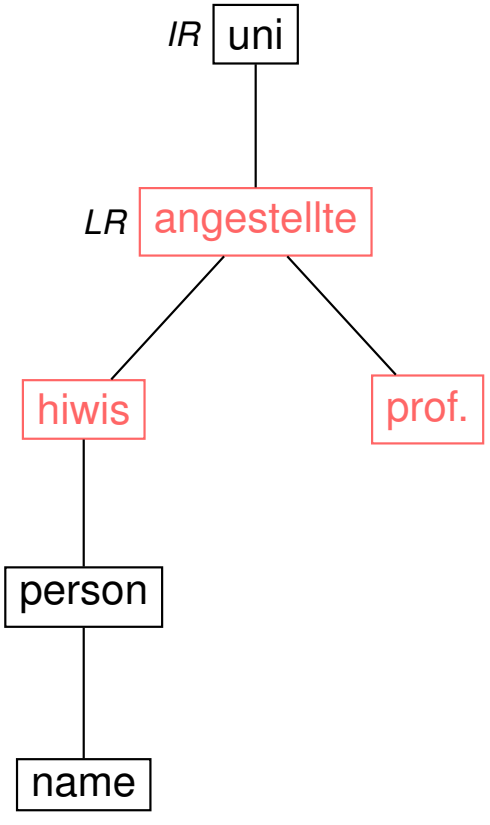
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



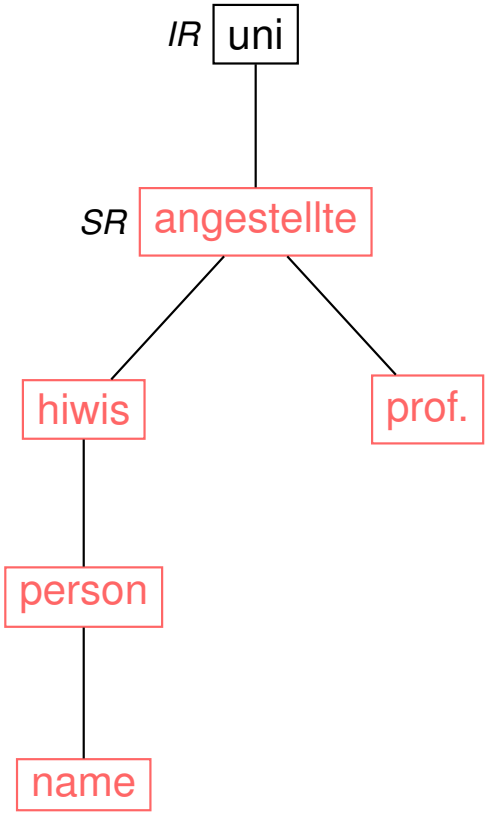
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



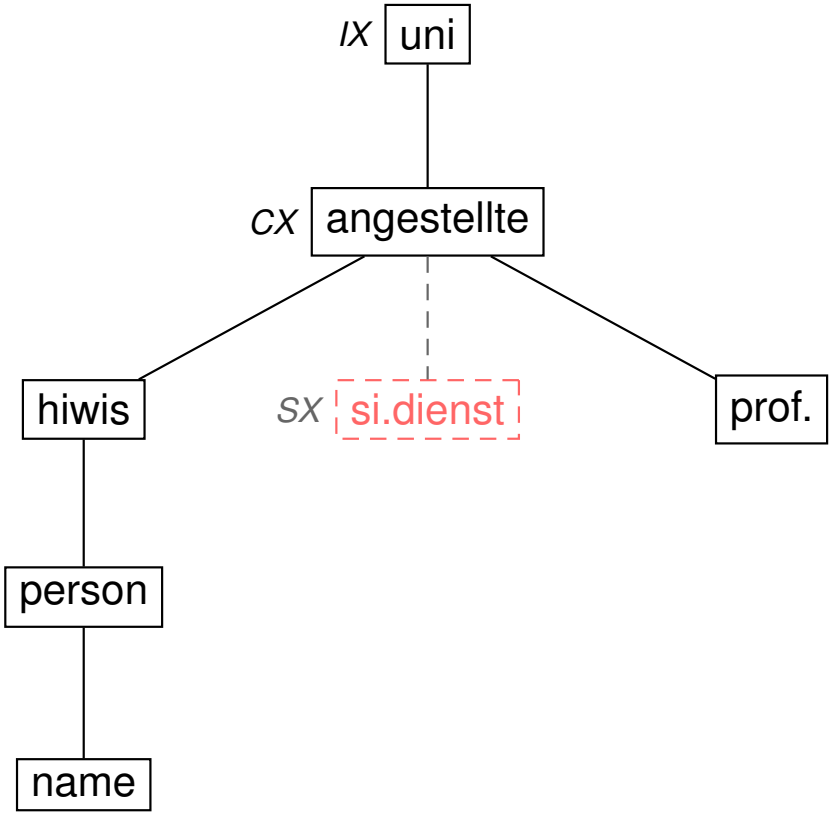
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



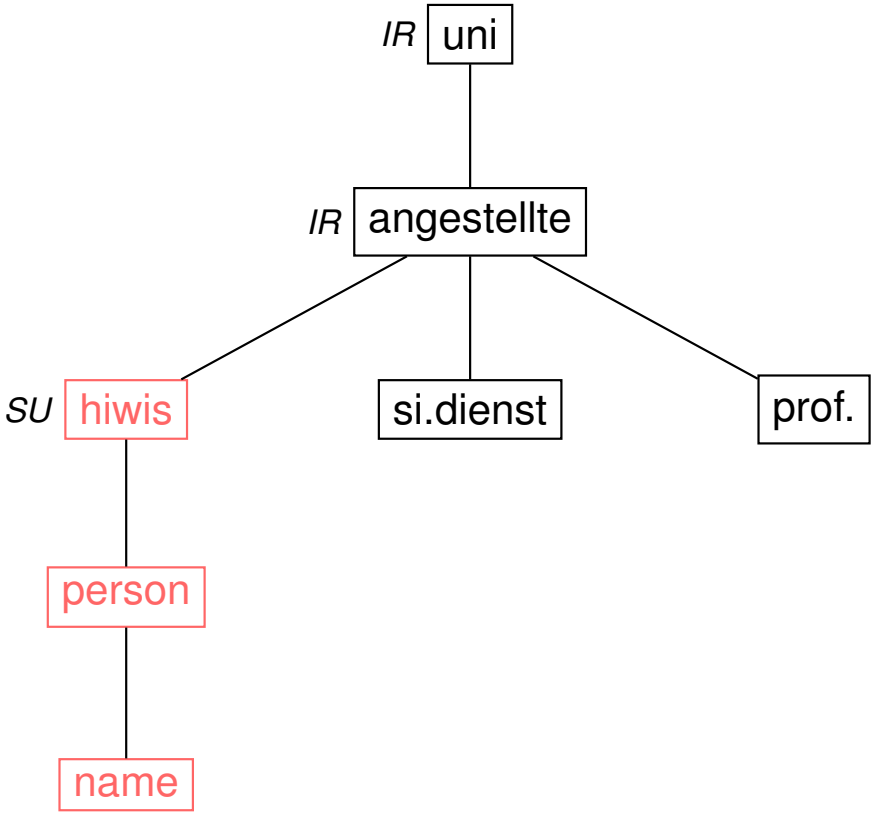
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



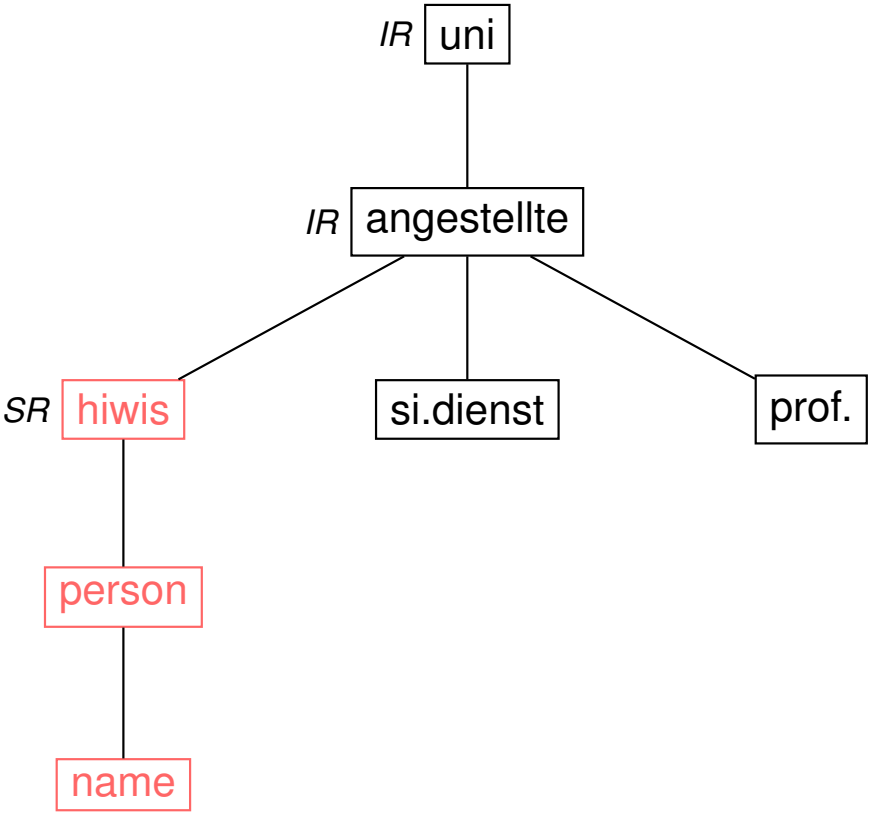
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



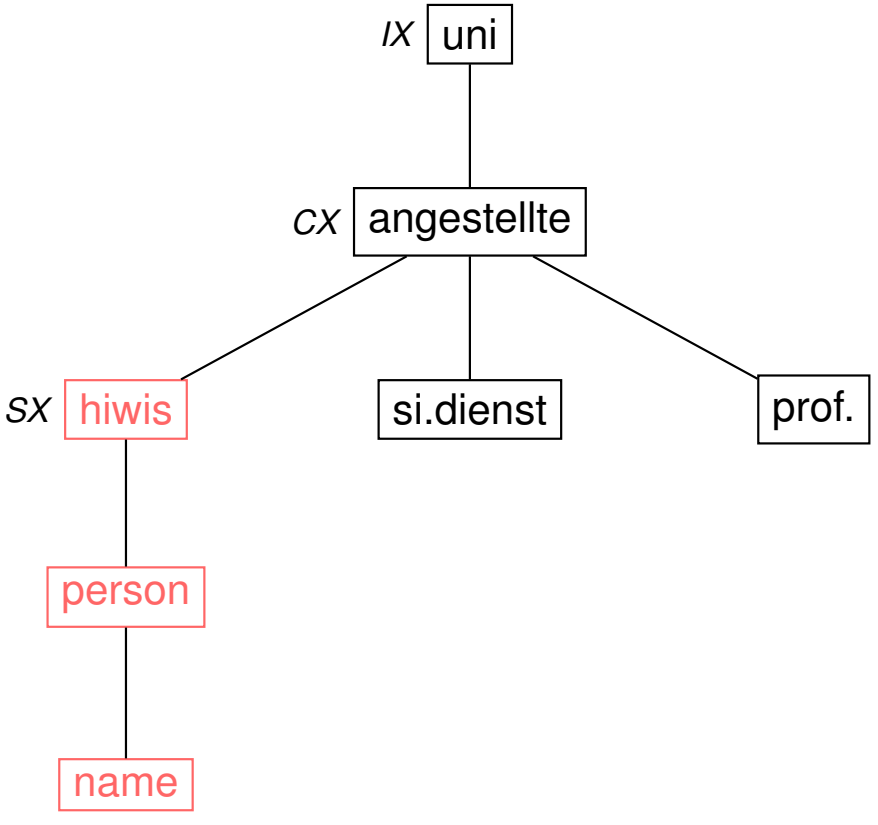
- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrmodi von taDOM2



- IR (*intention read*)
- NR (*node read*)
- LR (*level read*)
- SR (*subtree read*)
- IX (*intention exclusive*)
- CX (*child exklusive*)
- SX (*subtree exclusive*)
- SU (*subtree update*)

Sperrkonversion

- Eine Sperre pro Transaktion und Knoten

Sperrkonversion

- Eine Sperre pro Transaktion und Knoten
- Bei Anforderung einer neuen Sperre, Überführung in andere Sperre mittels Konversionstabelle

Sperrkonversion

- Eine Sperre pro Transaktion und Knoten
- Bei Anforderung einer neuen Sperre, Überführung in andere Sperre mittels Konversionstabelle
- Ziel: Beibehaltung des selben Isolationslevels der ursprünglichen Sperre

Sperrkonversion

- Eine Sperre pro Transaktion und Knoten
- Bei Anforderung einer neuen Sperre, Überführung in andere Sperre mittels Konversionstabelle
- Ziel: Beibehaltung des selben Isolationslevels der ursprünglichen Sperre
- Problem: Zugriff auf Dokument erforderlich

Sperrkonversion

- Eine Sperre pro Transaktion und Knoten
- Bei Anforderung einer neuen Sperre, Überführung in andere Sperre mittels Konversionstabelle
- Ziel: Beibehaltung des selben Isolationslevels der ursprünglichen Sperre
- Problem: Zugriff auf Dokument erforderlich
- Lösung: Einführung weiterer Sperrmodi (taDOM2+ und taDOM3+)

Phantomvermeidung

Wertbasierte Achsensperren

- Zugriff über Indexe
- Einfügen neuer Knoten

Syntax

$l_{axis}(deweyID, axis, value, mode)$

Validierung

- Lagebestimmung der Kontextknoten
- Achsenüberlagerung
- Sperrmodi

Weitere Eigenschaften

- Kantensperren
- Sperrtiefe
- taDOM2: Virtuelle Namensknoten

taDOM3+

	IR	NR	LR	SR	IX	NRIX	LRIX	SRIX	CX	NRCX	LRCX	SRCX	NU	LRNU	SRNU	NX	LRNX	SRNX	SU	SX
IR	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-
NR	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-
LR	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-
SR	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IX	+	+	+	-	+	+	+	-	+	+	+	-	+	+	-	+	+	-	-	-
NRIX	+	+	+	-	+	+	+	-	+	+	+	-	-	-	-	-	-	-	-	-
LRIX	+	+	+	-	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-
SRIX	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CX	+	+	-	-	+	+	-	-	+	+	-	-	+	-	-	+	-	-	-	-
NRCX	+	+	-	-	+	+	-	-	+	+	-	-	-	-	-	-	-	-	-	-
LRCX	+	+	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRCX	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NU	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-
LRNU	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-
SRNU	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NX	+	-	-	-	+	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-
LRNX	+	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SRNX	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SU	+	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bemerkungen zu den Verfahren

Allgmein

- Meißt nur eine Verarbeitungsmethode unterstützt
- Eingeschränkte Verwendung von XPath
- Fehlende Indexunterstützung
- Nicht immer ordnungserhaltend

Bemerkungen zu den Verfahren

*2PL

Vorteile

- Kompakte Sperrverfahren
- ID/IDREF-Unterstützung

Nachteile

- Node2PL und NO2PL zu restriktiv
- Zugriff über die Wurzel
- Keine Indexe
- Kein XPath

Bemerkungen zu den Verfahren

XDGL

Vorteile

- Vergleichsweise wenig Sperren

Nachteile

- DataGuide zu unflexibel (zu restriktiv)
- Mehrere DataGuides zu einem Dokument möglich
- Eingeschränktes XPath

Bemerkungen zu den Verfahren

OptiX

Vorteile

- Optimistisches Verfahren

Nachteile

- Effiziente Implementierung der Versionsverwaltung notwendig
- Optimistisches Verfahren
- Ancestor/Descendant-Beziehung

Bemerkungen zu den Verfahren

taDOM

Vorteile

- DeweyIDs
- Unterstützung von DOM, XPath und SAX
- Unterstützung von Indexen
- Vollständiges XPath
- Sperrtiefe

Nachteile

- Relativ viele Sperren
- Performance bei großen Dokumenten?
- Sperrtiefe

Ausblick

- Anforderungen abhängig von Anwendungsszenario
- Dokument- oder datenorientierte Nutzung von XML-Dokumenten
- Neue Technologien / Formate
- Neue Verarbeitungsmethoden für XML-Dokumente

Vielen Dank für Ihre Aufmerksamkeit.

Zusätzliches Material

- *2PL
- XDGL
- taDOM

Sperrmatrix

	T	M
T	+	-
M	-	-

Tabelle: Node2PL und NO2PL

	TL	TR	TA	TZ	ML	MR	MA	MZ
TL	+	+	+	+	-	+	+	+
TR	+	+	+	+	+	-	+	+
TA	+	+	+	+	+	+	-	+
TZ	+	+	+	+	+	+	+	-
ML	-	+	+	+	-	+	+	+
MR	+	-	+	+	+	-	+	+
MA	+	+	-	+	+	+	-	+
MZ	+	+	+	-	+	+	+	-

Tabelle: OO2PL

Sperrmatrix

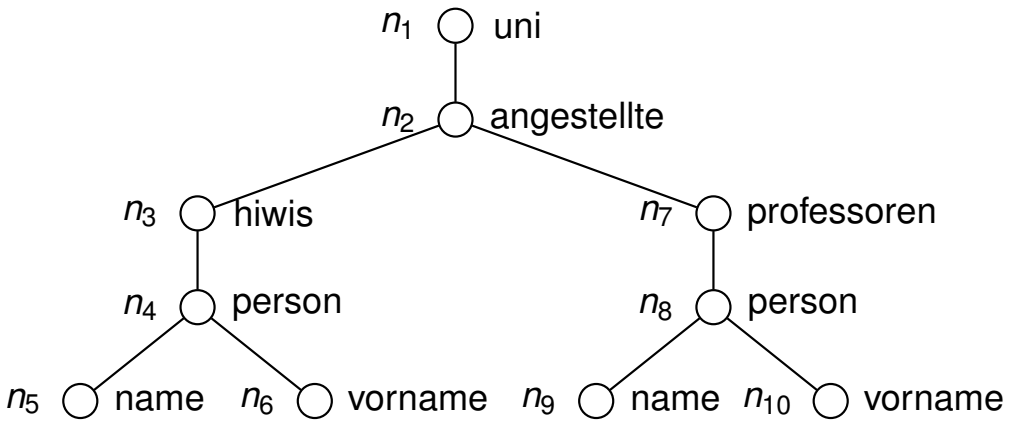
◂ Zurück zu Node2PL ◂ Zurück zu NO2PL ◂ Zurück zu OO2PL

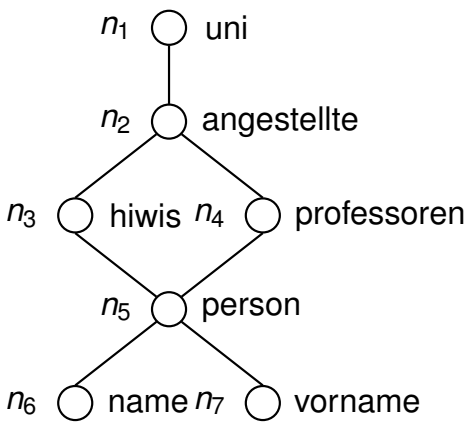
	SI	SA	SB	X	ST	XT	IS	IX
SI	-	+	+	-	+	-	+	+
SA	+	-	+	-	+	-	+	+
SB	+	+	-	-	+	-	-	+
X	-	-	-	-	-	-	+	+
ST	+	+	+	-	+	-	+	+
XT	-	-	-	-	-	-	-	-
IS	+	+	+	+	+	-	+	+
IX	+	+	+	+	-	-	+	+

Tabelle: Sperrmatrix von XDGL

◂ Zurück zu XDGL

DataGuide


n_1 uni
n_2 angestellte
n_3 hiwis
n_7 professoren
n_4 person
n_8 person
n_5 name
n_6 vorname
n_9 name
n_{10} vorname


n_1 uni
n_2 angestellte
n_3 hiwis
n_4 professoren
n_5 person
n_6 name
n_7 vorname

DataGuide

◂ Zurück

Konversionsmatrix zu taDOM2/taDOM2+

	IR	NR	LR	SR	IX	CX	SX	SU
IR	IR	NR	LR	SR	IX	CX	SX	SU
NR	NR	NR	LR	SR	IX	CX	SX	SU
LR	LR	LR	LR	SR	IX_{NR}	CX_{NR}	SX	SU
SR	SR	SR	SR	SR	IX_{SR}	CX_{SR}	SX	SR
IX	IX	IX	IX_{NR}	IX_{SR}	IX	CX	SX	SX
CX	CX	CX	CX_{NR}	CX_{SR}	CX	CX	SX	SX
SX	SX	SX	SX	SX	SX	SX	SX	SX
SU	SU	SU	SU	SU	SX	SX	SX	SU

Tabelle: taDOM2

Konversionsmatrix zu taDOM2/taDOM2+

	IR	NR	LR	SR	IX	LRIX	SRIX	CX	LRCX	SRCX	SX	SU
IR	IR	NR	LR	SR	IX	LRIX	SRIX	CX	LRCX	SRCX	SX	SU
NR	NR	NR	LR	SR	IX	LRIX	SRIX	CX	LRCX	SRCX	SX	SU
LR	LR	LR	LR	SR	LRIX	LRIX	SRIX	LRCX	LRCX	SRCX	SX	SU
SR	SR	SR	SR	SR	SRIX	SRIX	SRIX	SRCX	SRCX	SRCX	SX	SR
IX	IX	IX	LRIX	SRIX	IX	LRIX	SRIX	CX	LRCX	SRCX	SX	SX
LRIX	LRIX	LRIX	LRIX	SRIX	LRIX	LRIX	SRIX	LRCX	LRCX	SRCX	SX	SX
SRIX	SRIX	SRIX	SRIX	SRIX	SRIX	SRIX	SRIX	SRCX	SRCX	SRCX	SX	SX
CX	CX	CX	LRCX	SRCX	CX	LRCX	SRCX	CX	LRCX	SRCX	SX	SX
LRCX	LRCX	LRCX	LRCX	SRCX	LRCX	LRCX	SRCX	LRCX	LRCX	SRCX	SX	SX
SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SRCX	SX	SX
SX	SX	SX	SX	SX	SX	SX	SX	SX	SX	SX	SX	SX
SU	SU	SU	SU	SU	SX	SX	SX	SX	SX	SX	SX	SU

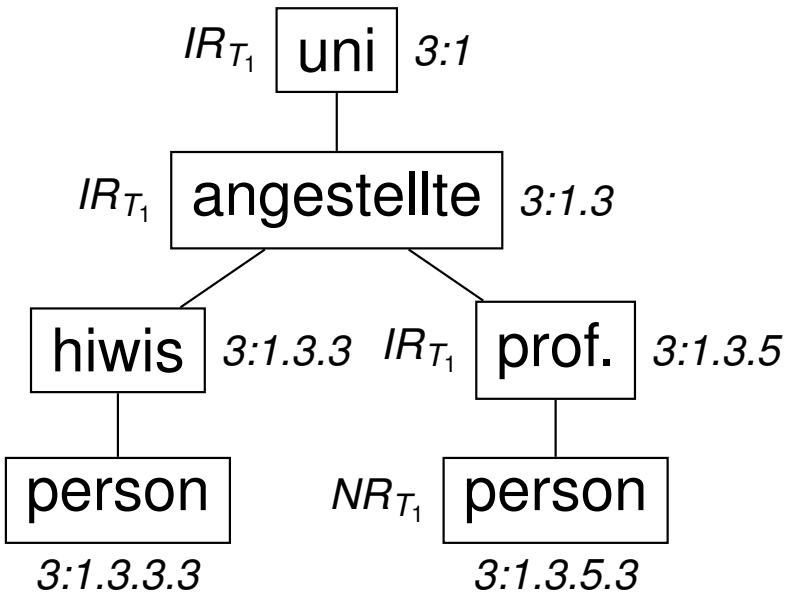
Tabelle: taDOM2+

Konversionsmatrix zu taDOM2/taDOM2+

◂ Zurück zu taDOM

Wertbasierte Achsensperren

Beispiel



- T_1 führt */uni/angestellte/hiwis/following::person* aus (Sperre: $l_{axis}(3{:}1.3.3, Following, \text{"}person\text{"}, S)$).
- T_2 möchte einen *person*-Knoten unter *professoren* einfügen (Sperre: $l_{axis}(3{:}1.3.5, Child, \text{"}person\text{"}, X)$).

Wertbasierte Achsensperren

	Vorhandene Sperre auf einem Knoten der *Preceding-Sibling*-Achse des Kontextknotens						
	Self	Attribute	Child	Descendant	Following-Sibling	Following	IDValue
Ancestor	nein	nein	nein	nein	nein	nein	nein
Parent	nein	nein	nein	nein	nein	nein	nein
Preceding	ja	nein	ja	ja	ja	ja	nein
Preceding-Sibling	ja	nein	nein	nein	ja	ja	nein
Self	nein	nein	nein	nein	ja	ja	nein
Attribute	nein	nein	nein	nein	nein	nein	nein
Child	nein	nein	nein	nein	nein	ja	nein
Descendant	nein	nein	nein	nein	nein	ja	nein
Following-Sibling	nein	nein	nein	nein	ja	ja	nein
Following	nein	nein	nein	nein	ja	ja	nein
IDValue	nein	nein	nein	nein	nein	nein	nein

Tabelle: Achsenüberlagerungstabelle der *preceding-sibling*-Lage

Wertbasierte Achsensperren

Zurück zu taDOM