

Seminar

Langzeitarchivierung

Frühjahrs-Trimester 2009

Gerrit Kahn

15.06.2009



**Der Einsatz des „Universal Virtual Computer“ in der
„Koninklijke Bibliotheek“**

- Einleitung
- Was ist der UVC?
- Einsatz in der KB
- Schlussfolgerungen

- **Einleitung**
 - Motivation
 - Bedürfnisse der KB
 - Zentrale Fragestellung
- Was ist der UVC?
- Einsatz in der KB
- Schlussfolgerungen

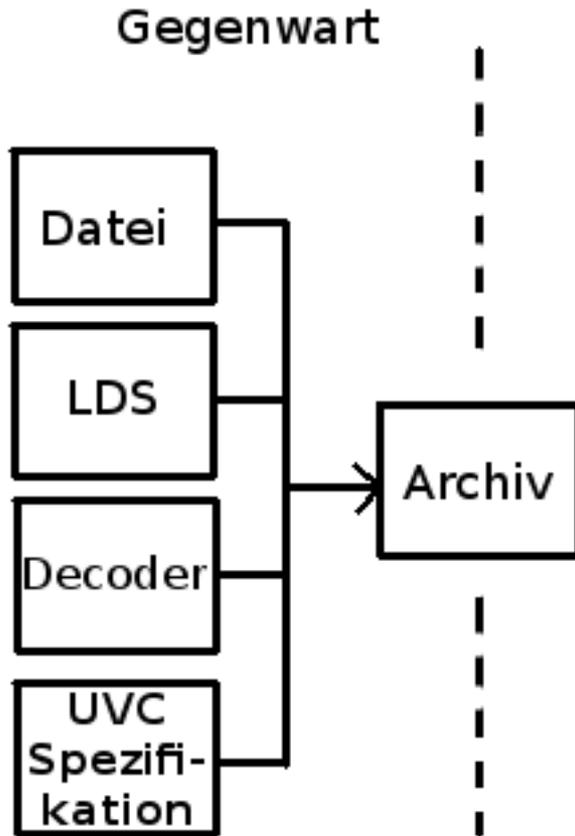
- Immer mehr Publikationen (vor allem wissenschaftliche) nur noch digital
- Viele Vorteile aber auch Gefahren
- Größtes Problem: dauerhafte Zugänglichkeit ist nicht sicher gestellt
- KB hat sich die „Sicherung der permanenten Zugänglichkeit von Informationen und Wissen“ zum Ziel gemacht
- Dabei: Erprobung eines Konzepts zur LZA mit Hilfe des UVC

- Dauerhaftes Archiv benötigt
- Dauerhaft bezüglich der Speicherung der Daten
- Aber vor allem auch dauerhaft bezüglich des Zugriffs auf die Daten
- Ein spezieller Emulator mit entsprechenden Funktionen könnte diese Aufgabe übernehmen

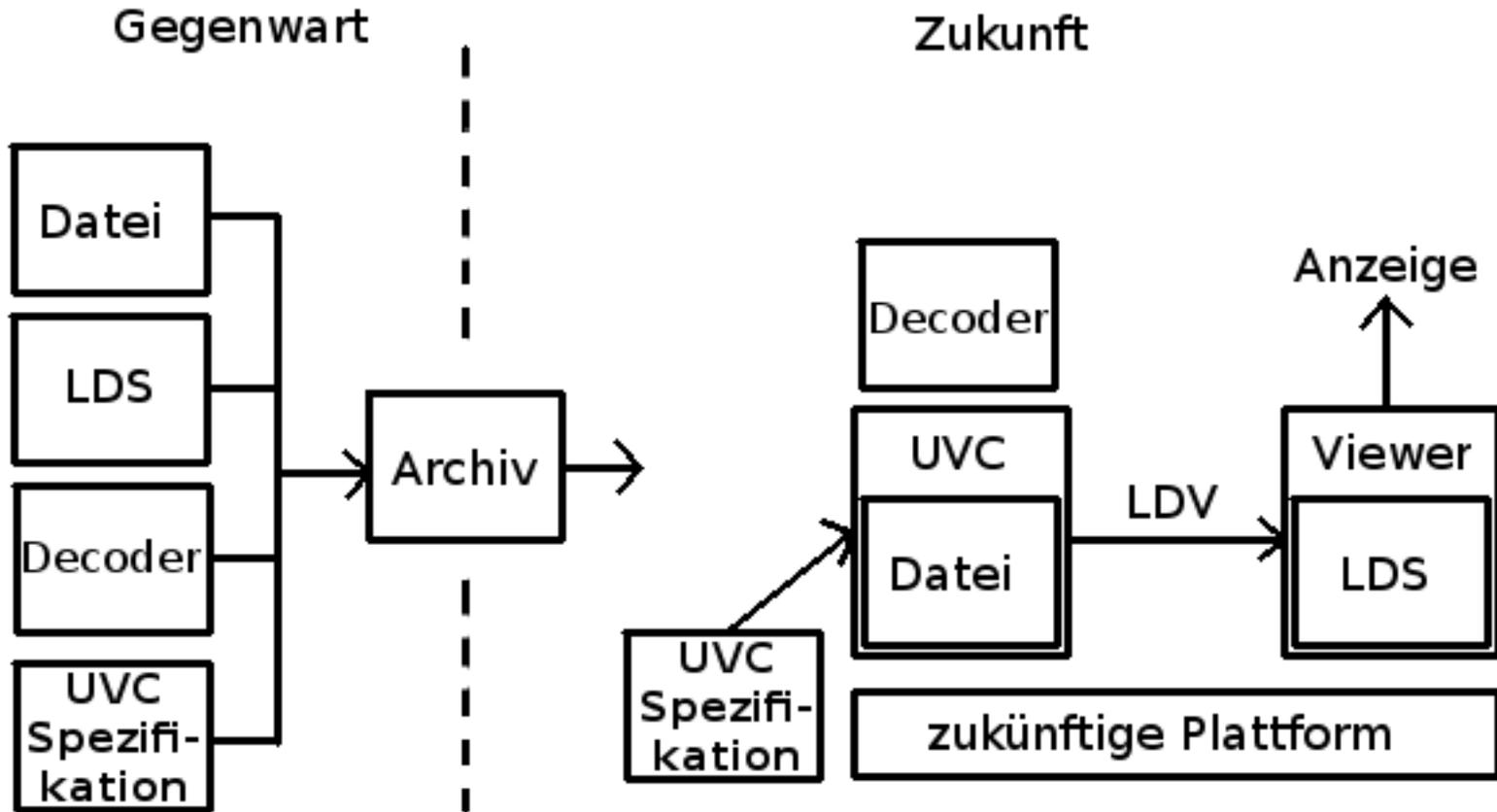
- „Wie können wir sicherstellen, dass ein Emulator, der imstande ist, archivierte digitale Objekte in ihrer (virtuellen) Originalumgebung auszuführen, korrekt auf zukünftigen Computern laufen wird, ohne zu Wissen, wie diese zukünftigen Plattformen aussehen werden?“
- Der UVC bietet eine mögliche Antwort

- Einleitung
- **Was ist der UVC?**
 - Konzept
 - Vorteile
 - Architektur
 - Speichermodell
 - Instruktionen
 - Bewertung
 - Realisierung
- Einsatz in der KB
- Schlussfolgerungen

- Unterscheidung zwischen:
 - Dem UVC als solches
 - Konzept zur LZA, das durch den UVC ermöglicht wird



Nach: [3]



Nach: [3]

■ LDS und in XML umgewandelte LDV einer Bilddatei

```
ELEMENT 01 [image](10,24+)  
  
ELEMENT 10 [imageSize](11,12)  
ELEMENT 11 [numScanLines]  
ELEMENT 12 [pixelsPerScanLine]  
  
ELEMENT 24 [pixel](25,29)  
  
ELEMENT 25 [colour](26,27,28)  
ELEMENT 26 [red]  
ELEMENT 27 [green]  
ELEMENT 28 [blue]  
  
ELEMENT 29 [pixelPosition](30,31)  
ELEMENT 30 [scanLine]  
ELEMENT 31 [pixelNum]
```

Quelle: [4]

```
<?xml version="1.0" encoding="UTF-8"?>  
<image>  
  <imageSize>  
    <numScanLines>38</numScanLines>  
    <pixelsPerScanLine>39</pixelsPerScanLine>  
  </imageSize>  
  <pixel>  
    <colour>  
      <red>255</red>  
      <green>255</green>  
      <blue>255</blue>  
    </colour>  
    <pixelPosition>  
      <scanLine>0</scanLine>  
      <pixelNum>0</pixelNum>  
    </pixelPosition>  
  </pixel>  
  ...
```

Quelle: eigener Probelauf des Demo UVC

- Weniger Aufwand als bei Migration jeder einzelnen Datei
- Nicht für jedes Dateiformat muss ein eigenes LDS entwickelt werden
- Oft reicht ein LDS für eine ganze Kategorie von Dateitypen (z.B. ein einziges Schema für Bilddateien wie JPEGs oder GIFs)
- Würde für jedes neue Dateiformat ein LDS und Decoder entwickelt, wäre der Zugriff so lange gesichert, wie es einen lauffähigen UVC gibt

- Bewusst so einfach gehalten, dass davon ausgegangen werden kann, dass der UVC auch in ferner Zukunft noch verstanden und implementiert werden kann
- Basis: segmentiertes Speichermodell
- 25 verschiedene Instruktionen

- Segmente haben theoretisch unbegrenzte Anzahl an Registern und theoretisch unbegrenzten Speicher
- Programme bestehen aus mehreren, miteinander interagierenden Teilen („Sections“)
- Jede Section wird in einem separaten Segment gespeichert
- Section kann auf alle Segmente ihres Adressraums zugreifen
- Reservierte Segmente: 0, 1, 2

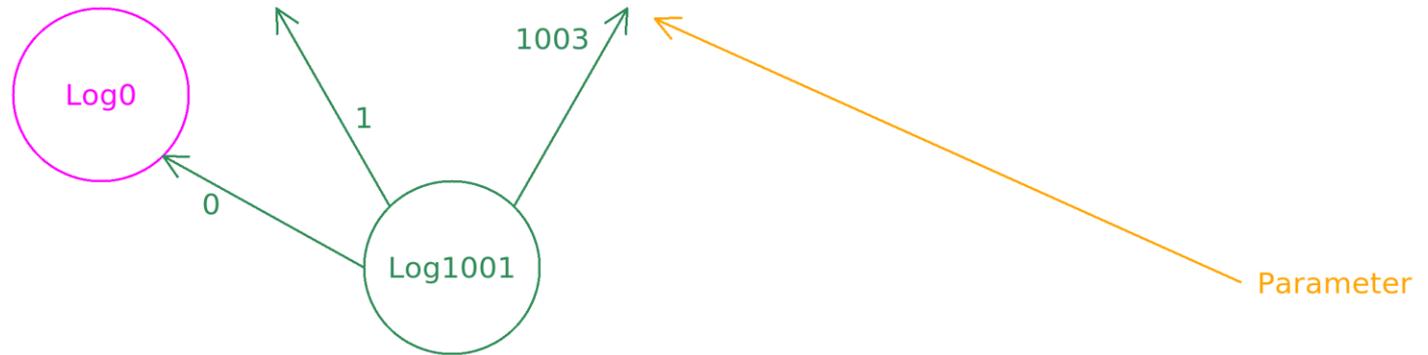
- Insgesamt 25 verschiedene Befehle
- Zur Manipulation von Daten, für arithmetische Operationen, Vergleiche, Sprünge, etc.
- Nur relativ einfache Operationen
- Keine „High-Level“ Befehlskonstrukte wie z.B. Schleifen

Beispiel: Rekursion

Physische
Segmente

Ph1	Ph2	Ph3	Ph4	Ph5	Ph6	Ph7	...
-----	-----	-----	-----	-----	-----	-----	-----

Logische
Segmente



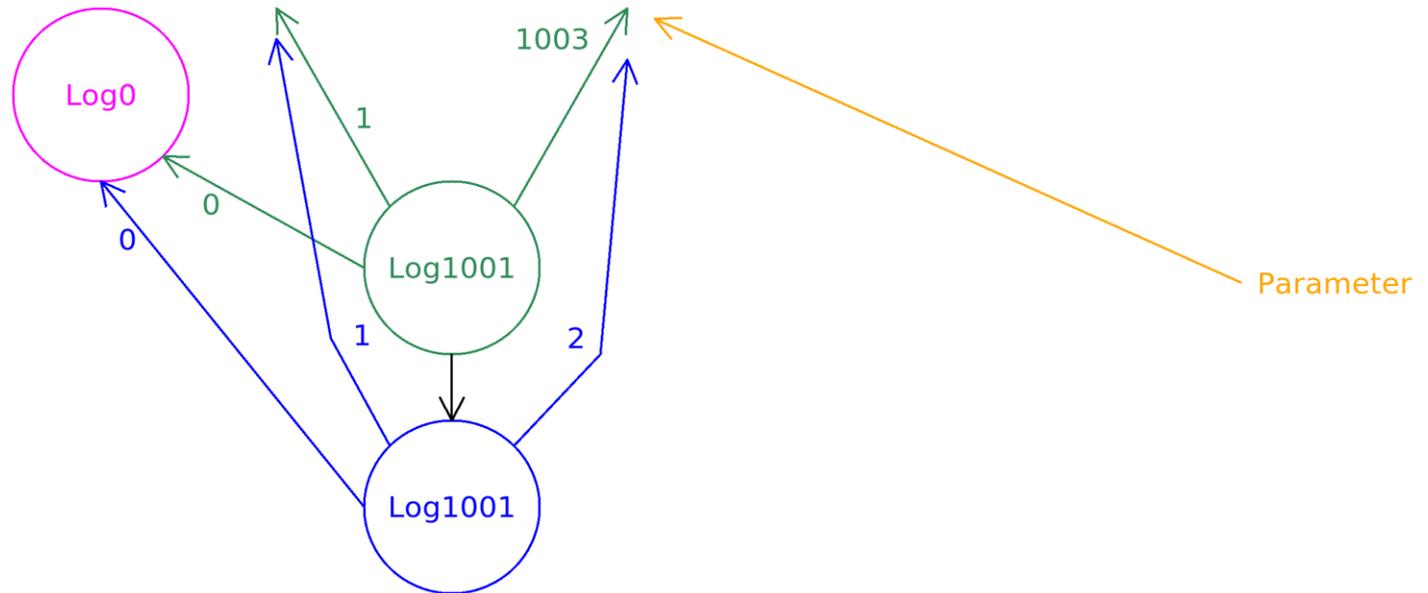
CALL 1001, 1003

Beispiel: Rekursion

Physische
Segmente

Ph1	Ph2	Ph3	Ph4	Ph5	Ph6	Ph7	...
-----	-----	-----	-----	-----	-----	-----	-----

Logische
Segmente



Beispiel: Rekursion

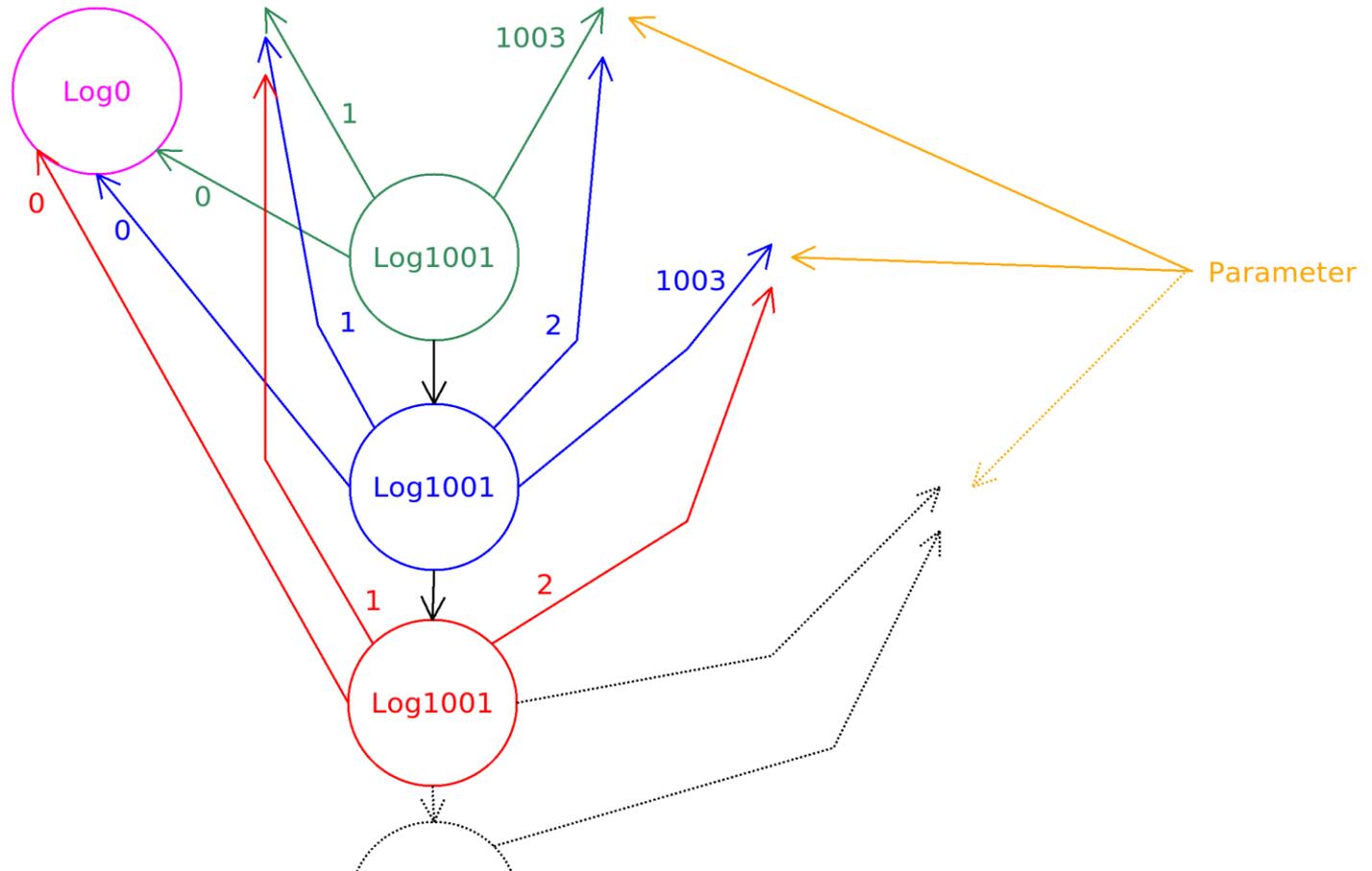
Physische Segmente

Ph1	Ph2	Ph3	Ph4	Ph5	Ph6	Ph7	...
-----	-----	-----	-----	-----	-----	-----	-----

Logische Segmente

CALL 1001, 1003

CALL 1001, 1003



- (bewusst) einfach
- Dadurch allerdings gewisse Einschränkung der Funktionalität
- Übersichtlich
- Zielorientiert
- Relativ leicht zu implementieren

- Tatsächlich wurde 2004 ein UVC im Rahmen der Erprobung des Konzepts implementiert¹
- Es mussten „nur“ folgende Komponenten umgesetzt werden:
 - Dispatch Table
 - Activation Stack
 - Memory Manager
 - Execution Manager
 - I/O Manager

¹ <http://www.alphaworks.ibm.com/tech/uvc>

- Einleitung
- Was ist der UVC?
- **Einsatz in der KB**
- Schlussfolgerungen

- Erprobung des UVC-basierten Konzepts zur LZA
- UVC Prototyp erfolgreich zur Archivierung und Betrachtung von Bilddateien eingesetzt
- Geplant: Ergänzung des e-Depot um ein „permanentes“ Zugriffstool
- Zur Zeit allerdings noch kein Einsatz

- Einleitung
- Was ist der UVC?
- Einsatz in der KB
- **Schlussfolgerungen**
 - Grenzen des Ansatzes
 - Forschungsschwerpunkte
 - Praxistauglichkeit

- Größtes Problem: Performance
- Anzeigen eines JPEG mit 595 mal 842 Pixeln dauert auf durchschnittlichen PC (Intel Core2Duo E7300 2x2,67GHz, 3GB RAM, Kubuntu 9.04, Sun Java 6) gut 10 Minuten¹
- Hochrechnung auf Dokument mit 300 Seiten: 2 Tage und 2 Stunden
- Auf Großrechner natürlich schneller aber nicht so sehr wie man erwartet

¹ ermittelt aus eigenen Tests des Prototyps

- Beim Prototyp stand Flexibilität im Vordergrund und bewusst nicht die Leistung
- Daher ist davon auszugehen, dass hier noch Verbesserungspotenzial vorhanden ist
- Tatsächlich wurde im Rahmen von „Planets“ ein schnellerer UVC entwickelt
 - C++ statt Java
 - Verwendung eigener Funktionen zur Speicherverwaltung statt Hashmaps etc.
 - Verwendung von Sprungtabellen statt Case- oder Select Anweisungen

- Hierdurch: Erhöhung der Leistung von ca. 17.000 Instruktionen pro Sekunde auf über 1 Million Instruktionen pro Sekunde
- Reduzierung des Zeitaufwandes für obiges Beispiel auf 10,2 Sekunden bzw. 51 Minuten
- Prototyp ist nur zur Archivierung bzw. Anzeigen von einzelnen Dateien
- Forschung an Erweiterung auf ganze Programme
- Bereits erfolgreich mit Tabellenkalkulationsprogramm getestet

- UVC ist noch nicht für den täglichen Einsatz bereit
- Nichtsdestotrotz ein sehr interessantes Konzept, das in der Zukunft wohl eine wichtige Rolle im Bereich der LZA spielen wird
- In welcher Form bleibt allerdings abzuwarten

Fragen?

1. <http://www.kb.nl/dnp/e-depot/factsandfigures-en.html>. 28.04.2009
2. Borghoff, Uwe M. ; Rödiger, Peter ; Scheffczyk, Jan: Long-Term Preservation of Digital Documents: Principles and Practices. Springer Verlag, 2006
3. Hoeven, Jeffrey van d.: UVC and emulation as preservation strategies. 2005. – Presentation for workshop on electronic publishing, Lund, Sweden
4. Hoeven, Jeffrey van d. ; Diessen, Raymond J. ; Meer, K. van d.: Development of a Universal Virtual Computer (UVC) for long-term preservation of digital objects. In: Journal of Information Science 31 (2005), Nr. 3, S. 196–208
5. Hoeven, Jeffrey van d. ; Lohman, Bram ; Verdegem, Remco: Emulation for Digital Preservation in Practice: The Results. In: The Journal of Digital Curation 2 (2007), Nr. 2, S. 196–208
6. Lorie, Raymond A.: Long Term Preservation of Digital Information. 2001
7. Lorie, Raymond A.: The UVC: a method for preserving digital documents - proof of concept. In: LTP reports series 4 (2002)
8. Lorie, Raymond A. ; Diessen, Raymond J.: UVC: A Universal Virtual Computer for Long-term Preservation of Digital Information. In: IBM Research Report (2005)

1. Oltmans, Erik ; Diessen, Raymond J. ; Wijngaarden, Hilde van: Preservation Functionality in a Digital Archive. In: 4th ACM/IEEE-CS joint conference on Digital libraries (2004), S. 279–286
2. Oltmans, Erik ; Wijngaarden, Hilde van: Digital preservation in practice: the e-Depot at the Koninklijke Bibliotheek. In: The Journal of Information and Knowledge Management Systems 34 (2004), Nr. 1, S. 21–26
3. Oltmans, Erik ; Wijngaarden, Hilde van: The KB e-Depot digital archiving policy. In: Library Hi Tech 24 (2006), Nr. 4, S. 604–613
4. E-Mail Korrespondenz mit Jasper Schroder, IBM Netherlands, siehe Anhang der schriftlichen Ausarbeitung