

RECOIN
Retrieval Component Integration

Präsentation von Konzepten und
Entwicklungsstand
4. Hildesheimer Evaluierungs- und
Retrieval (HIER) Workshop

Mittwoch, 20. Juli 2005
Jan H. Scheufen
j.h.scheufen@gmx.net

Recoin – Retrieval Component Integration

Ablauf

- Vorstellung von Konzepten (ca. 15 Min.)
- Demonstration der Workbench (ca. 10 Min.)
- Abschlussdiskussion
- Kaffeepause

20. Juli 2005 <http://www.recoin.org> Seite 2

Recoin – Retrieval Component Integration

Geschichte & Hintergrund

Erfahrungen in Projekten haben gezeigt:

- **Doppelentwicklungen**
Code von Vorgängerprojekten ist nur schwer wieder
verwendbar. Dokumentation, Programmierstile, etc.
- **Zeitverlust**
Aufwand für Basiskomponenten wird unterschätzt.
GUI, Steuerungsmechanismen, Businesslogik, etc.
- **Fehlende Vergleichs-/Metadaten**
Vergleich und Evaluierung von Implementierungen nur
schwer möglich.

20. Juli 2005 <http://www.recoin.org> Seite 3

Recoin – Retrieval Component Integration

Ziele

- Erhöhung der Wiederverwendbarkeit von
Algorithmen und Tools.
- Minimierung des Aufwands zum Aufbau von
IR Systemen.
- Offener, erweiterbarer Baukasten für
Retrieval Komponenten.
- Plattform für experimentelles Information
Retrieval

20. Juli 2005 <http://www.recoin.org> Seite 4

Recoin – Retrieval Component Integration

Retrieval Component Integration??

- **Retrieval**
Information Retrieval in Lehre, Forschung & Entwicklung
- **Component**
Software Komponenten zur Kapselung kohärenter
Aktionen oder Arbeitsschritte des Retrievalprozesses.
- **Integration**
Integration, Verwaltung und Konfiguration von Retrieval-
Komponenten in einem Framework zum flexiblen Aufbau
von IR Systemen.

20. Juli 2005 <http://www.recoin.org> Seite 5

Recoin – Retrieval Component Integration

Konzepte

I. Aufteilung des Retrievalprozesses

20. Juli 2005 <http://www.recoin.org> Seite 6

Konzepte - Retrievalprozess

Der Retrievalprozess lässt sich in verschiedene Schritte einteilen, in denen Daten verarbeitet werden und gegebenenfalls neue Daten entstehen.

Beispiele:

- IN: Zeichenkette -> Tokenizing, Stoppwortentfernung, Stemming -> OUT: Query
- IN: Query -> Matching -> OUT: Ranking
- IN: Rankings -> Gewichtete Fusion -> Gesamt-Ranking

Konzepte - Retrievalprozess

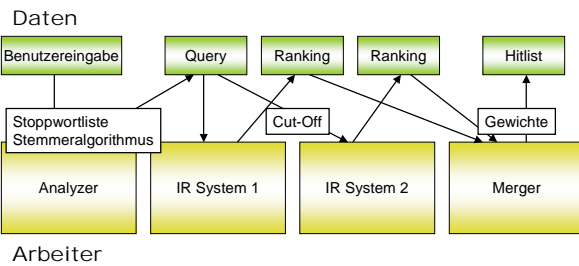
Schlussfolgerung:

Es gibt zwei Objekttypen, deren Implementierungen stark variieren können.

- **Datenobjekte**, die die Eingangsdaten und Ergebnisse sowie Zwischenergebnisse des Prozesses repräsentieren.
- **Arbeiterobjekte**, die Datenobjekte verarbeiten und/oder erzeugen.

Konzepte – Retrievalprozess

Ablauf des Retrievalprozesses



Konzepte - Retrievalprozess

Zusammengefasst:

- Arbeiter repräsentieren parametrisierbare Methoden, die Datenobjekte verarbeiten.

Forderung:

- Flexibilität bei der Verkettung, Konfiguration und Parametrisierung der Arbeiter- und Datenobjekte.

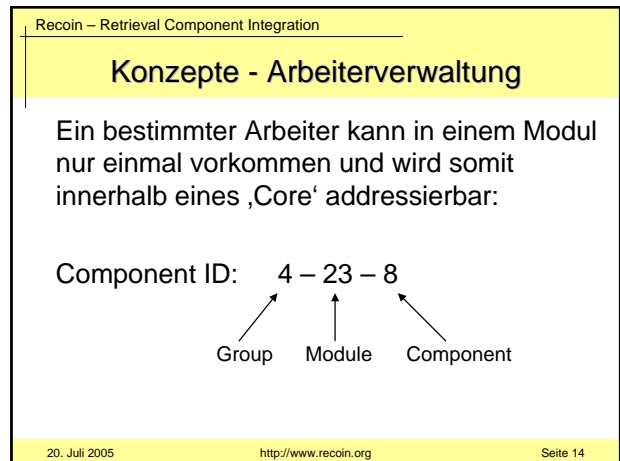
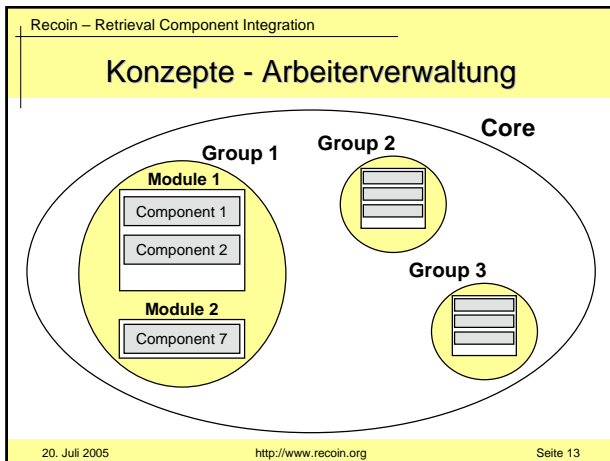
Konzepte

II. Verwaltung von Arbeitern

Konzepte - Arbeiterverwaltung

Arbeiter (*Components*) sind in verschiedenen Einheiten angeordnet:

Core → Group $\xrightarrow{\text{groß} \rightarrow \text{klein}}$ Module → Component



Recoin – Retrieval Component Integration

Konzepte - Arbeiterverwaltung

Zusammengefasst:

- ➔ Arbeiter sind nicht starr miteinander verkettet, sondern existieren als autonome, adressierbare Einheiten in einem Core.

Forderung:

- ➔ Steuerung des Retrievalprozesses durch *Instruktionen*, die festlegen, welche Objekte in welcher Reihenfolge von welchen Arbeitern wie verarbeitet werden.

20. Juli 2005 <http://www.recoin.org> Seite 15

Recoin – Retrieval Component Integration

Konzepte

III. Datentransport und Prozesssteuerung durch Container

20. Juli 2005 <http://www.recoin.org> Seite 16

Recoin – Retrieval Component Integration

Konzepte - Container

- ➔ Verwendung von Container-Objekten, die die Ausgangs-, Zwischen- und Endergebnisse transportieren
- ➔ Container enthalten Instruktionen, welche Objekte von welchen Arbeitern mit welchen Parametern verarbeitet werden sollen.

20. Juli 2005 <http://www.recoin.org> Seite 17

Recoin – Retrieval Component Integration

Konzepte - Container

XML-codierte Instruktionen:

```
<order group="3" module="8" component="5">
  <parameter name="cutoff">500</parameter>
  <containerID
    group ="2" module="1" component="3"/>
  <containerID
    group ="2" module="4" component="1">1:4
  </containerID>
</group>
```

20. Juli 2005 <http://www.recoin.org> Seite 18

RecoIn – Retrieval Component Integration

Konzepte – Container

Schematischer Aufbau eines Containers

20. Juli 2005 <http://www.recoIn.org> Seite 19

RecoIn – Retrieval Component Integration

Konzepte – Container

Zusammengefasst:

- Container enthalten sowohl Ausgangsdaten als auch Zwischen- und Ergebnisse.
- Weiterhin enthalten sie die Instruktionen, die zu diesen Ergebnissen geführt haben.
- Container eignen sich somit auch zur Aufbewahrung von Experimenten zur späteren Evaluierung.

20. Juli 2005 <http://www.recoIn.org> Seite 20

RecoIn – Retrieval Component Integration

Konzepte

IV. Dynamisches Laden von Klassen *Reflection*

20. Juli 2005 <http://www.recoIn.org> Seite 21

RecoIn – Retrieval Component Integration

Konzepte – Reflection

- Zur Sicherung der Erweiterbarkeit können dem RecoIn Framework Arbeiter- und Datenobjekte zur Laufzeit hinzugefügt werden.
- Ein Entwickler deklariert lediglich seine Erweiterungen und das Framework lädt sie über ihren Klassennamen.

20. Juli 2005 <http://www.recoIn.org> Seite 22

RecoIn – Retrieval Component Integration

Konzepte – Reflection

Klassendiagramme von Daten und Arbeitern

20. Juli 2005 <http://www.recoIn.org> Seite 23

RecoIn – Retrieval Component Integration

Konzepte – Reflection

- Bei der Erzeugung von Objekten anhand ihres Klassennamens wird der so genannte *Null-Konstruktor* verwendet, dem keine Argumente übergeben werden können.
- Die verlangte Attribuierung und Parametrisierung der Arbeiter und Daten wird daher über einen speziellen Mechanismus durchgeführt, dessen Kontrolle beim Programmierer der Erweiterung liegt.

20. Juli 2005 <http://www.recoIn.org> Seite 24

Zusammengefasst:

- Arbeiter- und Datenobjekte lassen sich in Recoin per Deklaration hinzufügen.
- Dies erlaubt den Aufbau von Sammlungen dieser Objekte zur gemeinsamen Nutzung.
- Attribute und Parameter werden vom Programmierer der Erweiterung vorgegeben.

Demonstration

V. Die Eclipse IR Workbench

Demonstration – Eclipse Workbench

- Aufteilung von Recoin in eine Bibliothek und eine Anwendung.
- Die Bibliothek enthält alle implementierungsunabhängigen Klassen und unterstützt den Aufbau einer Anwendung.
- Die Eclipse Plug-Ins sind nur ein mögliches Beispiel für eine Implementierung.
- Eclipse stellt Ablaufumgebung, User Interfaces, u.v.m zur Verfügung.

Demonstration – Eclipse Workbench

‘Roundtrip-Engineering‘ mit Eclipse

Programmierung, Tests, Debugging,
Distribution, Anwendung

→ alles mit einer Oberfläche

Demonstration – Eclipse Workbench

Verwaltung von Erweiterungen über Eclipse *Extension-Points*.

```
<extension point="org.recoin.base.componentData">
  <data
    name="DemoData"
    class="my.recoin.demo.DemoData">
    <description>Datenobjekt zu
    Demonstrationszwecken.</description>
  </data>
</extension>
```

Demonstration – Eclipse Workbench

Die Eclipse Workbench

Recoin – Retrieval Component Integration

Fazit

VI. Fazit

20. Juli 2005 <http://www.recoin.org> Seite 31

Recoin – Retrieval Component Integration

Fazit

Recoin ist:

- ➔ Ein abstraktes Framework zum Aufbau und zur Steuerung arbeitsteiliger Prozesse in Form der **Recoin Bibliothek**.
- ➔ Eine Implementierung dieses Frameworks in Form von **Eclipse Plug-Ins** unter Nutzung der Eclipse Features.

20. Juli 2005 <http://www.recoin.org> Seite 32

Recoin – Retrieval Component Integration

Fazit

Vorteile:

- ➔ Einsatz und Erweiterung des Frameworks ist der Fantasie des Entwicklers überlassen.
- ➔ Anhand des Klassennamens eindeutig identifizierbare Methoden und Objekte gepaart mit Metadaten, u.a. in Form von Attributen und Parametern, können als Grundlage für MIMOR dienen.
- ➔ Effektive Minimierung des Arbeitsaufwands für Entwickler.

20. Juli 2005 <http://www.recoin.org> Seite 33

Recoin – Retrieval Component Integration

Ausblick

Schwachstellen:

- ➔ Container-Konzept lässt derzeit nicht zu, dass Arbeiter mehrere Objekte ablegen.
- ➔ Interface zur Erstellung von Instruktionen ist noch nicht ausgereift.

20. Juli 2005 <http://www.recoin.org> Seite 34

Recoin – Retrieval Component Integration

Konzepte

VI. Ausblick

20. Juli 2005 <http://www.recoin.org> Seite 35

Recoin – Retrieval Component Integration

Ausblick

- ➔ Weiterentwicklung und Stabilisierung der Recoin Bibliothek.
- ➔ Ausbau der Funktionen der Recoin IR Workbench Plug-Ins.
- ➔ Wiederholung älterer CLEF Experimente und Evaluierung.
- ➔ Teilnahme an CLEF 2006.

20. Juli 2005 <http://www.recoin.org> Seite 36

Fragen?

Noch Fragen?

Fragen, Anregungen, Kritik, Lob und Angebote zur finanziellen Unterstützung sind jederzeit willkommen!

j.h.scheufen@gmx.net

<http://www.recoin.org>