

Das SeDaM – Projekt¹

Dokumente finden – natürlich-sprachlich anfragen

Jan Pretzel, GECKO mbH Rostock (jpr@gecko.de)

Geschichte

GECKO mbH arbeitet seit 1998 am Thema „Suchen in Textdokumenten“. Damals war GECKO mbH einer von vier Projektpartnern im BMBF-Projekt „GETESS“, an den noch das DFKI Saarbrücken, AIFB Karlsruhe, sowie DBIS der Uni Rostock beteiligt waren. Entstanden ist eine Technologiestudie, die die grundsätzliche Machbarkeit einer neuartigen Suche nach Informationen in natürlich-sprachlichen Textdokumenten nachweist.

Weitere wichtige und ergänzende Arbeiten folgten, so daß GECKO mbH jetzt in der Lage ist, einen funktionierenden Prototypen zu präsentieren.

Im Folgenden soll die Technologie hinter diesem Prototypen vorgestellt werden.

Anfrageprozess (Abbildung 1)

SeDaM ist als Web Service konzipiert und ermöglicht somit unterschiedlichen Anwendungen Zugriff auf die Dienste.

Client Mit dem SeDaM-FQM (Flexible Query Manager) existiert eine Schnittstelle, die mittels SOAP Dienste des SeDaM-DMA (DialogManager) abrufen. SeDaM-FQM ist so konzipiert, daß es möglich ist, personalisierte Sichten auf die Daten der zu suchenden Dokumente zu legen.

NL Parser Der SeDaM-TMA analysiert natürlich-sprachlichen Text und liefert anhand der syntaktischen Struktur den Syntaxbaum oder Paare von Konzepten (Dumbbells), die in einer möglichen Beziehung untereinander stehen. Die DumbBells sind **ontologieunabhängig!**

Server Der SeDaM-DMA als zentrale Schaltstelle ruft verschiedene Dienste auf (SeDaM-TMA, Knowledge Base etc.). Die DumbBells werden mittels Knowledge Bases in Frame-Strukturen eingebettet, die zur Dokumentenrecherche in der Abstraktdatenbank herangezogen werden. Darüber hinaus werden diese Frames mit Informationen für den Klärungsdialog angereichert.

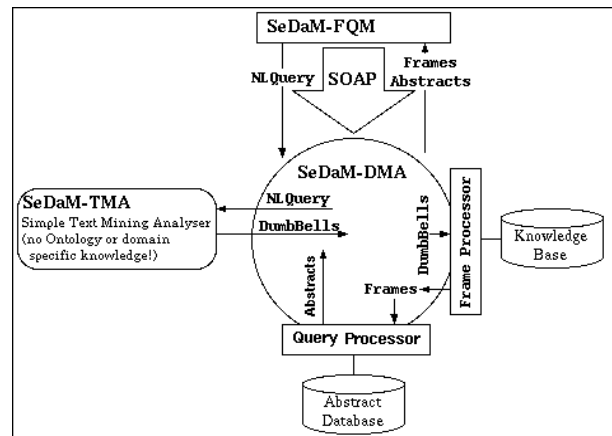


Abbildung 1: Architektur SeDaM-Prototyp

Gathering

Das Einsammeln des Dokumente erfolgt in einem eigenständigen Prozess, wobei die Inhalte der Dokumente als Abstracts gespeichert werden. Die Daten der Dokumente werden faktisch komprimiert.

Als **Abstracts** bezeichnen wir die Menge von DumbBells zu einem Dokument. Es wird ausschließlich der SeDaM-TMA genutzt. Die ermittelten Konzeptpaare (DumbBells) werden in einer relationalen Datenbank (PostgreSQL) abgelegt und stehen dann dem Anfrageprozess zur Verfügung.

Anfrageprozess (Beispiel)

Folgende Anfrage an das SeDaM System soll die Funktionalität des Systems verdeutlichen:

Eine Bank in der Stadt

Nachdem der SeDaM-DMA (Server) die Anfrage entgegengenommen hat, wird der NL-Text zuerst an den SeDaM-TMA geschickt. Die Verarbeitung des NL-Textes erfolgt in zwei Schritten.

1. Syntaxbaum

```
<Syntax>
<NP begin="0" end="2" >
  <DET string="eine" />
  <Nom string="Bank" />
</NP>
<PP begin="2" end="5" >
  <Prep string="in" />
  <DET string="der" />
  <Nom string="Stadt" />
</PP>
</Syntax>
```

2. Dumbbells

```
<DumbBells>
<DumbBell heuristic=NP-PP>
  <NP begin="0" end="2" >
    <DET string="eine" />
    <Nom string="Bank" />
  </NP>
  <PP begin="2" end="5" >
    <Prep string="in" />
    <DET string="der" />
    <Nom string="Stadt" />
  </PP>
</DumbBell>
</DumbBells>
```

Aus den DumbBells erzeugt der Frame-Prozessor des SeDaM-DMA mit Hilfe der Knowledge Base (Wissensnetz/Ontologie) eine Frame-Struktur, die interne Wissensrepräsentationsstruktur von SeDaM.

```
<FrameList>
<Frame name="Sitzmoebel">
  <SlotList>
    <Slot>
      <Frame name="Stadt" />
    </Slot>
  </SlotList>
</Frame>
<Frame name="GeldInstitut">
  <SlotList>
    <Slot>
      <Frame name="Stadt" />
    </Slot>
  </SlotList>
</Frame>
```

Aus den Frames berechnet der Query Prozessor Anfragen an die Dokument-Datenbank (Abstraktdatenbank)

Der SeDaM-DMA ermittelt so die Dokumente, die zum Frame „passen“.

In einem letzten Schritt werden dann die Frames mit weiteren Informationen angereichert (rot), die dann für den Klärungsdialog mit dem Benutzer zur Verfügung stehen.

```
<FrameList>
<Frame id=1 name="Sitzmoebel">
  <SlotList>
    <Slot id=1 name=location type=location>
      <FrameList>
        <Frame id=2 name="Stadt" />
      </FrameList>
    </Slot>
    <Slot id=3 name=consists-of type=Material clarify=Yes>
      <FrameList>
        <Frame id=5 name="Holz" />
        <Frame id=5 name="Plast" />
        <Frame id=5 name="Metall" />
      </FrameList>
    </Slot>
  </SlotList>
</Frame>
<Frame id=3 name="GeldInstitut">
  <SlotList>
    <Slot>
      <FrameList>
        <Frame id=4 name="Stadt" />
      </FrameList>
    </Slot>
    <Slot id=4 name=opens type=TimeDate clarify=Yes>
      <FrameList>
        <Frame id=5 name="Montag" />
        <Frame id=5 name="Dienstag" />
        <Frame id=5 name="Freitag" />
      </FrameList>
    </Slot>
  </SlotList>
</Frame>
```

Verbal ausgedrückt würde sich folgender Dialog entspannen (Nutzer N / SsDaM-System S)

N1> (ich suche Dokumente über) eine Bank in der Stadt

S1 > gefunden: xxx Dokumente;

1. Was meinen Sie mit „Bank“?

1. Ein Sitzmöbel oder
2. ein Geldinstitut?

2. In den xxx Dokumenten ist noch folgende Information enthalten:

1. Aus welchem Material soll „Sitzmöbel“ sein? (Holz, Plast, Metall)
2. Wann soll „Geldinstitut“ geöffnet haben? (Montag, Dienstag, Freitag)

Der Nutzer navigiert also mittels Klärungsdialog durch die Daten der Dokumente.

So wird das gewünschten Dokumente gefunden!

Kontakt:

Jan Pretzel
GECKO mbH Rostock
Herweghstr. 20
D-18055 Rostock
Tel: +381 454 88 0
mail: jpr@gecko.de