GATE

Architekturen für Language Engineering

Christopher Holm, Martin Burmester und Benjamin Schrage

Universität Hamburg

FB Informatik – AB NATS

Proseminar KI Architekturen

Dr. Cristina Vertan

Struktur des Vortrags

- Einführung – Worum geht es bei GATE?
- Implementation von GATE1
- zwei Beispiele für den Einsatz von GATE
- Ausblick – GATE2, GATE3

GATE? SALE!

GATE? SALE!

Bei der **S**oftware **A**rchitecture for **L**anguage **E**ngineering handelt es sich um die “Schnittmenge“ der beiden Teilgebiete

- Computer-Infrastruktur für Software-Entwicklung (Software Architecture)
- Coumputerverarbeitung von menschlicher Sprache (Language Engineering)

GATE? SALE!

Bei der **S**oftware **A**rchitecture for **L**anguage **E**ngineering handelt es sich um die “Schnittmenge“ der beiden Teilgebiete

- Computer-Infrastruktur für Software-Entwicklung (Software Architecture)
- Coumputerverarbeitung von menschlicher Sprache (Language Engineering)

Zielsetzung: SALE soll ein generelles, erweiterbares Unterstützungs-Werkzeug für Forscher und Entwickler sein, die Software zur Sprachverarbeitung herstellen

Was ist Gate, was ist es nicht?

- Abdeckung eines breiten Spektrums an Sprachverarbeitungs-Bereichen
- Bereitstellung von vielfach genutzten und benötigten Operationen

Was ist Gate, was ist es nicht?

- Abdeckung eines breiten Spektrums an Sprachverarbeitungs-Bereichen
- Bereitstellung von vielfach genutzten und benötigten Operationen

? Software, die ein Forschungsziel löst

? Software, die Implementation von oft auftretenden Aufgaben anbietet

Was ist Gate, was ist es nicht?

- Abdeckung eines breiten Spektrums an Sprachverarbeitungs-Bereichen
- Bereitstellung von vielfach genutzten und benötigten Operationen

× Software, die ein Forschungsziel löst

√ Software, die Implementation von oft auftretenden Aufgaben anbietet

Was ist Gate, was ist es nicht? (II)

Was ist Gate, was ist es nicht? (II)

- ? **Wie** wird ein oft gebrauchter Algorithmus umgesetzt?
- ? **Welche**/r Algorithmus/Repräsentation wird verwendet?

Was ist Gate, was ist es nicht? (II)

- √ **Wie** wird ein oft gebrauchter Algorithmus umgesetzt?
- × **Welche**/r Algorithmus/Repräsentation wird verwendet?

Was ist Gate, was ist es nicht? (II)

- √ **Wie** wird ein oft gebrauchter Algorithmus umgesetzt?
- × **Welche**/r Algorithmus/Repräsentation wird verwendet?

Ausnahmen:

Was ist Gate, was ist es nicht? (II)

- √ **Wie** wird ein oft gebrauchter Algorithmus umgesetzt?
- × **Welche**/r Algorithmus/Repräsentation wird verwendet?

Ausnahmen:

- Das Thema wird noch erforscht, ist aber übergreifend nützlich und verwendbar

Was ist Gate, was ist es nicht? (II)

- √ **Wie** wird ein oft gebrauchter Algorithmus umgesetzt?
- × **Welche**/r Algorithmus/Repräsentation wird verwendet?

Ausnahmen:

- Das Thema wird noch erforscht, ist aber übergreifend nützlich und verwendbar
- Es wird im Kontext entwickelt

Für wen ist Gate?

- Designexperten und Programmierer, die Software zur Verarbeitung natürlicher Sprache herstellen
- normale Programmierer, die experimentelle Software zu Forschungszwecken herstellen
- Sprachforscher, die Experimente mit der Software anderer machen
- Sprachverarbeitungswissenschaften- und -technik-Lehrer
- Systemadministratoren, die Sprachforscher unterstützen

GATE Use Cases

- Forschungs- und Diagnose- (R&D) Arbeiter in der Sprachverarbeitung unterstützen

GATE Use Cases

- Forschungs- und Diagnose- (R&D) Arbeiter in der Sprachverarbeitung unterstützen
- Die Architektur dokumentieren, warten und unterstützen

GATE Use Cases

- Forschungs- und Diagnose- (R&D) Arbeiter in der Sprachverarbeitung unterstützen
- Die Architektur dokumentieren, warten und unterstützen
- Die Benutzung der Architektur in und für unterschiedliche Sprachen ermöglichen

GATE Use Cases

- Forschungs- und Diagnose- (R&D) Arbeiter in der Sprachverarbeitung unterstützen
- Die Architektur dokumentieren, warten und unterstützen
- Die Benutzung der Architektur in und für unterschiedliche Sprachen ermöglichen
- Ausgezeichnete Softwaretechnik in der Sprachverarbeitung fördern

GATE Use Cases (II)

- Ausnutzung der Vorteile des GATE-Gerüsts

GATE Use Cases (II)

- Ausnutzung der Vorteile des GATE-Gerüsts
- Verfügbare Komponenten müssen selbständig erkannt, geladen und initialisiert werden

GATE Use Cases (II)

- Ausnutzung der Vorteile des GATE-Gerüsts
- Verfügbare Komponenten müssen selbständig erkannt, geladen und initialisiert werden
- System-Erstellung aus einzelnen Komponenten ermöglichen

GATE Use Cases (II)

- Ausnutzung der Vorteile des GATE-Gerüsts
- Verfügbare Komponenten müssen selbständig erkannt, geladen und initialisiert werden
- System-Erstellung aus einzelnen Komponenten ermöglichen
- System-Erstellung basiert auf rechnerverteilten Komponenten (Distributed processing)

GATE Use Cases (III)

- Asynchrone Auführung der Komponenten (Parallel processing)

GATE Use Cases (III)

- Asynchrone Auführung der Komponenten (Parallel processing)
- Assoziation von strukturierten Daten mit Sprachressourcen und Verarbeitungsressourcen

GATE Use Cases (III)

- Asynchrone Auführung der Komponenten (Parallel processing)
- Assoziation von strukturierten Daten mit Sprachressourcen und Verarbeitungsressourcen
- Gemeinsamkeiten verbundener Komponenten minimieren

GATE Use Cases (III)

- Asynchrone Auführung der Komponenten (Parallel processing)
- Assoziation von strukturierten Daten mit Sprachressourcen und Verarbeitungsressourcen
- Gemeinsamkeiten verbundener Komponenten minimieren
- Bereitstellung einheitlicher, einfacher Methoden zum Zugriff auf Datenkomponenten

GATE Use Cases (IV)

- Dokumente auf effiziente Weise managen

GATE Use Cases (IV)

- Dokumente auf effiziente Weise managen
- Die Nutzung diverser Dokumentformate

GATE Use Cases (IV)

- Dokumente auf effiziente Weise managen
- Die Nutzung diverser Dokumentformate
- Theorie-neutrale, formatunabhängige Dokumentkommentare erlauben

GATE Use Cases (IV)

- Dokumente auf effiziente Weise managen
- Die Nutzung diverser Dokumentformate
- Theorie-neutrale, formatunabhängige Dokumentkommentare erlauben
- Erstellung und Wartung von sprachbeschreibenden Sprachressourcen fördern

GATE Use Cases (V)

- Indizierung und Abruf diverser Datenstrukturen liefern

GATE Use Cases (V)

- Indizierung und Abruf diverser Datenstrukturen liefern
- Eine Bibliothek mit bekannten Algorithmen für native Datenstrukturen bereitstellen

GATE Use Cases (V)

- Indizierung und Abruf diverser Datenstrukturen liefern
- Eine Bibliothek mit bekannten Algorithmen für native Datenstrukturen bereitstellen
- Bereitstellung einfacher Methoden zum Vergleichen von Datenstrutkturen

GATE Use Cases (V)

- Indizierung und Abruf diverser Datenstrukturen liefern
- Eine Bibliothek mit bekannten Algorithmen für native Datenstrukturen bereitstellen
- Bereitstellung einfacher Methoden zum Vergleichen von Datenstrutkturen
- Alle architekturnativen Datenstrukturen sollten von Dauer sein

GATE Use Cases (VI)

- Das GATE-Gerüst soll eine Nutzung in diversen Zusammenhängen ermöglichen

GATE Use Cases (VI)

- Das GATE-Gerüst soll eine Nutzung in diversen Zusammenhängen ermöglichen
- Datenaustausch mit anderen Infrastrukturen sowie Einbettung in andere Umgebungen muss möglich sein

GATE Use Cases (VI)

- Das GATE-Gerüst soll eine Nutzung in diversen Zusammenhängen ermöglichen
- Datenaustausch mit anderen Infrastrukturen sowie Einbettung in andere Umgebungen muss möglich sein
- Sprachverarbeitungs-Datenstrukturen müssen manipulierbar sein

GATE Use Cases (VI)

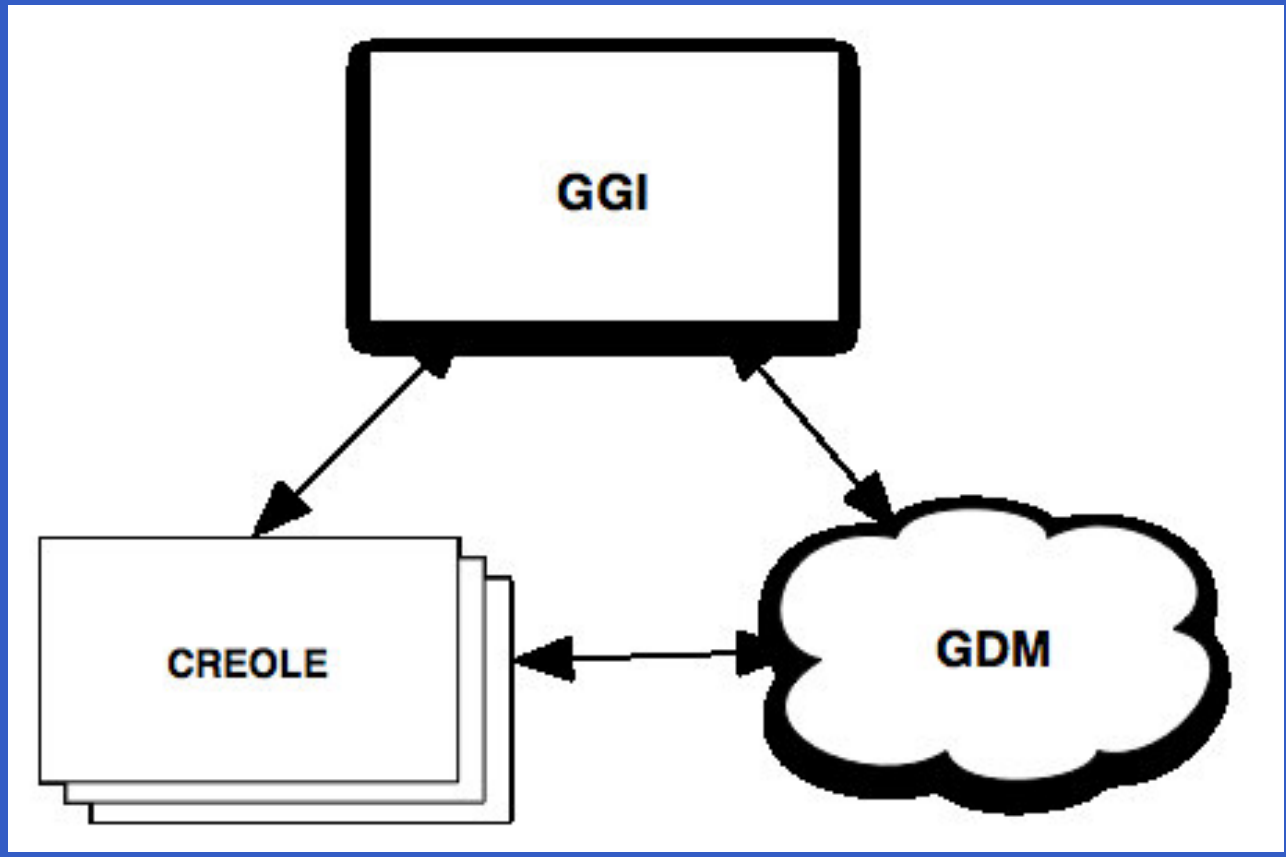
- Das GATE-Gerüst soll eine Nutzung in diversen Zusammenhängen ermöglichen
- Datenaustausch mit anderen Infrastrukturen sowie Einbettung in andere Umgebungen muss möglich sein
- Sprachverarbeitungs-Datenstrukturen müssen manipulierbar sein
- Zugriff auf das GATE-Gerüst und seine Dienste ermöglichen, sowie die Entwicklung von Sprachverarbeitungs-Experimenten und -Anwendungen fördern

Implementation von GATE1

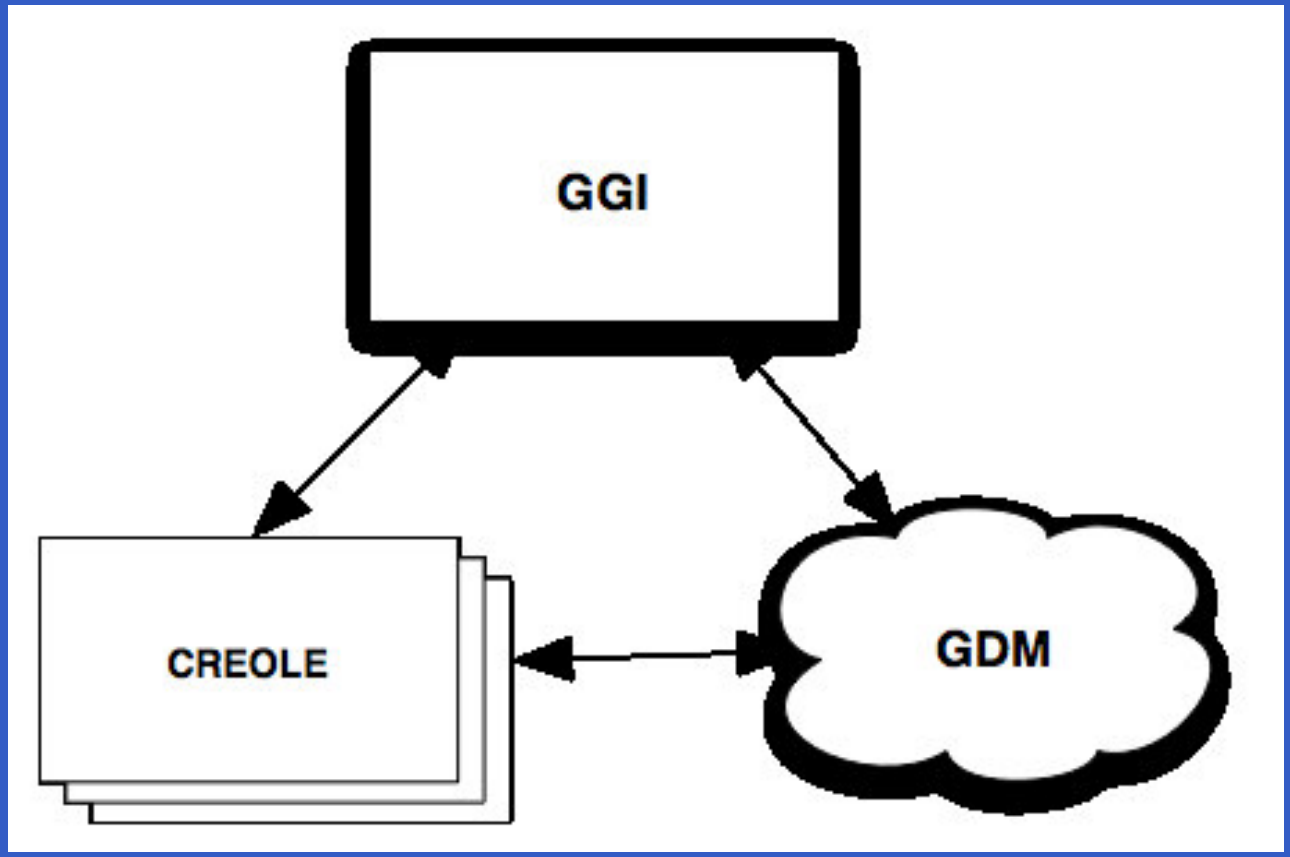
3 Komponenten Framework

- GDM – **G**ATE **D**ocument **M**anager
- GGI – **G**ATE **G**raphical **I**nterface
- CREOLE – **C**ollection of **RE**usable **O**bjects for **L**anguage **E**ngineering

Kommunikation der Komponenten



Kommunikation der Komponenten



Wichtig: keine direkte Kommunikation zwischen den CREOLE Modulen

GDM

- verwaltet Dokumente und *Annotations*
- basiert auf dem TIPSTER System (in Konkurrenz zu SGML/XML)
- implementatiert in C++
- stellt API zur Verfügung

CREOLE

- CREOLE Module (Objekte) erledigen die eigentliche Arbeit
- Schnittstelle:
  - Tcl, C/C++ und Java Code kann eingebunden werden
  - über *Wrapper* kann auf beliebige Software zugegriffen werden $\Rightarrow$ z.B. Perl, Prolog und LISP können eingesetzt werden
  - auf TIPSTER ausgelegter Code kann mit wenig Overhead verwendet werden

CREOLE – Metadaten

Metadaten liefern Informationen über:

- Möglichkeiten eines Moduls
- Reihenfolge der Ausführung
- Visualisierung

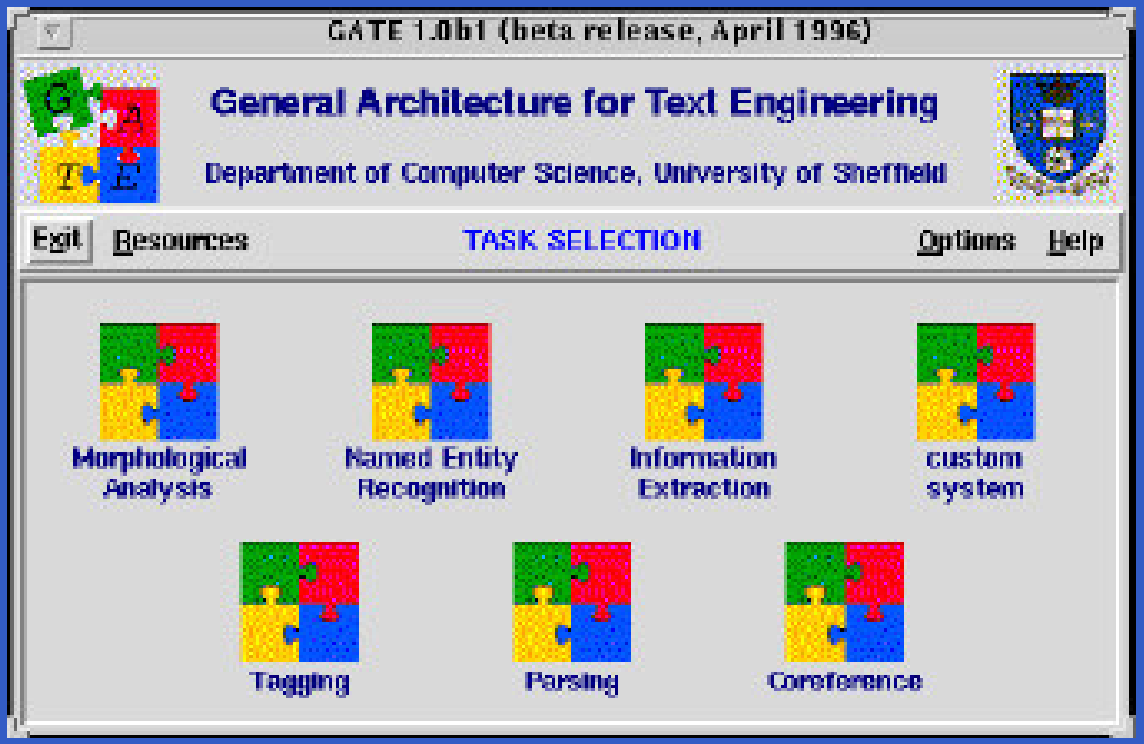
Metadaten bestehen aus:

- preconditions
- postconditions
- Informationen für die Visualisierung

CREOLE – Metadaten – ein Beispiel

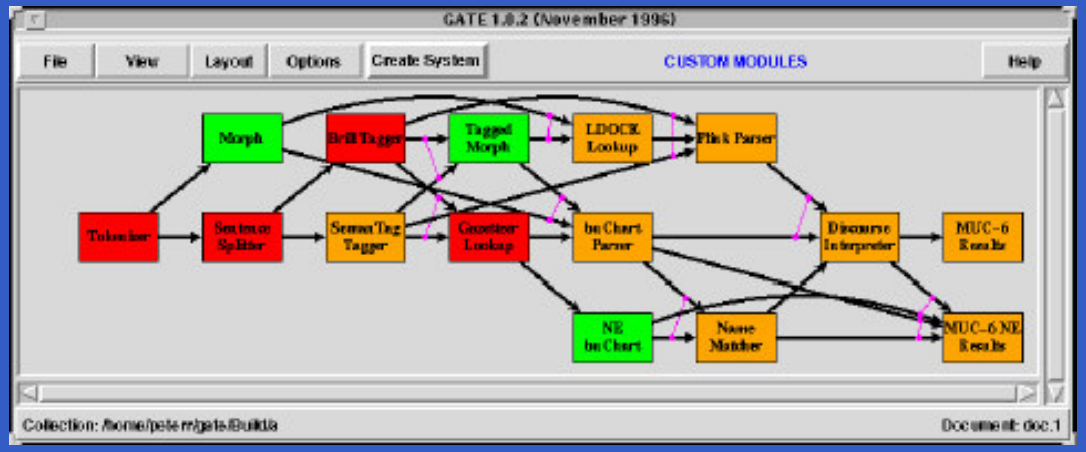
```
set creole_config(buchart) {
  title {buChart Parser}
  pre_conditions {
    document_attributes {language_english}
    annotations {token sentence morph lookup}
  }
  post_conditions {
    document_attributes {language_english}
    annotations {name syntax semantics}
  }
  viewers {
    {name single_span}
    {syntax tree}
    {semantics raw}
  }
}
```

GGI



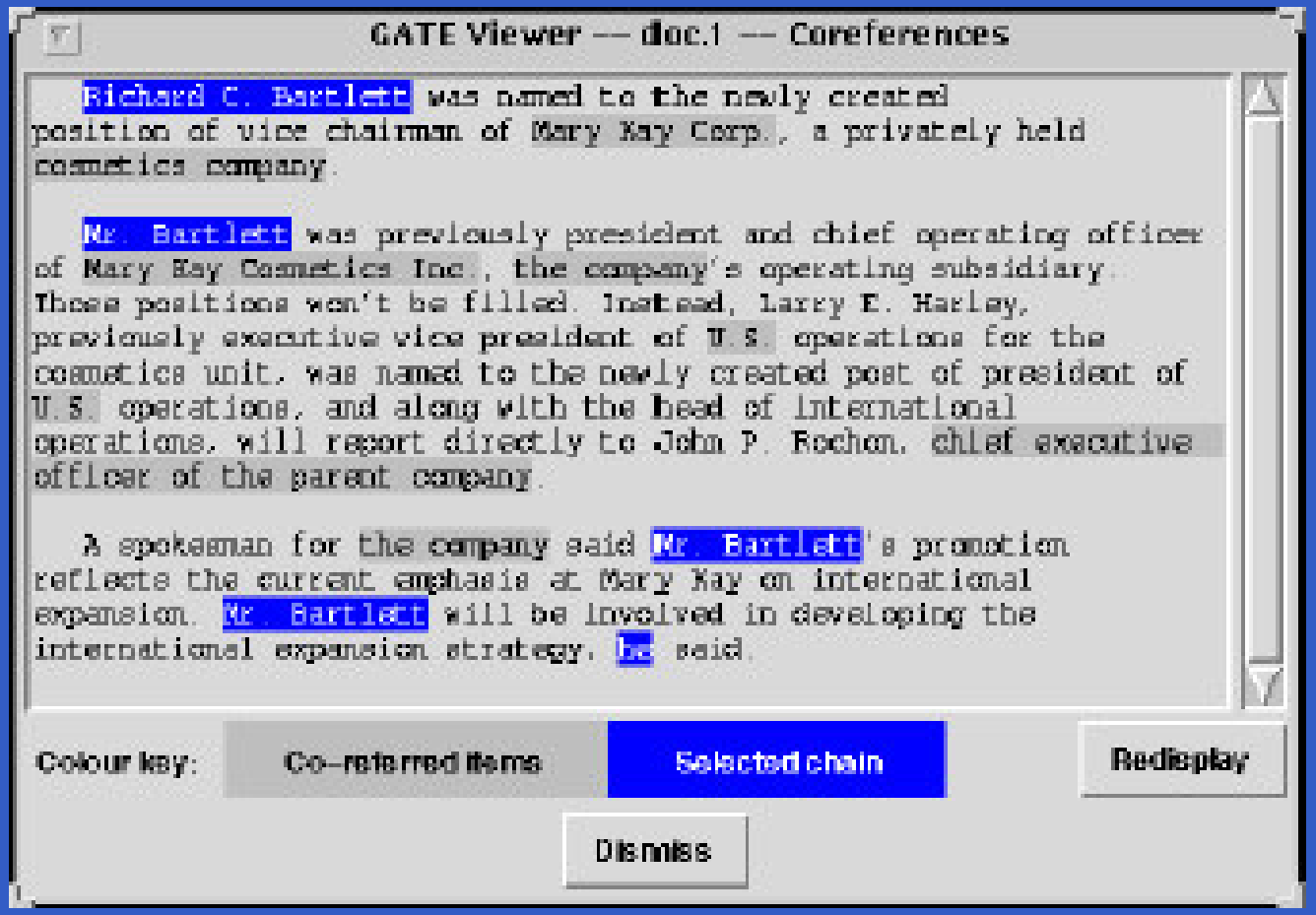
Implementation in Tcl/Tk: relativ plattformübergreifend

GGI – Ausführung von Komponenten

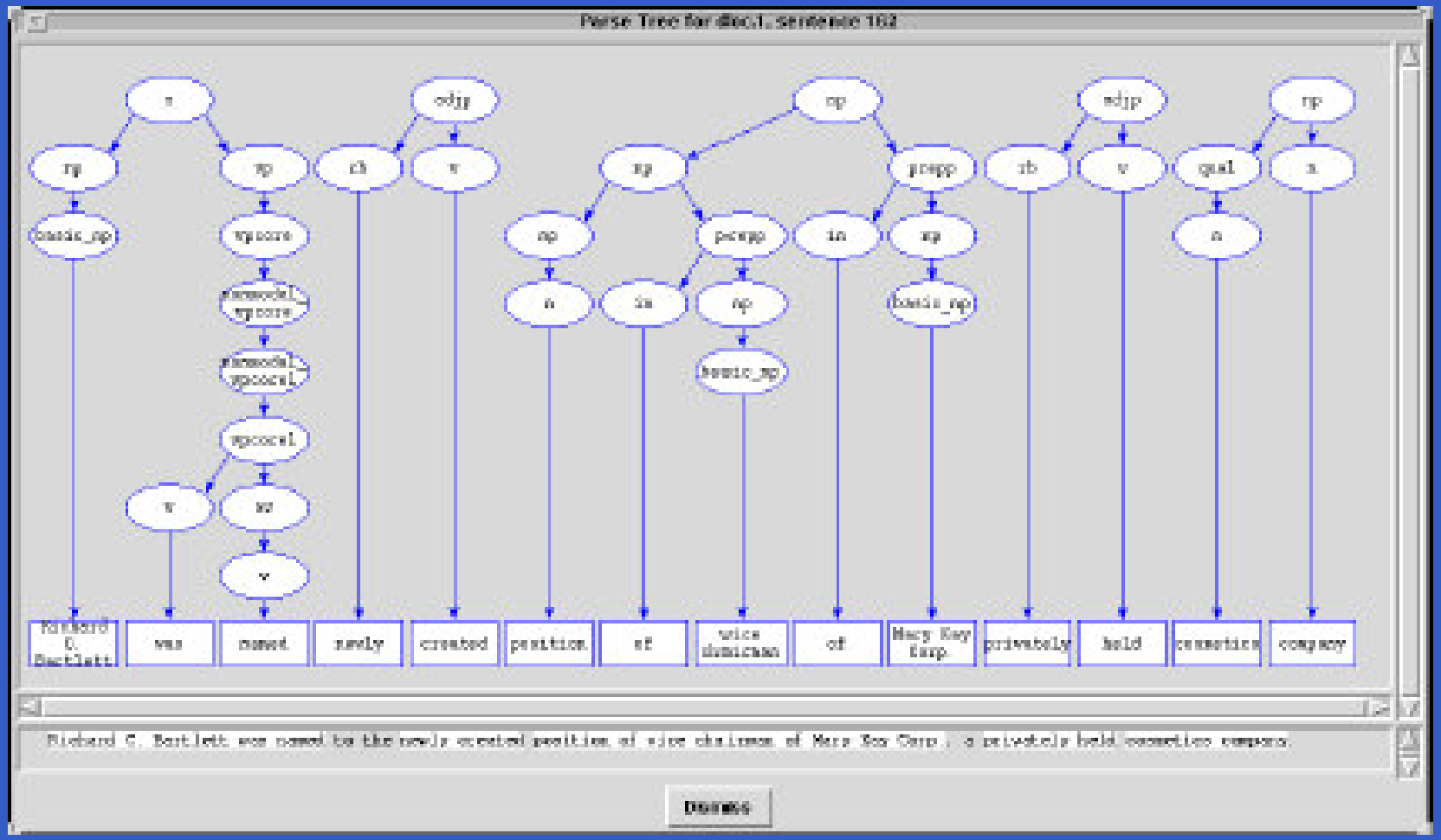


- Graph wird nach Informationen aus den Metadaten der Module erstellt
- momentaner Stand der Bearbeitung wird durch Farben dargestellt
- GATE1 unterstützt keine verteilte oder parallele Verarbeitung

GGI – Visualisierung – Annotation Vie



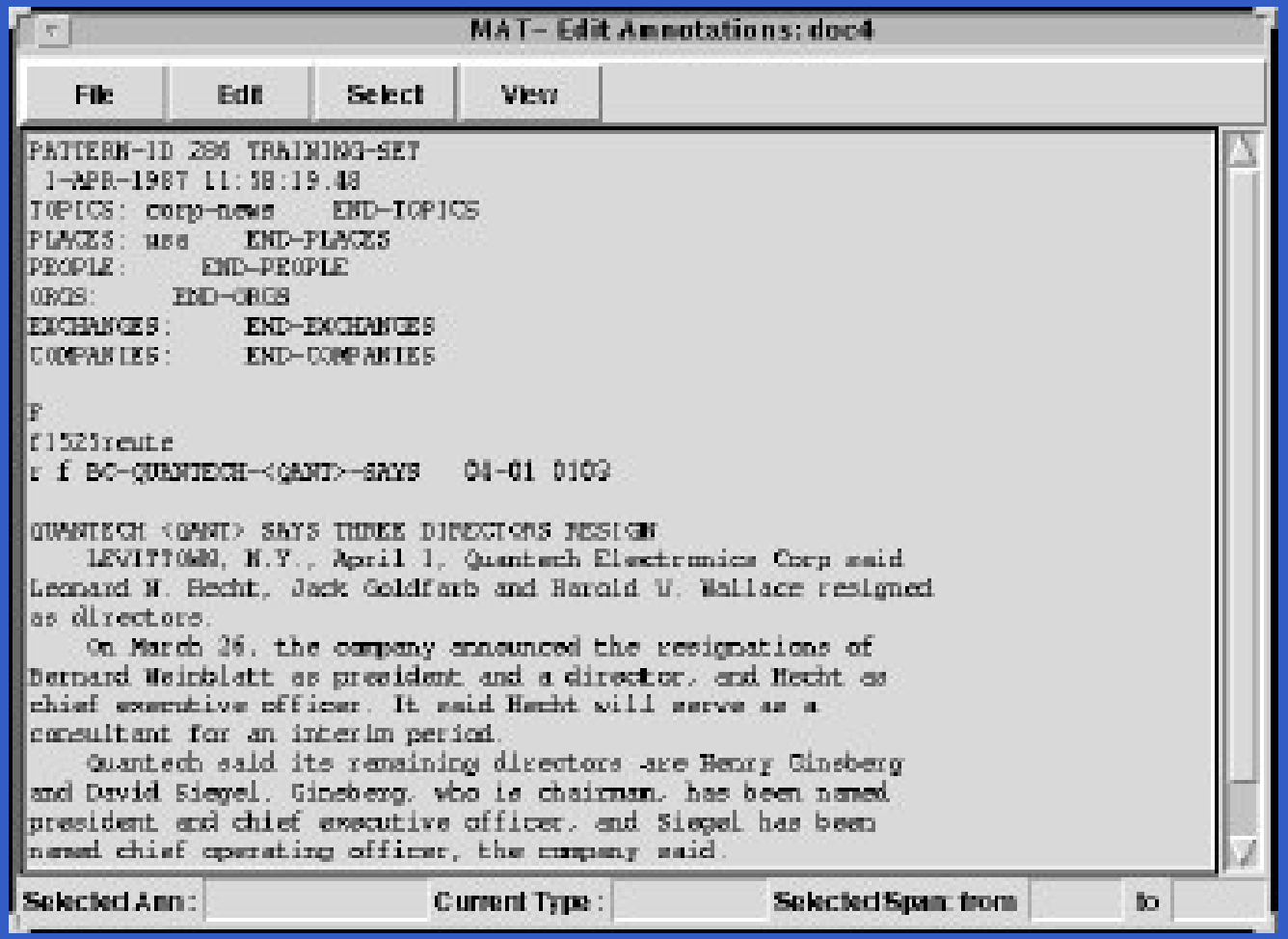
GGI – Visualisierung – Tree View



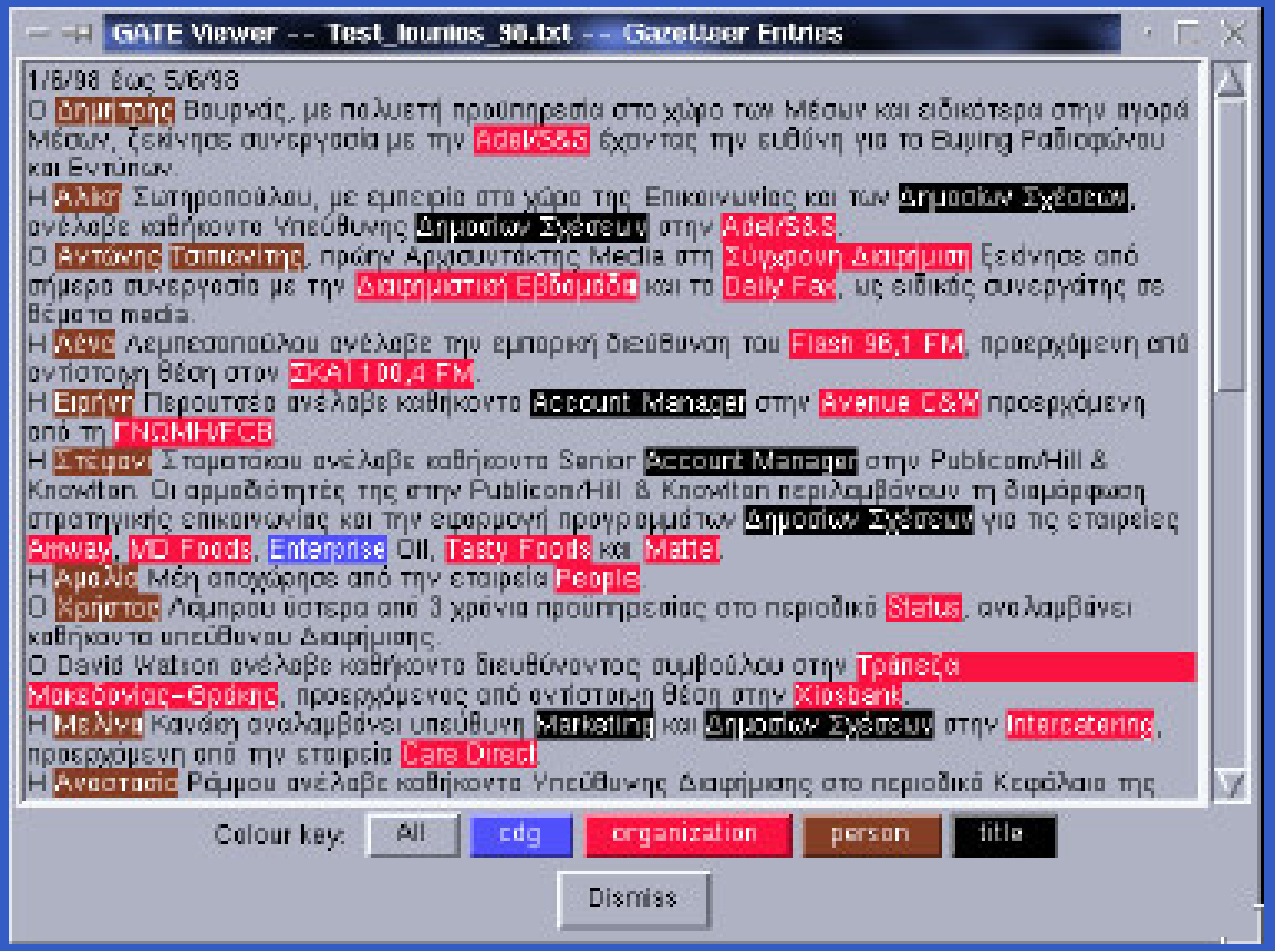
GGI – Vergleich von Daten

Ermöglicht den Vergleich von unterschiedlichen Arten der Bearbeitung

GGI – Eingabe von Daten



GGI – Internationalisierung



bisher beschränkt auf 8-Bit Zeichensätze

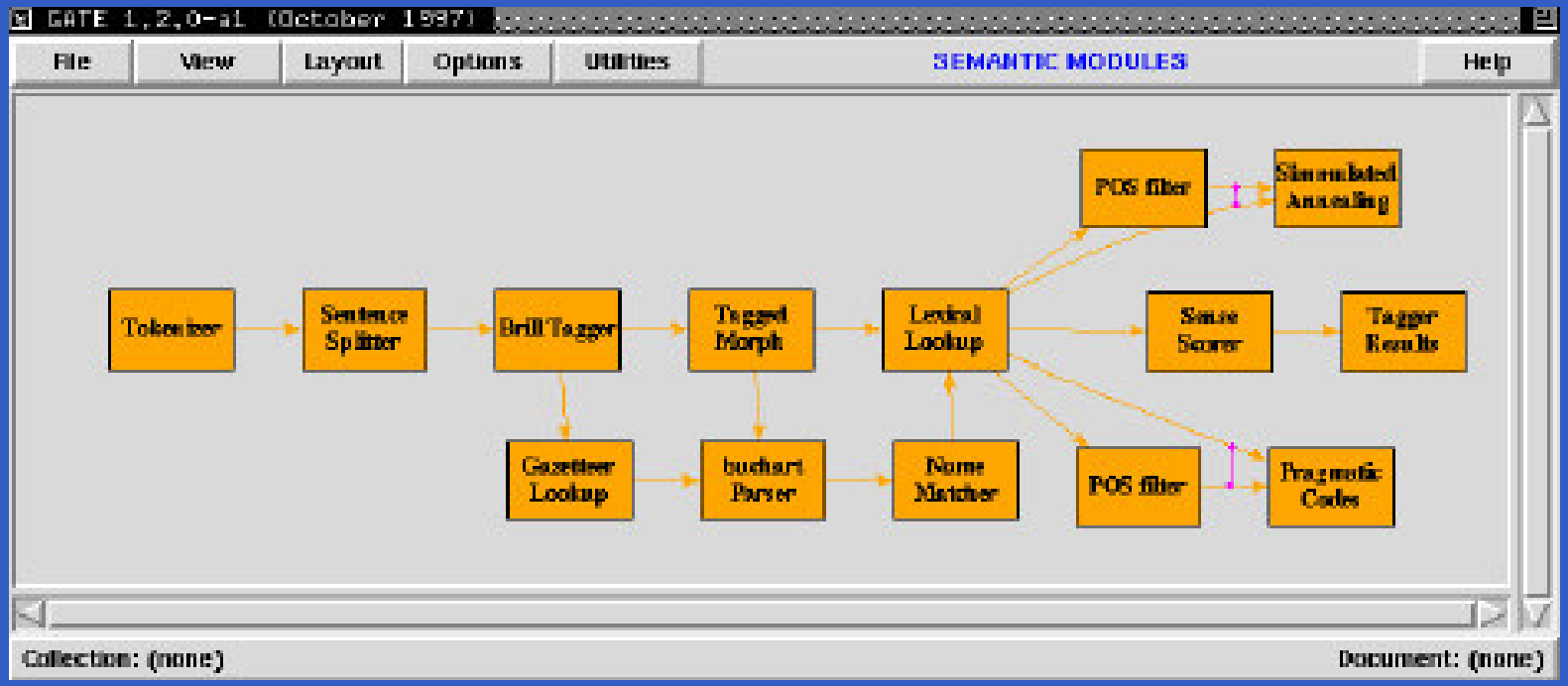
zwei Beispiele für den Einsatz von GATE

Case Study 1 - Sense Tagging

- “Sense Tagging“
- Es besteht aus 11 CREOLE Modulen

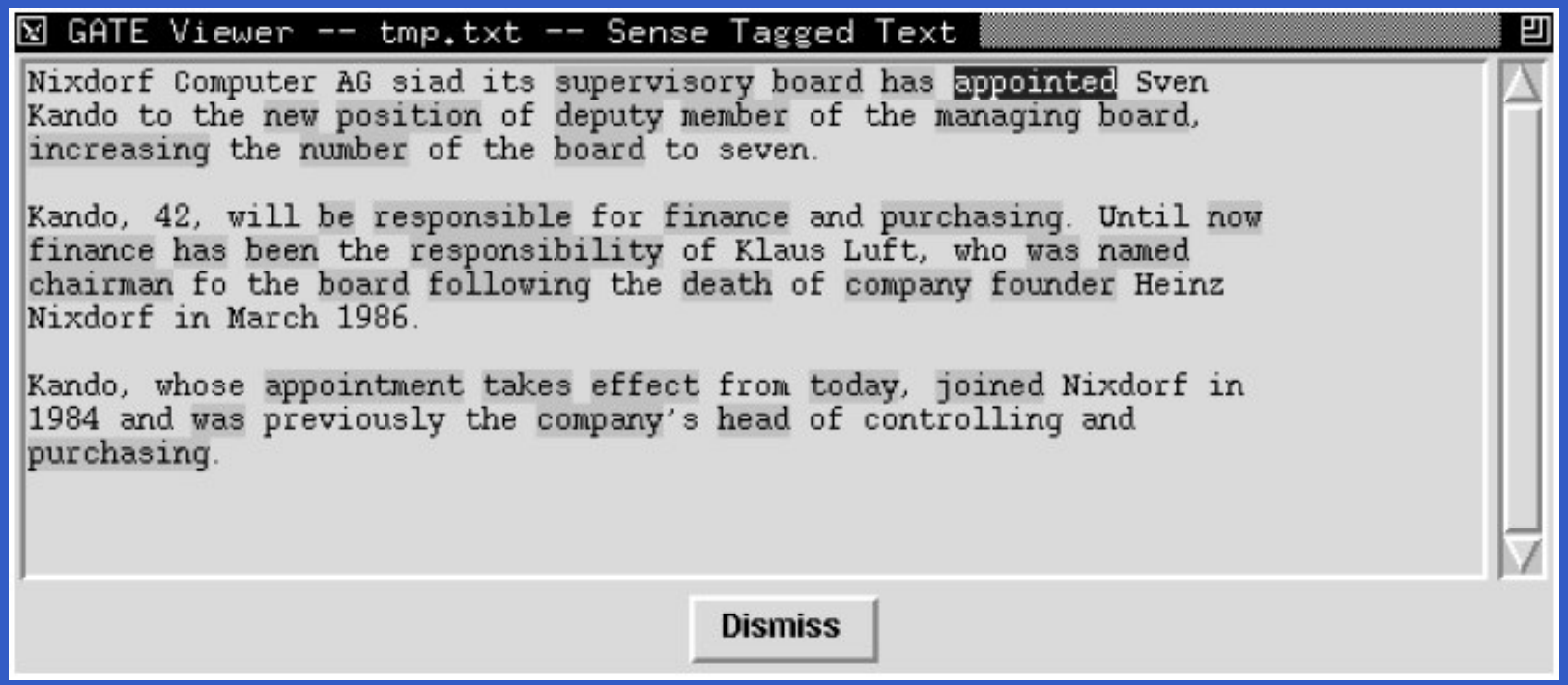
Case Study 1 - Sense Tagging (II)

GATE - Graphische Repräsentation vom “sense tagger“



Case Study 1 - Sense Tagging (III)

Möglichkeit der Graphischen Darstellung nach dem Durchlaufen des taggers



Lessons Learned for GATE

- Ermutigende Implementierung eines neuen Systems
- Unterstützung eines wohldefinierten Standards für Datentransfer zwischen den Komponenten
- Bereits vorhandene Komponenten werden genutzt

Lessons Learned for GATE (II)

- Weiterer Geschwindigkeitsgewinn durch “Ergebnis–Manager“
- Möglichkeit, Komponenten ein- und auszuschließen
- Problem von nichtmonotonen Datenbank–Updates
- geplante Lösung: Änderung der Reset-Funktion

Graphen

Graphen leisten 2 Haupt-Beiträge zum System

- gaphische Repräsentation des Kontrollflusses
- Möglichkeit, Ausführung von Komponenten zu manipulieren

CASE Study 2 - Lasie

- existierende “Language Enginieering“ Systeme bei GATE implementieren
- Die Leichtigkeit LaSIE zu implementieren
- LaSIE ist ein typisches System für existierende “LE’s“
  - Module nicht auf Wiederverwendbarkeit ausgelegt
  - Viele verschiedene Programmiersprachen verwendet

CREOLEisation

- Die größte Arbeit bei der Konvertierung war es
  - Verbindungen zwischen den Modulen trennen
  - neue Hüllen für die CREOLEisation schaffen
  - Verbindungen wieder neu herstellen
- 10 CREOLE-Module in GATE
- Gleiche Funktionalität bei besserer Bedienbarkeit

CREOLEisation (II)

Es gibt 3 große Barrieren bei der Implementation weiterer “LE“s:

- Die Bewerkstelligung der Speicherung und des Austausches von Informationen
- Die Inkompatibilität bei der Darstellung von Informationen über Texte
- Die Inkompatibilität der verschiedenen Typen von Information

Ausblick – GATE2, GATE3

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

- maschinelle Übersetzung

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

- maschinelle Übersetzung
- Informationsabfrage

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

- maschinelle Übersetzung
- Informationsabfrage
- Automatische Spracherkennung

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

- maschinelle Übersetzung
- Informationsabfrage
- Automatische Spracherkennung
- Dialogverarbeitung

GATE2

GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

- maschinelle Übersetzung
- Informationsabfrage
- Automatische Spracherkennung
- Dialogverarbeitung
- Einbeziehung wohlbekannter Algorithmen und Methoden

GATE2

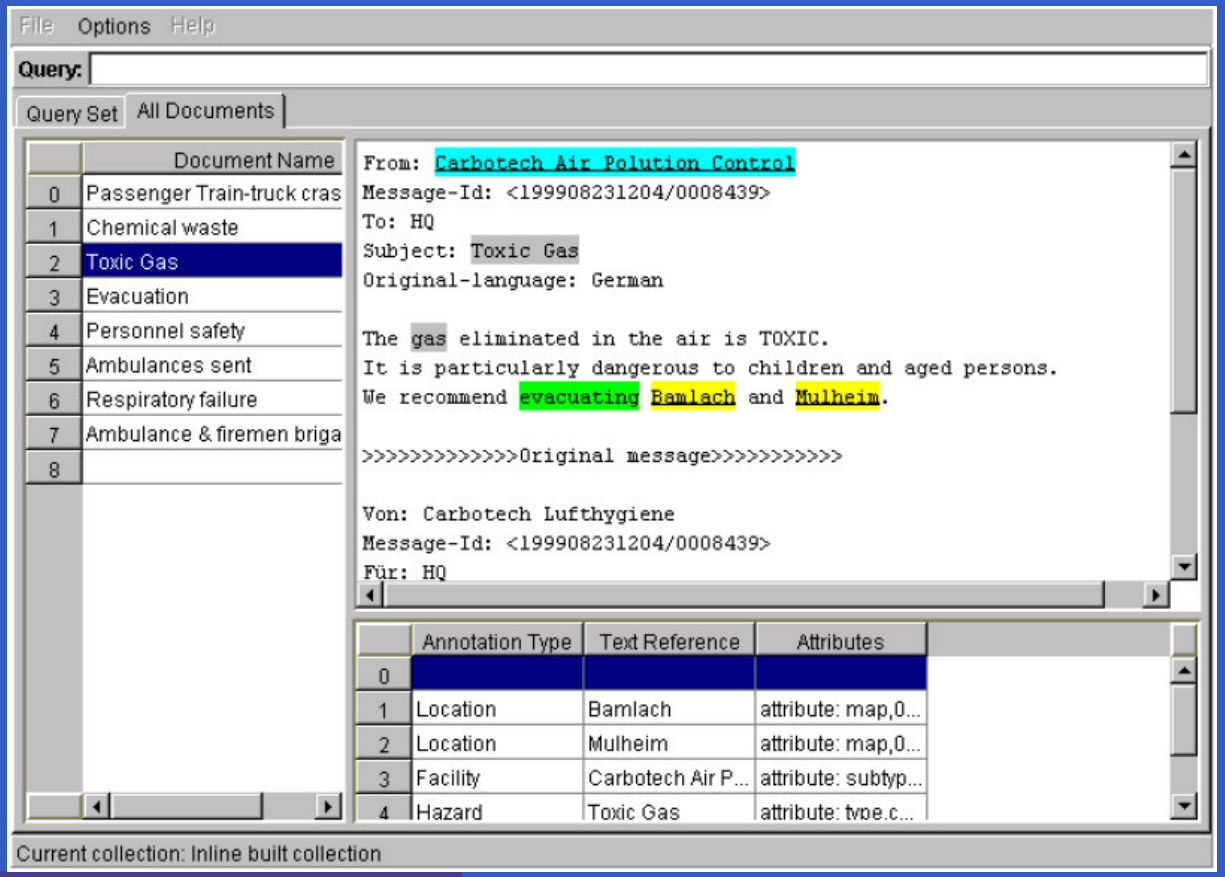
GATE1 hatte Schwerpunkt auf Informationsgewinnung

GATE2 erweitert die Nutzung auf:

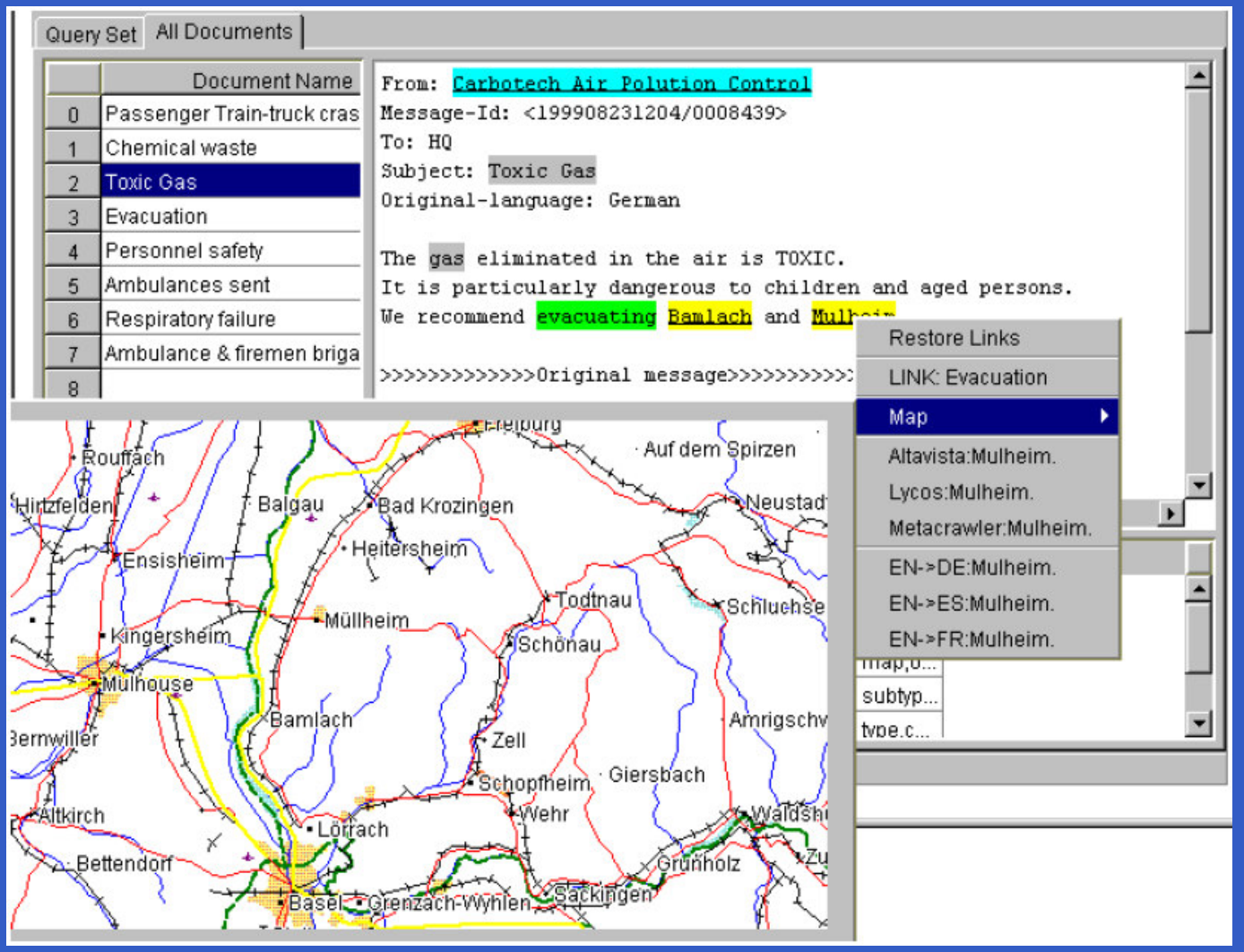
- maschinelle Übersetzung
- Informationsabfrage
- Automatische Spracherkennung
- Dialogverarbeitung
- Einbeziehung wohlbekannter Algorithmen und Methoden
- Sprachressourcen wie Lexika, Ontologien und Thesauri (als relationale Datenbanken)

LOTTIE

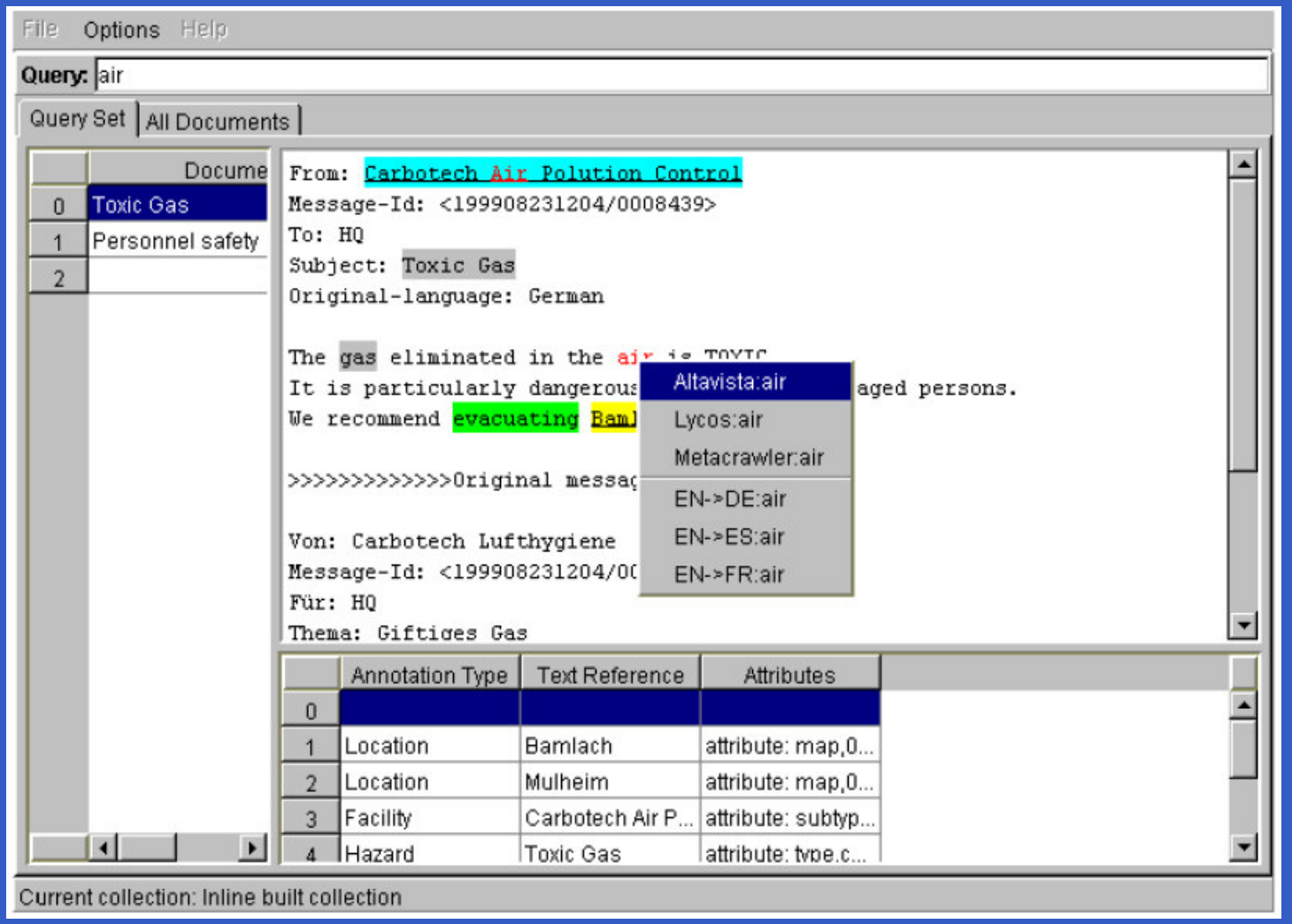
Wenig Overhead produzierende Text-Selektion mit Informationsgewinnung



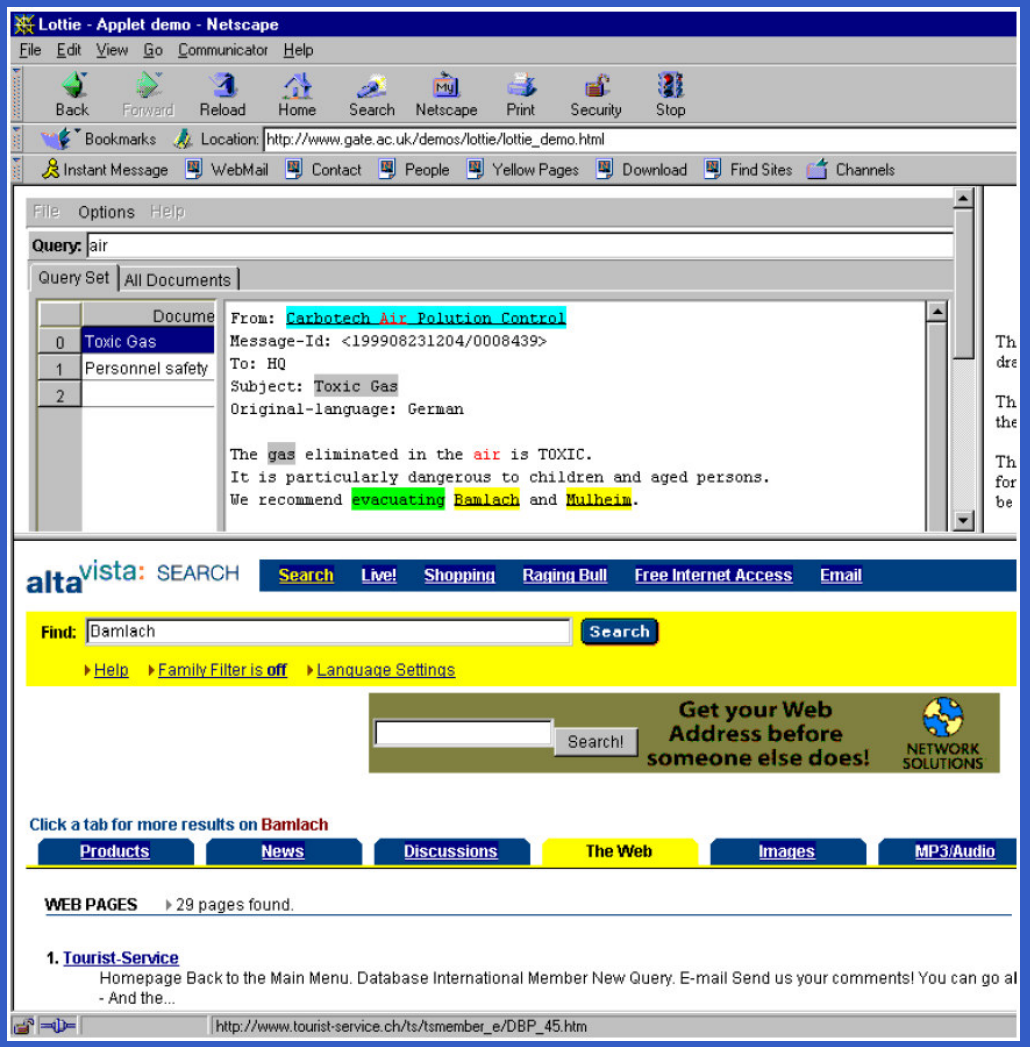
LOTTIE (II)


Query Set
All Documents
Document Name
0 Passenger Train-truck cras
1 Chemical waste
2 Toxic Gas
3 Evacuation
4 Personnel safety
5 Ambulances sent
6 Respiratory failure
7 Ambulance & firemen briga
8
From: Carbotech Air Polution Control
Message-Id: <199908231204/0008439>
To: HQ
Subject: Toxic Gas
Original-language: German
The gas eliminated in the air is TOXIC.
It is particularly dangerous to children and aged persons.
We recommend evacuating Bamlach and Mulh
>>>>>>>>>>>>>Original message>>>>>>>>>>
Restore Links
LINK: Evacuation
Map
Altavista:Mulheim.
Lycos:Mulheim.
Metacrawler:Mulheim.
EN->DE:Mulheim.
EN->ES:Mulheim.
EN->FR:Mulheim.
subtyp...
Rouffach
Hirtzfelden
Ensisheim
Kingersheim
Mulhouse
Bernwiller
Altkirch
Bettendorf
Balgau
Bad Krozingen
Heitersheim
Müllheim
Bamlach
Auf dem Spirzen
Neustad
Todtnau
Schluchse
Schönau
Zell
Amrigschv
Giersbach
Schopfheim
Wehr
Lörrach
Grunholz
Basel
Grenzach-Wyhlen
Säckingen

LOTTIE (III)


File Options Help
Query: air
Query Set
All Documents
Docume
0 Toxic Gas
1 Personnel safety
2
From: Carbotech Air Polution Control
Message-Id: <199908231204/0008439>
To: HQ
Subject: Toxic Gas
Original-language: German
The gas eliminated in the air is TOXIC
It is particularly dangerous
aged persons.
We recommend evacuating Baml
>>>>>>>>>>>>>Original messa
Von: Carbotech Lufthygiene
Message-Id: <199908231204/0
Für: HQ
Thema: Giftiges Gas
Altavista:air
Lycos:air
Metacrawler:air
EN->DE:air
EN->ES:air
EN->FR:air
Annotation Type
Text Reference
Attributes
1 Location Bamlach attribute: map,0...
2 Location Mulheim attribute: map,0...
3 Facility Carbotech Air P... attribute: subtyp...
4 Hazard Toxic Gas attribute: type,c...
Current collection: Inline built collection

LOTTIE (IV)


Lottie - Applet demo - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.gate.ac.uk/demos/lottie/lottie_demo.html
Instant Message WebMail Contact People Yellow Pages Download Find Sites Channels
File Options Help
Query: air
Query Set All Documents
Docume
0 Toxic Gas
1 Personnel safety
2
From: Carbotech Air Polution Control
Message-Id: <199908231204/0008439>
To: HQ
Subject: Toxic Gas
Original-language: German
The gas eliminated in the air is TOXIC.
It is particularly dangerous to children and aged persons.
We recommend evacuating Bamlach and Mulheim.
altavista: SEARCH
Search Live! Shopping Raging Bull Free Internet Access Email
Find: Bamlach
Search
Help Family Filter is off Language Settings
Search!
Get your Web Address before someone else does!
NETWORK SOLUTIONS
Click a tab for more results on Bamlach
Products News Discussions The Web Images MP3/Audio
WEB PAGES 29 pages found.
1. Tourist-Service
Homepage Back to the Main Menu. Database International Member New Query. E-mail Send us your comments! You can go al
- And the...
http://www.tourist-service.ch/ts/tsmember_e/DBP_45.htm

Quellen

Der Vortrag orientiert sich an:

Hamisch Cunningham, Software Architecture for Language Engineering

(vgl. `http://www.dcs.shef.ac.uk/~hamish/`)

Weitere Informationen zu GATE:

`http://gate.ac.uk/`