Information Extraction with GATE
General Architecture for Text Engineering

Some experiences developing a very small medical information extraction prototype

Johann Mitloehner
Vienna University of Economics, Austria
http://mitloehner.net

Some GATE Predefined Components

ANNIE - *A Nearly-New Information Extraction*

- tokenizer
- gazetteer
- sentence splitter
- part of speech tagger
- named entities transducer
- co-reference tagger

Other Components

Ontology viewer and editor

- accepts OWL ontologies in XML and N3

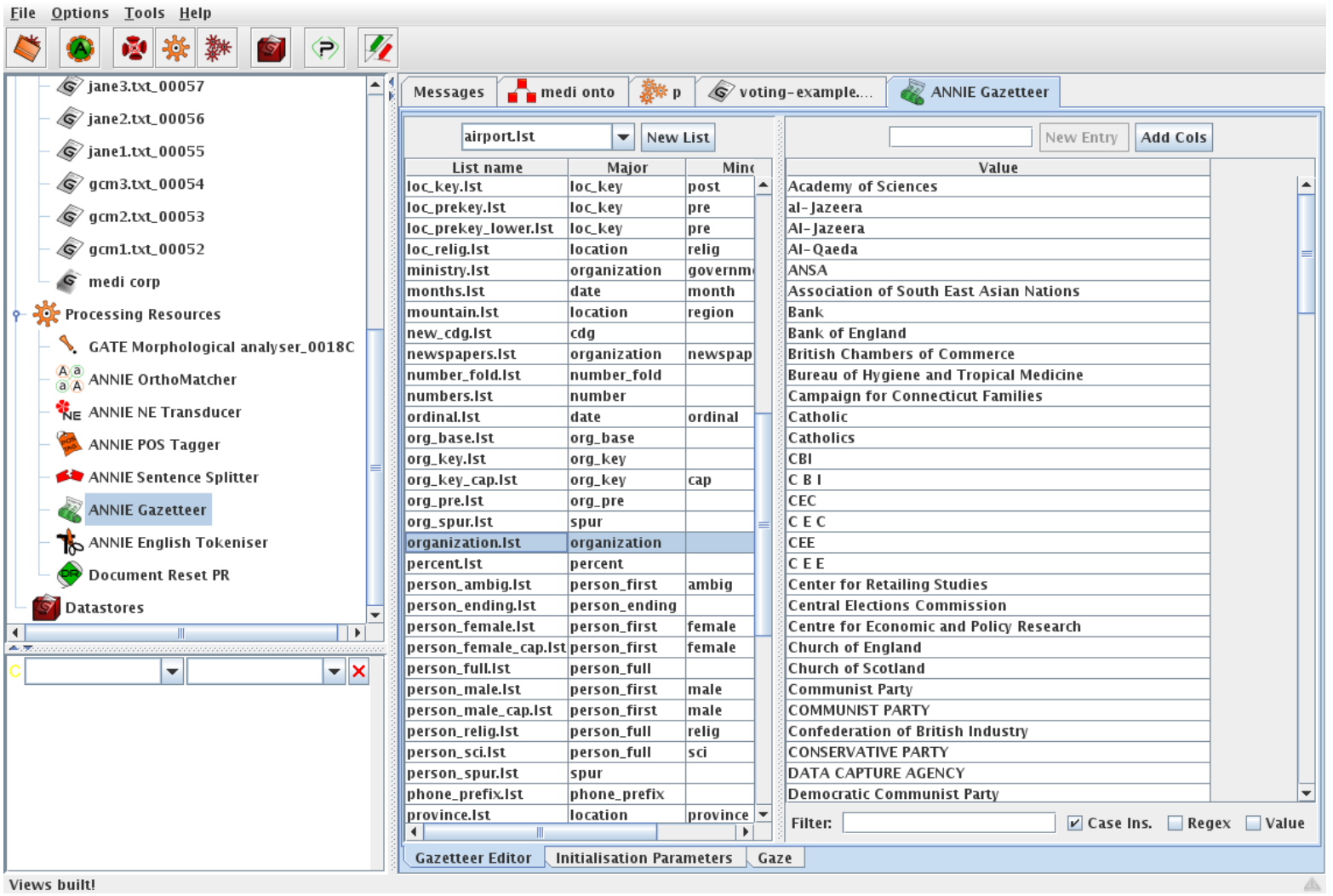
JAPE Java Annotation Patterns Engine

- regular expressions over annotations
- grammar files and rules, including Java code
- ontology-aware

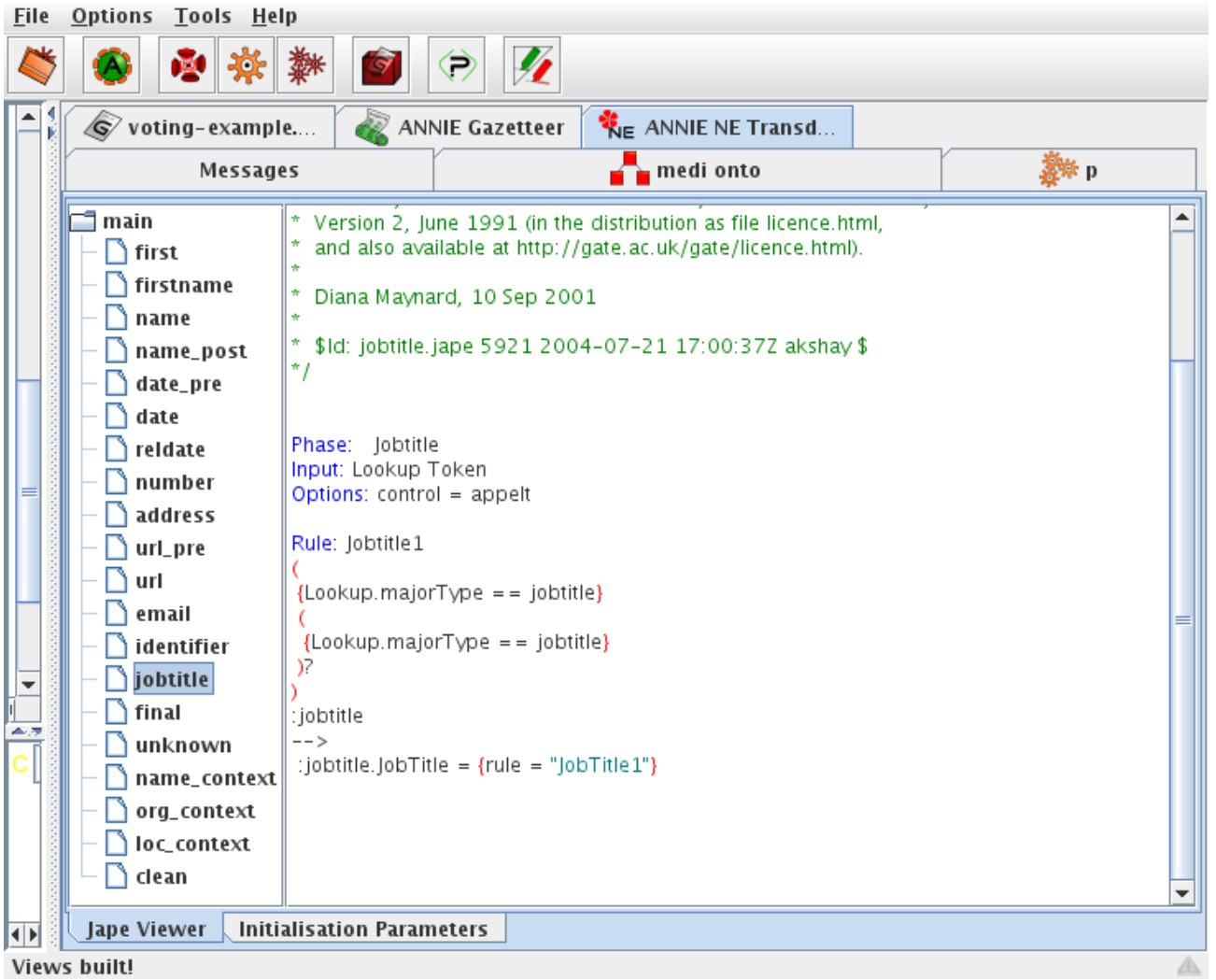
GUI Interface

- Easy to construct processing pipeline with predefined components
- Easy to view input documents and results, including statistics
- Cumbersome to automate and include customized processing resources

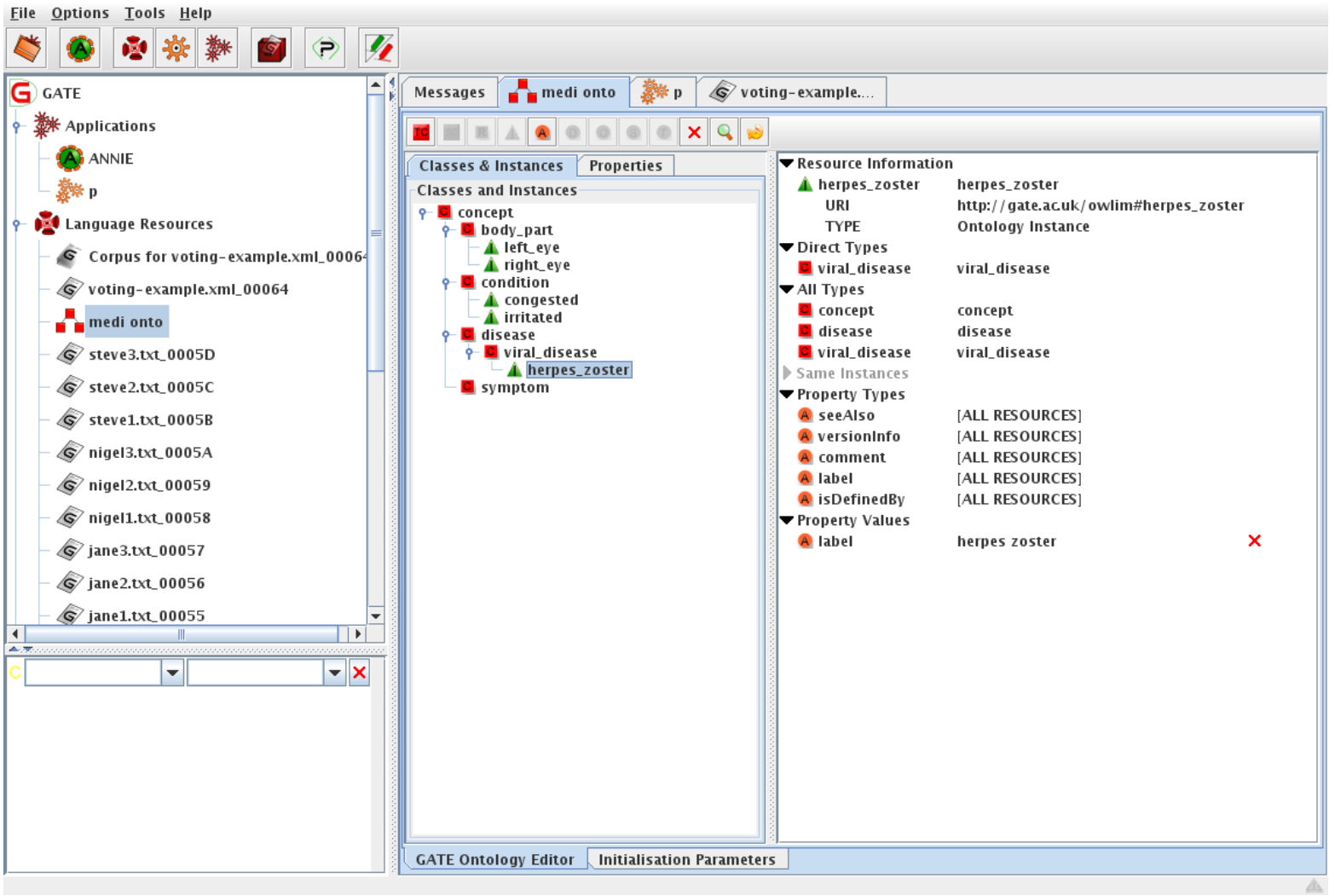
Gazetteer Editor



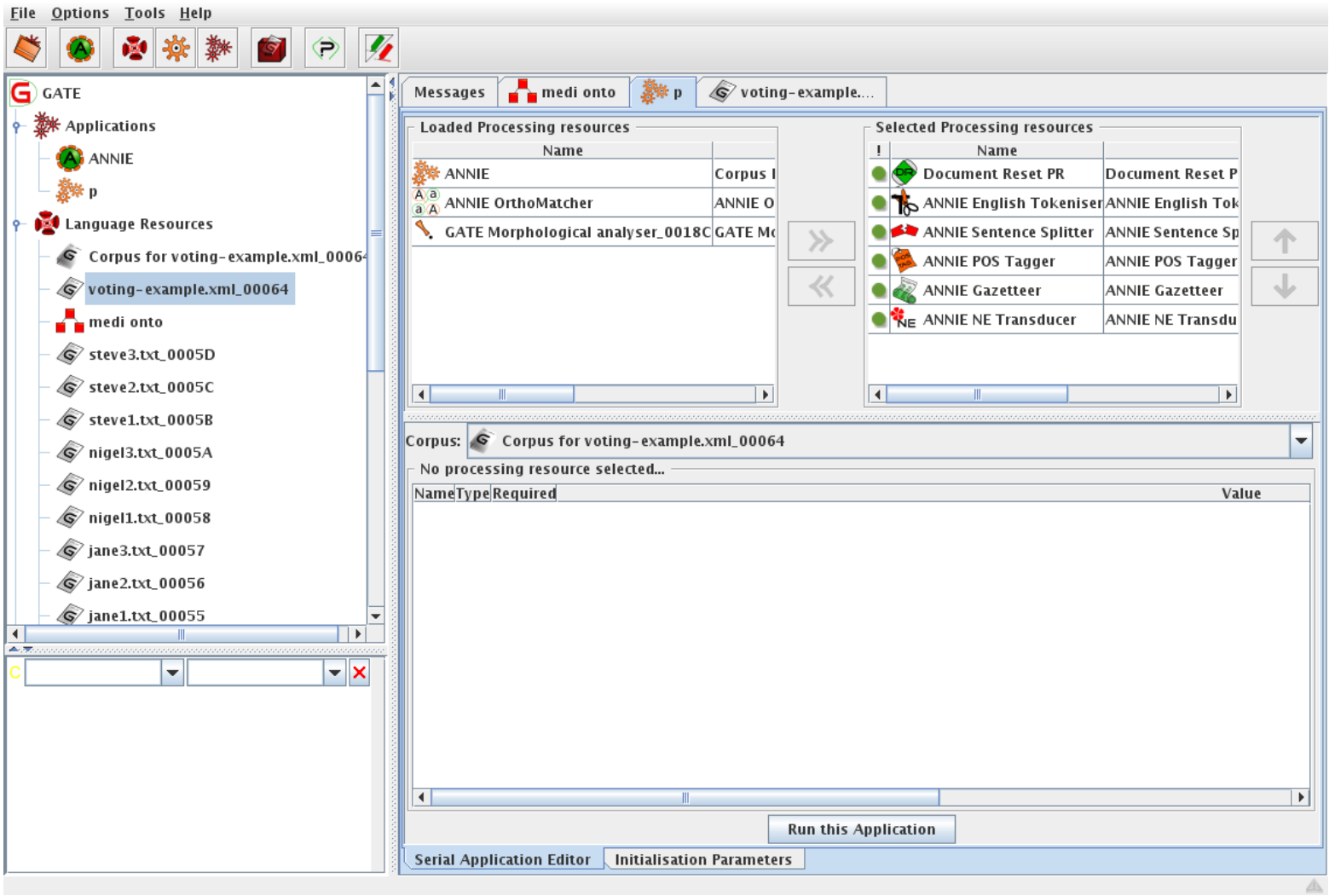
NE Transducer



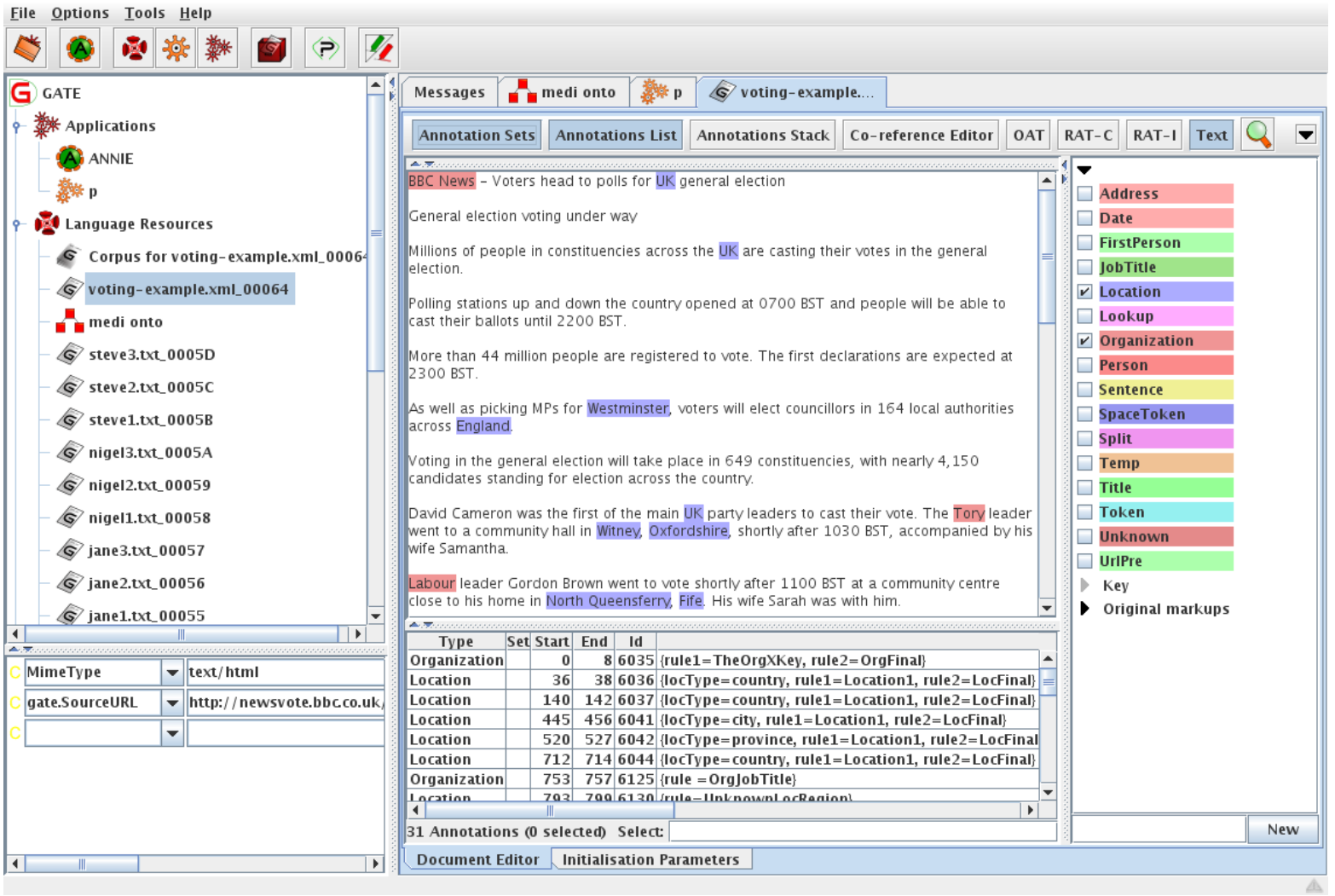
Ontology Editor



Processing Pipeline



Document view



GATE Embedded

Use GATE libs in Java applications

- easy, just set CLASSPATH
- BUT lib component names do not correspond to GUI names
--> Best to proceed from sample code

00071905@CSSE2702:~/medical:543> cat makefile
GATE=$(HOME)/GATE-6.1

Pipe.class:	Pipe.java
	javac -classpath $(GATE)/bin/gate.jar:$(GATE)/lib/* Pipe.java

```
import gate.*;
import gate.util.*;
import gate.creole.*;
import gate.creole.ontology.*;
import gate.creole.gazetteer.*;
import java.io.*;
import java.util.*;
import java.net.*;

public class Pipe {
  private gate.Corpus corpus;
  public static void main(String[] args) throws Exception {
    new Pipe(args);
  }

  Pipe(String[] files) throws Exception {
    Gate.init();
    Gate.getCreoleRegister().registerDirectories(
            new File(System.getProperty("user.dir")).toURL());
    System.out.println("\n== OBTAINING DOCUMENTS ==");
    createCorpus(files);
    System.out.println("\n== USING GATE TO PROCESS THE DOCUMENTS ==");
    runProcessingResources();
    System.out.println("\n== DOCUMENT FEATURES ==");
    display();
  }
```

```
// Create corpus from documents on command line

private void createCorpus(String[] files) throws GateException {
    corpus = Factory.newCorpus("Transient Gate Corpus");

    for (int file = 0; file < files.length; file++) {
      System.out.print("\t " + (file + 1) + ") " + files[file]);
      try {
        corpus.add(Factory.newDocument(new File(files[file]).toURL()));
        System.out.println(" -- success");
      } catch(gate.creole.ResourceInstantiationException e) {
        System.out.println(" -- failed (" + e.getMessage() + ")");
      } catch(Exception e) {
        System.out.println(" -- " + e.getMessage());
      }
    }
  }
```

```
// construct the processing pipeline and run it on the corpus

private void runProcessingResources() throws Exception {
  SerialAnalyserController pipeline = (SerialAnalyserController)Factory
    .createResource("gate.creole.SerialAnalyserController");

  // add processing resources to the pipeline: start with tokenizer

  ProcessingResource tokeniser = (ProcessingResource)
    Factory.createResource("gate.creole.tokeniser.DefaultTokeniser",
      Factory.newFeatureMap());

  pipeline.add(tokeniser);
```

```
// gazetteer needs parameters, have to construct feature map first

FeatureMap feats = Factory.newFeatureMap();
FeatureMap params = Factory.newFeatureMap();
URL mappingURL = new URL("file:" + System.getProperty("user.home") +
 "/medical/gaz/mappings.def");
URL listsURL = new URL("file:" + System.getProperty("user.home") +
 "/medical/gaz/lists.def");
params.put("mappingURL", mappingURL);
params.put("listsURL", listsURL);
params.put("gazetteerName", "com.ontotext.gate.gazetteer.HashGazetteer");

OntoGazetteer ontoGazetteer = (OntoGazetteer) Factory.createResource(
  "gate.creole.gazetteer.OntoGazetteerImpl", params, feats, "OntoGazetteer");

pipeline.add(ontoGazetteer);
```

```
    // ontology and jape transducer

    FeatureMap fm = Factory.newFeatureMap();
    fm.put("n3URL", new URL("file:" + System.getProperty("user.home") +
     "/medical/medi.n3"));
    fm.put("defaultNameSpace", "");
    Ontology ontology = (Ontology) Factory.createResource(
      "gate.creole.ontology.owlim.OWLIMOntologyLR", fm, null, "medi");

    fm = Factory.newFeatureMap();
    fm.put("encoding","ISO-8859-1");
    fm.put("grammarURL", new URL("file:" + System.getProperty("user.home") +
     "/medical/medi.jape"));
    fm.put("ontology", ontology);
    ProcessingResource tr = (ProcessingResource)
      Factory.createResource("gate.creole.Transducer", fm);

    pipeline.add(tr);

    // all done lets rock

    pipeline.setCorpus(corpus);
    pipeline.execute();
}
```

```
// display specific annotations only

private void display() throws Exception {
  Iterator docs = corpus.iterator();
  while (docs.hasNext()) {
    Document doc = (Document) docs.next();
    feature("Symptom", doc);
    feature("Disease", doc);
  }
}

// need to use document and offsets to get the text inside the annotation

void feature(String feat, Document doc) throws Exception {
  Iterator it = doc.getAnnotations().get(feat).iterator();
  while (it.hasNext()) {
    System.out.println(feat + ": " + doc.getSourceUrl());
    Annotation a = (Annotation) it.next();
    System.out.println(doc.getContent().getContent(a.getStartNode().getOffset(),
   a.getEndNode().getOffset()));
    System.out.println();
  }
}
```

00071905@CSSE2702:~/medical:530> cat gaz/lists.def

bodyparts.lst:bodypart:precise
conditions.lst:condition:imprecise
viral_diseases.lst:viral_disease:unreliable

col 2 is majorType, col 3 is minorType (feature names for annotation)

00071905@CSSE2702:~/medical:531> cat gaz/bodyparts.lst

eyes
limbs
left eye
right eye

--> “left eye” gets a “Lookup” annotation with feature majorType=bodypart

00071905@CSSE2702:~/medical:532> cat gaz/mappings.def

bodyparts.lst::bodypart
conditions.lst::condition
viral_diseases.lst::viral_disease

mappings connect gazetteer list names to ontology class names
note that there is no “diseases” list

```
00071905@CSSE2702:~/medical:535> cat medi.n3
@prefix : <http://gate.ac.uk/owlim#> .

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix protons: <http://proton.semanticweb.org/2005/04/protons#> .
@prefix protont: <http://proton.semanticweb.org/2005/04/protont#> .

:concept a owl:Class .
:symptom a owl:Class .
:body_part a owl:Class .
:condition a owl:Class .
:disease a owl:Class .
:condition rdfs:subClassOf :concept .
:symptom rdfs:subClassOf :concept .
:disease rdfs:subClassOf :concept .
:body_part rdfs:subClassOf :concept .
:viral_disease a owl:Class .
:viral_disease rdfs:subClassOf :disease .

:left_eye a :body_part .
:right_eye a :body_part .
:herpes_zoster a :viral_disease .
:congested a :condition .
:irritated a :condition .
```

```
00071905@CSSE2702:~/medical:537> cat medi.jape

Phase: OntoMatching
Input: Token Lookup
Options: control = appelt

# annotate body parts affected by conditions as “Symptom”

Rule: Symptom
(
  ({Lookup.majorType == bodypart}):part
  ({Token.string != ""})[0,3]
  ({Lookup.majorType == condition}):cond
):partcond
-->
 :partcond.Symptom = { majorType = medical, minorType = unreliable }

# annotate anything that belongs to the class “disease” according to ontology

Rule: Disease
  ({Lookup.class == disease}):dis
-->
  :dis.Disease = { majorType = medical, minorType = speculative }
```

00071905@CSSE2702:~/medical:539> cat gcm1.txt

RE: ######## ######### DOB: ##/##/###### Our Ref: 25886
###, Kojonup WA 6395 Tel: ####

Thank you for referring this lady who had a left sided herpes zoster involving the maxillary division of her trigeminal nerve.
This occurred last May, and she is really still recovering. At this point of time she still has post hepatic pain and has a partial seventh nerve palsy on the left side.
Her visual acuity uncorrected was right and left 6/7.5 slightly better in the left than the right.
The left eye was a little congested, and she had a slow blink rate with some degree of exposure keratitis present.
The intraocular pressures were normal, and dilated examination showed no fundus pathology. The optic discs were healthy.
I told Mrs ###### that whilst the virus had burnt out, she had residual symptoms which may go on for some time.
In the meantime I have pointed out the importance of preserving the cornea of the left eye and the need for frequent lubrication and artificial tears.

With kind regards,
Yours sincerely,

00071905@CSSE2702:~/medical:538> java Pipe *.txt

== OBTAINING DOCUMENTS ==
 1) gcm1.txt -- success
 2) gcm2.txt -- success
...

== USING GATE TO PROCESS THE DOCUMENTS ==
HashGazetteer is being initialized!
Loading OWLIMOntologyLR, doSetAutoLabel is true
ruleSet=owl-max, partialRdfs=false

== DOCUMENT FEATURES ==
Symptom: file:/home/res.labs/5/00071905/linux/medical/gcm1.txt
left eye was a little congested

Disease: file:/home/res.labs/5/00071905/linux/medical/gcm1.txt
herpes zoster

note that “herpes zoster” is annotated as “viral_disease” by the gazetteer
only via the ontology the class “disease” can be inferred

00071905@CSSE2702:~/medical:540> cat gaz/viral_diseases.lst
herpes zoster