Unstructured Information Processing with Apache UIMA

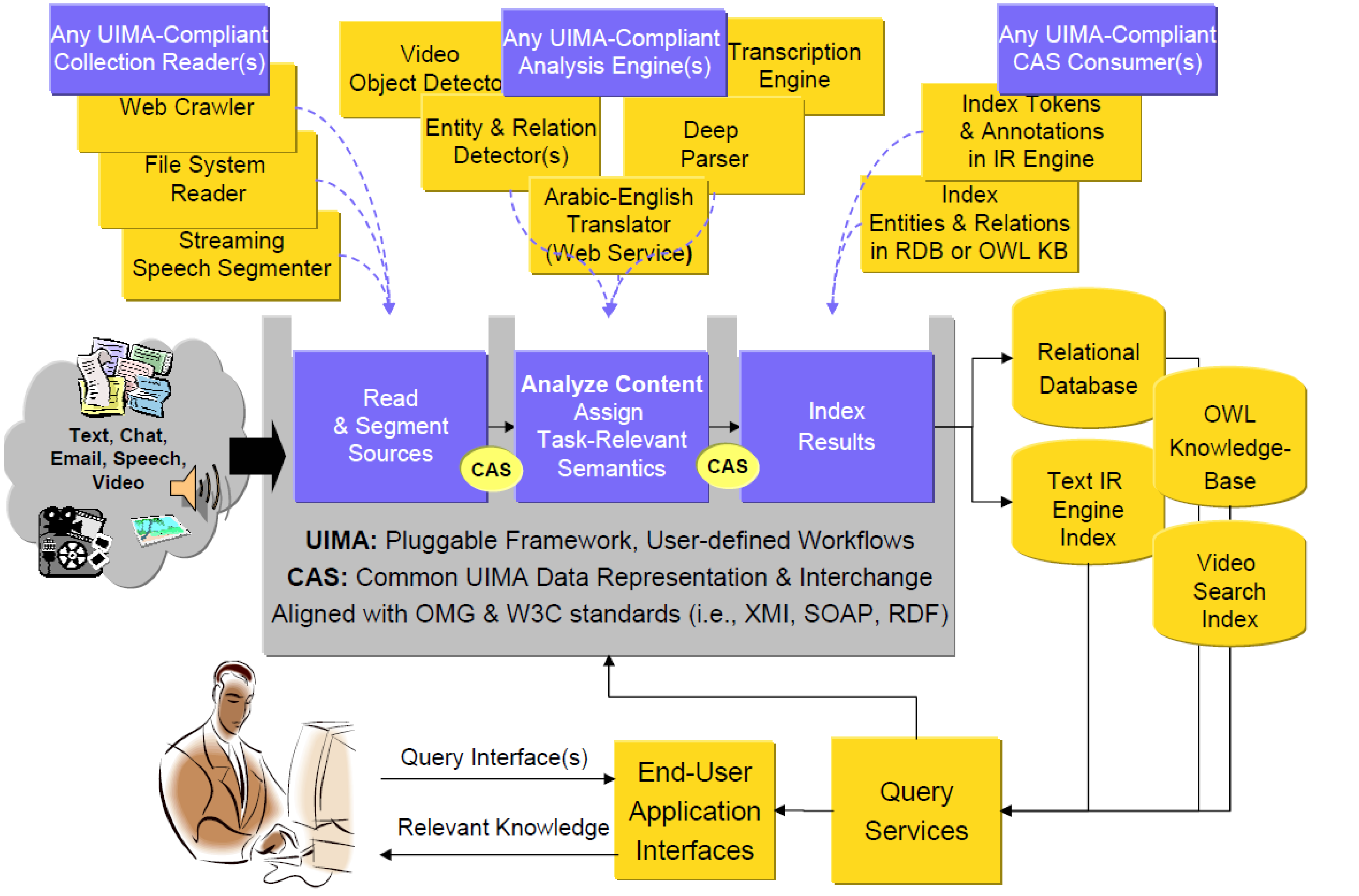
Computers Playing Jeopardy! Course

Stony Brook University

What is UIMA?

- UIMA is a framework, a means to integrate text or other unstructured information analytics
- Reference implementations available for Java, C++ and others
- An Open Source project under the umbrella of the Apache Foundation

  http://uima.apache.org



Any UIMA-Compliant Collection Reader(s)

Web Crawler

File System Reader

Streaming Speech Segmenter

Video Object Detecto

Any UIMA-Compliant Analysis Engine(s)

Transcription Engine

Entity & Relation Detector(s)

Deep Parser

Arabic-English Translator (Web Service)

Any UIMA-Compliant CAS Consumer(s)

Index Tokens & Annotations in IR Engine

Index Entities & Relations in RDB or OWL KB

Text, Chat, Email, Speech, Video

Read & Segment Sources

CAS

Analyze Content Assign Task-Relevant Semantics

CAS

Index Results

UIMA: Pluggable Framework, User-defined Workflows

CAS: Common UIMA Data Representation & Interchange

Aligned with OMG & W3C standards (i.e., XMI, SOAP, RDF)

Relational Database

OWL Knowledge-Base

Text IR Engine Index

Video Search Index

Query Interface(s)

Relevant Knowledge

End-User Application Interfaces

Query Services

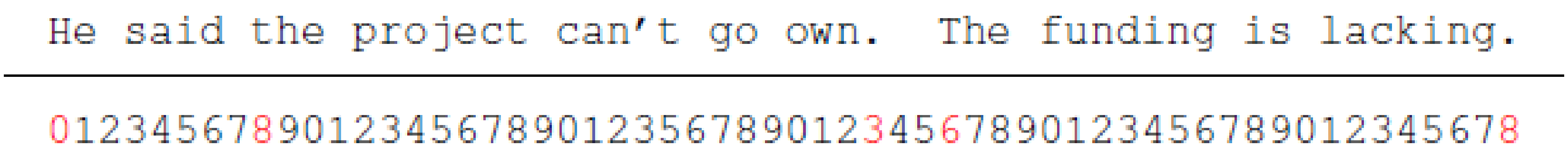
T-FaNT 07

Analytics Frameworks

- EXAMPLE: find all telephone numbers in running text
- Regular expression: (([0-9]{3})|[0-9]{3}) -? [0-9]{3} -? [0-9]{4}
  - How to feed this further processing?
  - How to query current knowledge and add information to knowledge?
  - Acquiring technology from external vendors, free software projects, etc?
  - How to mix technologies?
- Ad-hoc in-line annotations, e.g., modify text to include annotations: This/DET happy/ADJ puppy/N
  - Gets very messy very quickly: (S (NP (This/DET happy/ADJ puppy/N) (VP eats/V (NP (the/DET bone/N)))

Standoff Annotations

- Do not modify the text, BUT Keep the annotations as offsets within the original text
- UIMA is built with standoff annotations at its core.
- Example:



```
He said the project can't go own.  The funding is lacking.
0123456789012345678901235678901234567890123456789012345678
```

- Sentence Annotation: 0-33, 36-58.

Type Systems

- Key to integrating analytic packages developed by independent vendors.
- Clear metadata about
  - Expected Inputs
    - Tokens, sentences, proper names
  - Produced Outputs
    - Parse trees, focus
- The framework creates an unified typesystem for a given set of annotators being run.

UIMA Concepts

- Common Annotation Structure or CAS
  - Subject of Analysis (SofA or View)
  - JCas
- Feature Structures
  - Annotations
- Analysis Engines

UIMA tutorial

- http://uima.apache.org/downloads/releaseDocs/2.1.0-incubating/docs/html/tutorials_and_users_guides/tutorials_and_users_guides.html
- Analysis Engine that identifies room numbers in text:
- Example CS patterns: CompSci-1145, CS-1020, CS2030
  - Regular Expression Pattern:

  (CompSci | CS) -? (1|2) [0-9][0-9] [0-9]
- Steps:

  1 Define the CAS types that the annotator will use.

  2 Generate the Java classes for these types.

  3 Write the actual annotator Java code.

  4 Create the Analysis Engine descriptor.

  5 Test the annotator.

The XML descriptor

```xml
<?xml version="1.0" encoding="UTF-8" ?>
        <typeSystemDescription xmlns="http://uima.apache.org/resourceSpecifier">
                <name>TutorialTypeSystem</name>
                <description>Type System Definition for the tutorial examples -
                                as of Exercise 1</description>
                <vendor>Apache Software Foundation</vendor>
                <version>1.0</version>
                <types>
                        <typeDescription>
                                <name>org.apache.uima.tutorial.RoomNumber</name>
                                <description></description>
                                <supertypeName>uima.tcas.Annotation</supertypeName>
                                <features>
                                        <featureDescription>
                                                <name>building</name>
                                                <description>Building containing this room</description>
                                                <rangeTypeName>uima.cas.String</rangeTypeName>
                                        </featureDescription>
                                </features>
                        </typeDescription>
                </types>
        </typeSystemDescription>
```

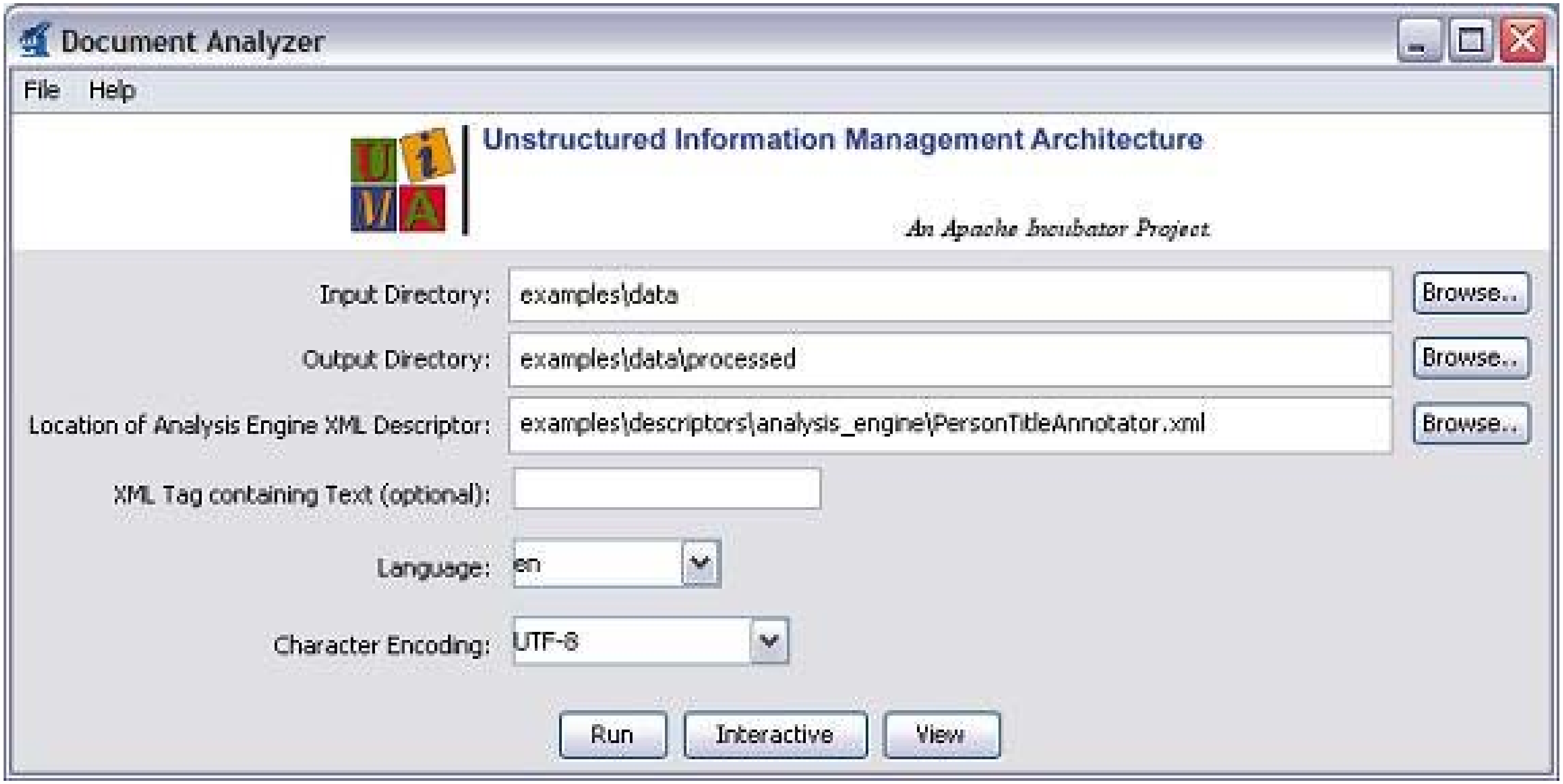
The Analysis Engine code

```
package org.apache.uima.tutorial.ex1;

import java.util.regex.Matcher;
import java.util.regex.Pattern;

import
  org.apache.uima.analysis_component.JCasAnnotator_ImplBase;
import org.apache.uima.jcas.JCas;
import org.apache.uima.tutorial.RoomNumber;

public class RoomNumberAnnotator extends
  JCasAnnotator_ImplBase {
        private Pattern myPattern =
             Pattern.compile("\\b[0-4]\\d-[0-2]\\d\\d\\b");
```

The Analysis Engine code

```
public void process(JCas aJCas) {
        // get document text
        String docText = aJCas.getDocumentText();
        // search for room numbers
        Matcher matcher = myPattern.matcher(docText);
        int pos = 0;
        while (matcher.find(pos)) {
                // found one - create annotation
                RoomNumber annotation = new RoomNumber(aJCas);
                annotation.setBegin(matcher.start());
                annotation.setEnd(matcher.end());
                annotation.setBuilding("Yorktown");
                annotation.addToIndexes();
                pos = matcher.end();
        }
...}
```

UIMA Document Analyzer



Document Analyzer
File
Help
Unstructured Information Management Architecture
An Apache Incubator Project
Browse...
Browse...
Browse...
Run
Interactive
View

Input Directory: examples\data

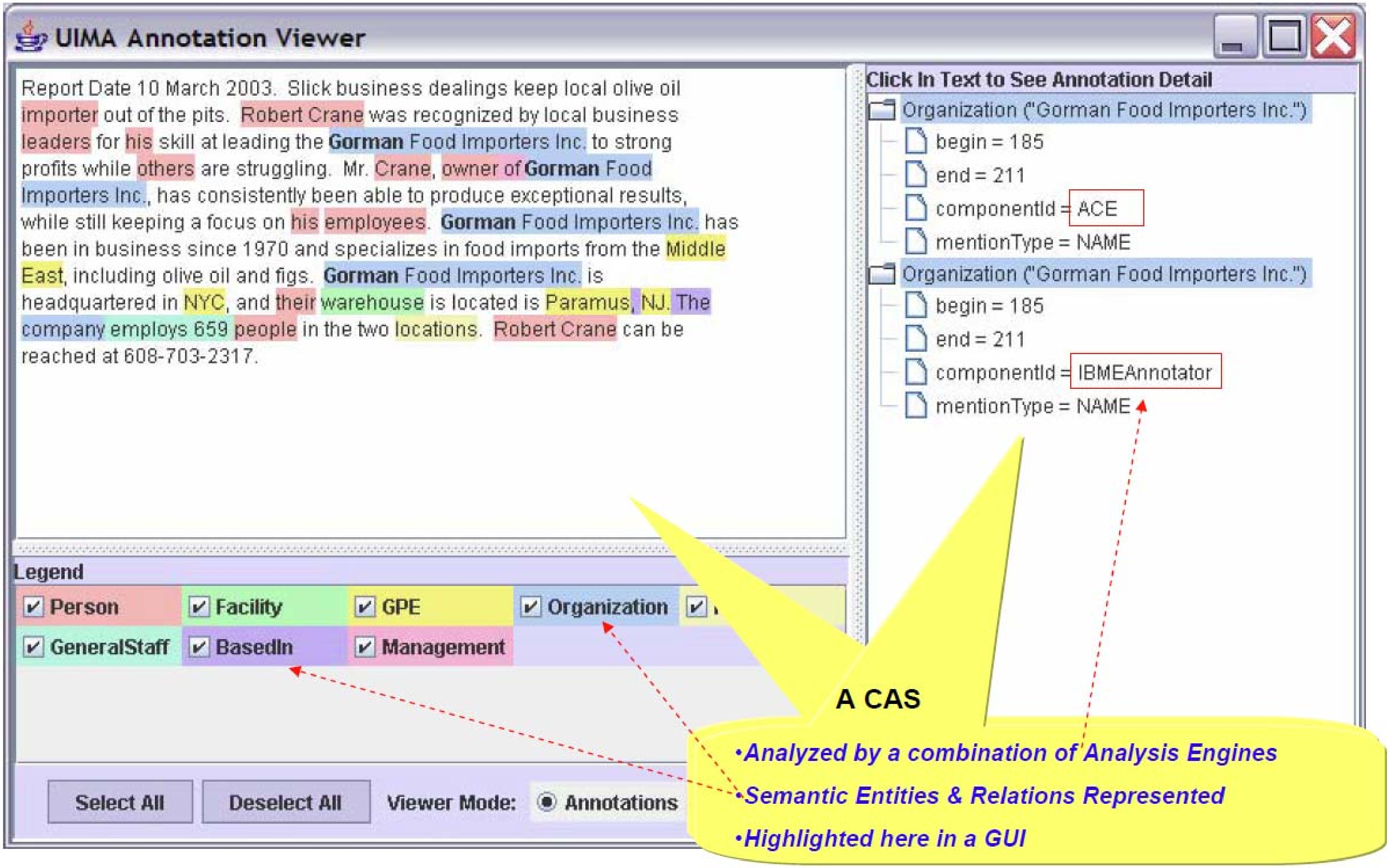
Output Directory: examples\data\processed

Location of Analysis Engine XML Descriptor: examples\descriptors\analysis_engine\PersonTitleAnnotator.xml

XML Tag containing Text (optional):

Language: en

Character Encoding: UTF-8



Report Date 10 March 2003. Slick business dealings keep local olive oil importer out of the pits. Robert Crane was recognized by local business leaders for his skill at leading the Gorman Food Importers Inc. to strong profits while others are struggling. Mr. Crane, owner of Gorman Food Importers Inc., has consistently been able to produce exceptional results, while still keeping a focus on his employees. Gorman Food Importers Inc. has been in business since 1970 and specializes in food imports from the Middle East, including olive oil and figs. Gorman Food Importers Inc. is headquartered in NYC, and their warehouse is located is Paramus, NJ. The company employs 659 people in the two locations. Robert Crane can be reached at 608-703-2317.

Click In Text to See Annotation Detail

- Organization ("Gorman Food Importers Inc.")
  - begin = 185
  - end = 211
  - componentId = ACE
  - mentionType = NAME
- Organization ("Gorman Food Importers Inc.")
  - begin = 185
  - end = 211
  - componentId = IBMEAnnotator
  - mentionType = NAME

Legend

☑ Person	☑ Facility	☑ GPE	☑ Organization	☑
☑ GeneralStaff	☑ BasedIn	☑ Management		

A CAS

- *Analyzed by a combination of Analysis Engines*
- *Semantic Entities & Relations Represented*
- *Highlighted here in a GUI*

Future assignment

- Create UIMA annotator for rooms in Computer Science
  (CompSci | CS) -? (1|2) [0-9][0-9] [0-9]

UIMA installation

- 1. Download eclipse at http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/indigor
- 1.1 Install Eclipse Modeling Framework
  - http://www.eclipse.org/modeling/emf/updates/
    - Go to help -> Install new software
- 2. Go to help -> Install new software
- 3. Add http://www.apache.org/dist/uima/eclipse-update-site/ and install all updates and restart eclipse
- 4. Download the SDK at http://uima.apache.org/downloads.cgi (Do not download the source, the examples will not work)

UIMA installation

- 5. Extract the SDK into a directory.
- 6. Set up a class path variable named UIMA_HOME, whose value is the directory where you installed the UIMA SDK. This is done as follows:

- Go to Window → Preferences → Java → Build Path → Classpath Variables.

- Click "New"

- Enter UIMA_HOME (all capitals, exactly as written) in the "Name" field.

- Enter your installation directory (e.g. C:/Program Files/apache-uima) in the "Path" field

- Click "OK" in the "New Variable Entry" dialog

UIMA installation

- 6. (cont.)

- Click “OK” in the “Preferences” dialog

If it asks you if you want to do a full build, click “Yes”

- 7. Select the File → Import menu option, Select “General/Existing Project into Workspace” and click the “Next” button.
- 8. Browse to the SDK and you should see uimaj-examples.
- 9. Click finish.

UIMA installation

- 10. Now, the first step which you actually *do* something is when you open the "TutorialTypeSystem.xml" with the "Component Descriptor Editor". The first time I right clicked the file to try to open this file it was not in the menu. To fix this, I had to restart eclipse with the - clean option. It was there after I restarted.