Advanced GATE Embedded

Additional material: UIMA/GATE integration

Fifth GATE Training Course
June 2012

Outline

Outline

What is UIMA?

- Language processing framework originally developed by IBM
- Similar document processing pipeline architecture to GATE
- Concentrates on performance and scalability
- Supports components written in different programming languages (currently Java and C++)
- Native support for distributed processing via web services

UIMA Terminology

- Processing tasks in UIMA are encapsulated in *Analysis Engines* (AEs)
- In UIMA, AEs can be *primitive* ($\sim$ a single PR in GATE terms), or *aggregate* ($\sim$ a GATE controller).
  - Aggregate AE can include other primitive or aggregate AEs
- GATE includes interoperability layer to run
  - GATE controller as a (primitive) AE in UIMA
  - UIMA AE (primitive or aggregate) as a GATE PR

UIMA and GATE

- In GATE, unit of processing is the *Document*
  - Text, plus features, plus annotations
  - Annotations can have arbitrary features, with any Java object as value
- In UIMA, unit of processing is *CAS* (common analysis structure)
  - Text, plus *Feature Structures*
  - Annotations are just a special kind of FS, which includes start and end offset features

Key Differences

- In GATE, annotations can have any features, with any values
- In UIMA, feature structures are *strongly typed*
  - Must declare what types of annotations are supported by each analysis engine
  - Must specify what features each annotation type supports
  - Must specify what *type* feature values may take
    - Primitive types - string, integer, float
    - Reference types - reference to another FS in the CAS
    - Arrays of the above
  - All defined in XML descriptor for the AE

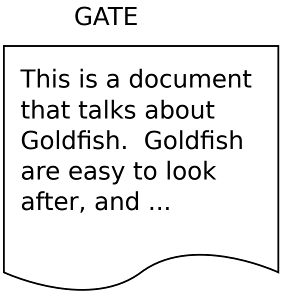
Integrating GATE and UIMA

- So the problem is to map between the loosely-typed GATE world and the strongly-typed UIMA world
- Best explained by example…

Example 1

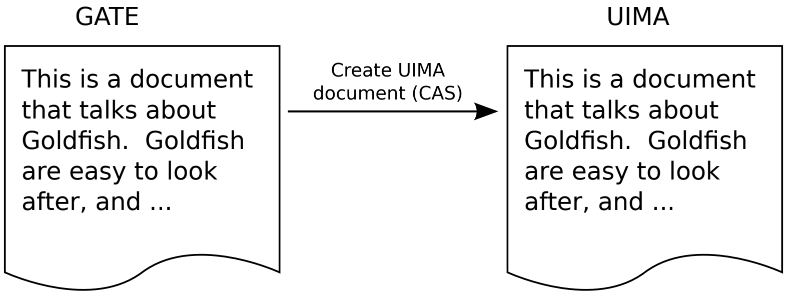
- Simple UIMA annotator that annotates each instance of the word "Goldfish" in a document.
- Does not need any input annotations
- Produces output annotations of type `gate.example.Goldfish`

Example 1



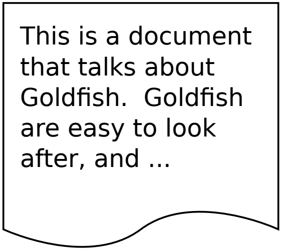
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

Example 1


GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Create UIMA document (CAS)
UIMA
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

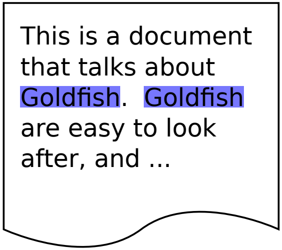
Example 1

GATE



This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

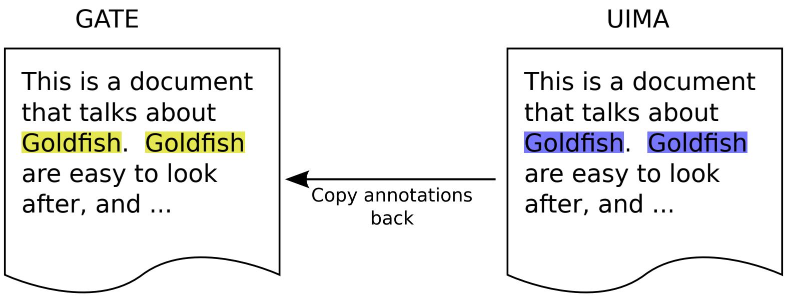
UIMA



This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

UIMA AE runs, creating gate.example.Goldfish annotations

Example 1

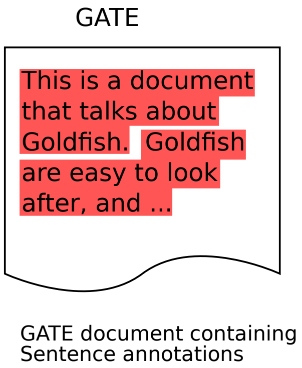

GATE
UIMA
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Copy annotations back

Create GATE annotations of type Goldfish at the corresponding places

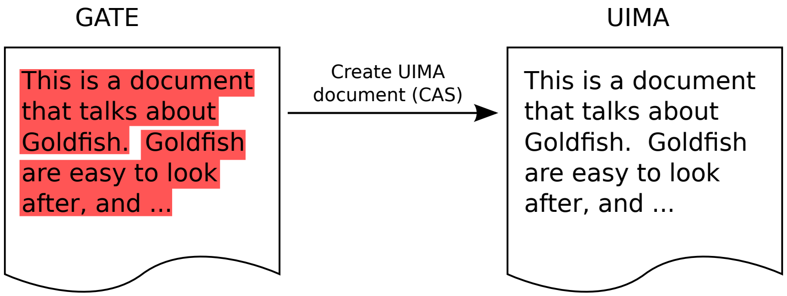
Example 2

- We may want to copy annotations, as well as text, from the original GATE document.
- Consider a UIMA annotator that
  - takes `gate.example.Sentence` annotations as input
  - annotates “Goldfish” as before
  - also adds a feature `GoldfishCount` to each Sentence giving the number of goldfish annotations in that sentence

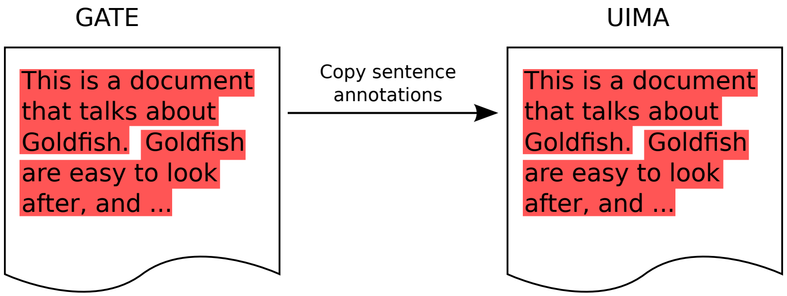
Example 2


GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
GATE document containing Sentence annotations

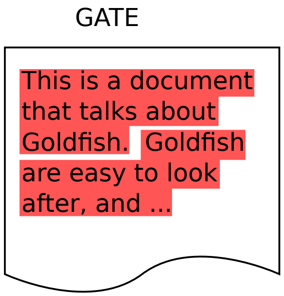
Example 2

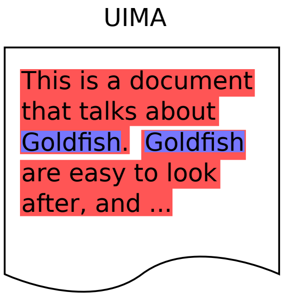

GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Create UIMA document (CAS)
UIMA
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

Example 2


GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Copy sentence annotations
UIMA
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

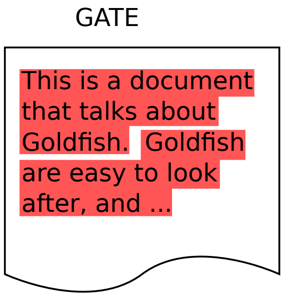
Example 2

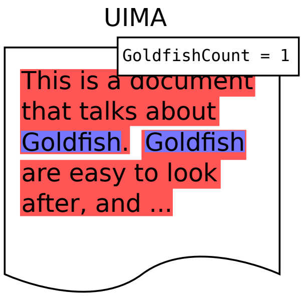

GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...


UIMA
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

UIMA AE runs, creating gate.example.Goldfish annotations

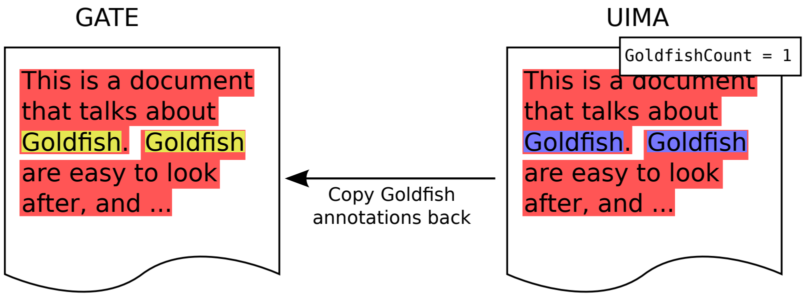
Example 2


GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

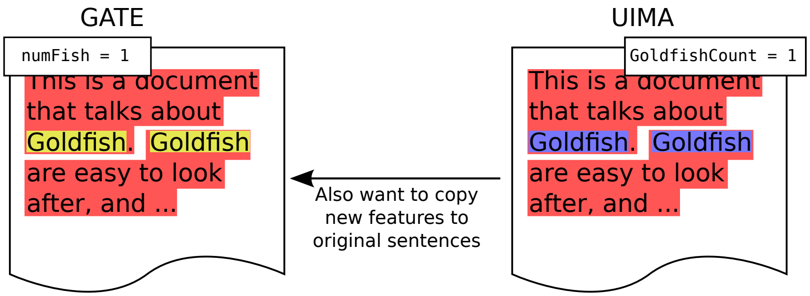

UIMA
GoldfishCount = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

and adding a feature to each sentence

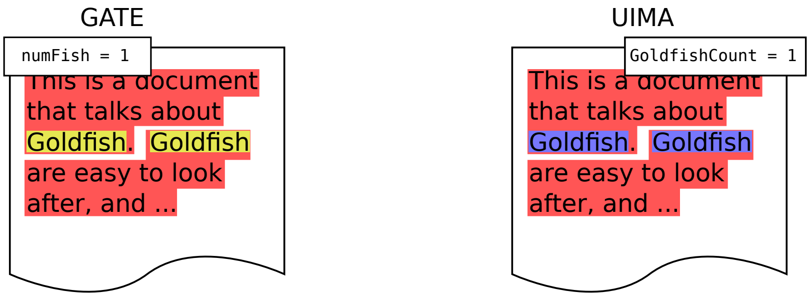
Example 2


GATE
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Copy Goldfish annotations back
UIMA
GoldfishCount = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

Example 2


GATE
numFish = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
UIMA
GoldfishCount = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
Also want to copy new features to original sentences

Example 2


GATE
numFish = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...
UIMA
GoldfishCount = 1
This is a document that talks about Goldfish. Goldfish are easy to look after, and ...

We need an index linking the UIMA annotations to the GATE annotations they came from

Defining the Mapping

The mapping is defined by the user in an XML file:

```
<uimaGateMapping>
  <inputs>
    <uimaAnnotation type="gate.example.Sentence"
                    gateType="Sentence"
                    indexed="true"/>
  </inputs>
```

Defining the Mapping

The mapping is defined by the user in an XML file:

```
<uimaGateMapping>
  <inputs>
    <uimaAnnotation type="gate.example.Sentence"
                    gateType="Sentence"
                    indexed="true"/>
  </inputs>
```

For each GATE annotation of type `Sentence` . . .

Defining the Mapping

The mapping is defined by the user in an XML file:

```
<uimaGateMapping>
  <inputs>
    <uimaAnnotation type="gate.example.Sentence"
                    gateType="Sentence"
                    indexed="true"/>
  </inputs>
```

…create a UIMA annotation of type `gate.example.Sentence` at the same place …

Defining the Mapping

The mapping is defined by the user in an XML file:

```
<uimaGateMapping>
  <inputs>
    <uimaAnnotation type="gate.example.Sentence"
                    gateType="Sentence"
                    indexed="true"/>
  </inputs>
```

. . . and remember this mapping.

Defining the Mapping

```
<outputs>
  <added>
    <gateAnnotation type="Goldfish"
          uimaType="gate.example.Goldfish" />
  </added>
```

For each UIMA annotation of this type . . .

Defining the Mapping

```
<outputs>
  <added>
    <gateAnnotation type="Goldfish"
          uimaType="gate.example.Goldfish" />
  </added>
```

. . . add a GATE annotation at the same place.

Defining the Mapping

```
    <updated>
      <gateAnnotation type="Sentence"
          uimaType="gate.example.Sentence">
        <feature name="numFish">
          <uimaFSFeatureValue
            name="gate.example.Sentence:GoldfishCount"
            kind="int" />
        </feature>
      </gateAnnotation>
    </updated>
  </outputs>
</uimaGateMapping>
```

For each UIMA annotation of this type . . .

Defining the Mapping

```
      <updated>
        <gateAnnotation type="Sentence"
            uimaType="gate.example.Sentence">
          <feature name="numFish">
            <uimaFSFeatureValue
              name="gate.example.Sentence:GoldfishCount"
              kind="int" />
          </feature>
        </gateAnnotation>
      </updated>
    </outputs>
  </uimaGateMapping>
```

. . . find the GATE annotation it came from . . .

Defining the Mapping

```
    <updated>
      <gateAnnotation type="Sentence"
          uimaType="gate.example.Sentence">
        <feature name="numFish">
          <uimaFSFeatureValue
            name="gate.example.Sentence:GoldfishCount"
            kind="int" />
        </feature>
      </gateAnnotation>
    </updated>
  </outputs>
</uimaGateMapping>
```

. . . and set this annotation's `numFish` feature . . .

Defining the Mapping

```
    <updated>
      <gateAnnotation type="Sentence"
          uimaType="gate.example.Sentence">
        <feature name="numFish">
          <uimaFSFeatureValue
            name="gate.example.Sentence:GoldfishCount"
            kind="int" />
        </feature>
      </gateAnnotation>
    </updated>
  </outputs>
</uimaGateMapping>
```

...to the value of the `GoldfishCount` feature from the UIMA annotation.

Embedding UIMA in GATE

- Write the mapping descriptor
  - Must ensure that all the annotations and features declared as input capabilities by the UIMA AE are supplied by the mapping.
  - Must not attempt to map to a UIMA FS type that is not declared in the AE's type system.
- For a Java AE, need to get UIMA AE implementation class onto the GATE ClassLoader: define a plugin with just the relevant `<JAR>` entries:

```
<CREOLE-DIRECTORY>
  <JAR>myUimaAE.jar</JAR>
  <JAR>some-dependency.jar</JAR>
</CREOLE-DIRECTORY>
```

- Load this plugin (in addition to the `UIMA` plugin)

Embedding UIMA in GATE

- For C++ AEs, put the implementation library somewhere Java can find it.
- For remote service AEs no additional config is required.
- Create an instance of `gate.uima.AnalysisEnginePR` ("UIMA Analysis Engine" in GATE Developer)
- Init parameters are URLs to the UIMA AE descriptor XML and the mapping descriptor.
- Runtime parameter is the `annotationSetName` containing the annotations to map.
  - If you need to map annotations from several sets, use annotation set transfer or JAPE.

Embedding GATE in UIMA

- Embedding a GATE `CorpusController` as a UIMA AE is the mirror-image of this process.
- Controller must be saved as an `.xgapp` with all PR runtime parameter values (except document and corpus) pre-configured correctly.
- Mapping descriptor format is the same (but `<gateAnnotation>` in the input section and `<uimaAnnotation>` in the output section)
- Each `<gateAnnotation>` or `<uimaAnnotation>` element can specify an `annotationSet` attribute, to support mapping to/from several GATE annotation sets.
  - on input – create the GATE annotation in this set
  - on output – look for the GATE annotation in this set

Embedding GATE in UIMA

- Include `gate.jar`, the appropriate JARs from GATE's `lib`, and `uima-gate.jar` from the UIMA plugin on classpath.
- GATE provides a skeleton AE descriptor which needs to be customized
  - type system and capabilities to match the GATE mapping
  - external resource bindings to point to the saved `.xgapp` and the mapping descriptor.
- The AE will initialize GATE if necessary – UIMA application doesn't need to know it's embedding GATE.
- For more details, see the user guide (`http://gate.ac.uk/userguide/chap:uima`) and the `test` directory under `plugins/UIMA`.

Exercise 1: Embedding UIMA in GATE

Run some of the example UIMA-in-GATE code provided with GATE

- Load the UIMA plugin
- Load `plugins/UIMA/examples` as a plugin (you'll need to "Add a CREOLE repository")
    - This loads the implementation classes for the example UIMA AEs.
- Load a default ANNIE application
- Create a UIMA Analysis Engine PR with these parameters (relative to `plugins/UIMA/examples/conf`) and add it to the end of the ANNIE application
    - analysisEngineDescriptor: `uima_descriptors/TokenHandlerAggregate.xml`
    - mappingDescriptor: `mapping/TokenHandlerGateMapping.xml`

Exercise 1: Embedding UIMA in GATE

- Run the application over a document of your choice - Token annotations have a `numLower` feature giving the number of lowercase letters in the token.
- Code is in `plugins/UIMA/examples/src`, have a look at the code and the mapping descriptor, see how the mapping is configured.
- Try changing the mapping to map the LowerCaseLetters feature from UIMA to a different name in GATE.
- Other AE descriptors and their associated mappings if you want to experiment further.

Exercise 2: Embedding GATE in UIMA

- The `plugins/UIMA/test` directory contains an example UIMA AE descriptor that wraps a GATE application.
- `conf/TokenizerAndPOSTagger.xml` is an aggregate AE that runs
  - A native UIMA token and sentence annotator
  - The GATE POS tagger to add POS tags to the tokens
- UIMA provides a basic UI to run an AE and inspect the results, which you can run with `../../bin/ant documentanalyser` in `plugins/UIMA` (backslashes on Windows).
  - This starts up the tool with a classpath that includes the relevant JARs to run the GATE application AE.

Exercise 2: Embedding GATE in UIMA

- Start the document analyser tool.
- Create an empty directory, and set the "Output directory" option to point to it.
- Set the "Location of Analysis Engine XML Descriptor" to point to the aggregate descriptor (`test/conf/TokenizerAndPOSTagger.xml`).
- Click the "Interactive" button
- Type (or paste) some text and click "Analyze".
- If you're a confident UIMA user, try modifying the mapping to change the POS feature name (you will need to edit the type system to match).