Implementing a Variety of Linguistic Annotations

through a Common Web-Service Interface

Adam Funk, Ian Roberts, Wim Peters

University of Sheffield

18 May 2010

Outline

Introduction

Goals:

- contribute to the CLARIN European Demonstrator prototype
- web-services
- reusability
- re-use
- uniform interface

Web service implementation

- SOAP
- MTOM specification (send binary data as binary, saving 1/3 compared with Base64 encoding
- common WSDL specification

  input binary data to be turned into a GATE *Document* (XML, HTML, PDF, etc.)

  output any valid XML Element

Web service implementation

- Spring configuration framework
- Apache CXF toolkit (which uses Spring)
- an instance of GATE's *DocumentProcessor* interface at the core of each service

the *DocumentProcessor* interface

- one method: *processDocument(gate.Document)*
- API description: “Very simple interface for a component that processes GATE documents. Typical implementations of this interface would contain a Controller but the interface is deliberately generic.”
- implemented by *LanguageAnalyserDocumentProcessor*, which also has a *setAnalyser(LanguageAnalyser)* method (including *Controller*)
- can be implemented by a more complicated class

Services currently available

annie-alpha runs the ANNIE NER and co-reference pipeline and returns the fully annotated document in GATE XML format (including original HTML or XML mark-up).

maf-en runs GATE's basic NLP components (sentence-splitter, tokenizer, POS-tagger, and lemmatizer) for English and returns an XML document according to the MAF standard.

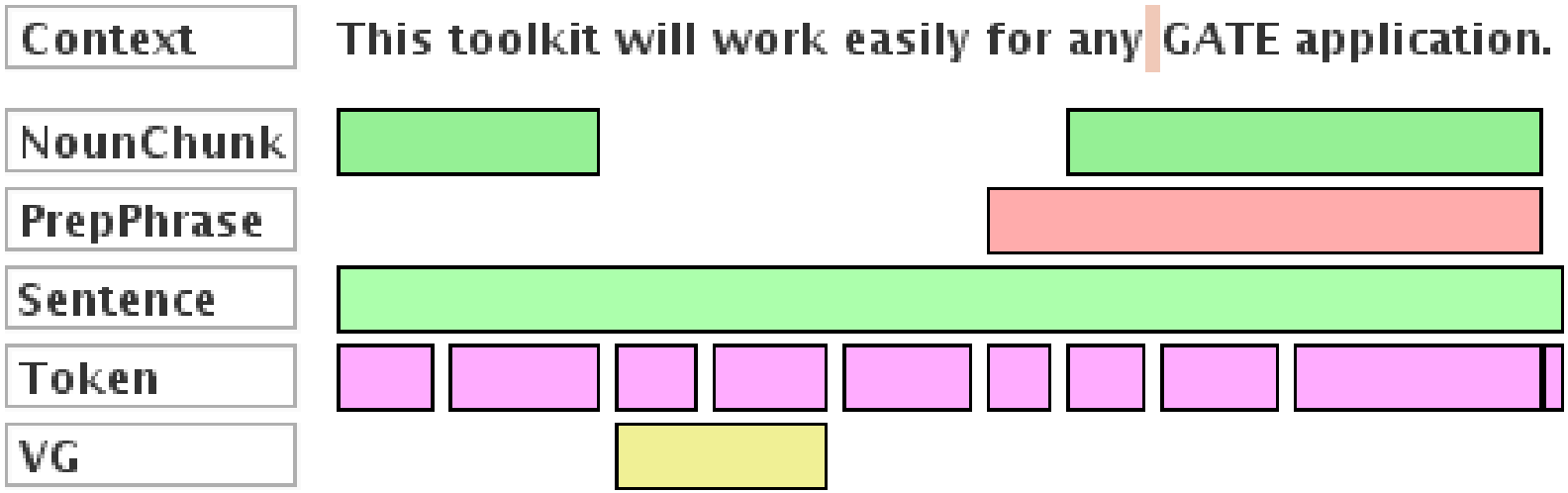
(XML document . . . root element in the SOAP response)

Services currently available

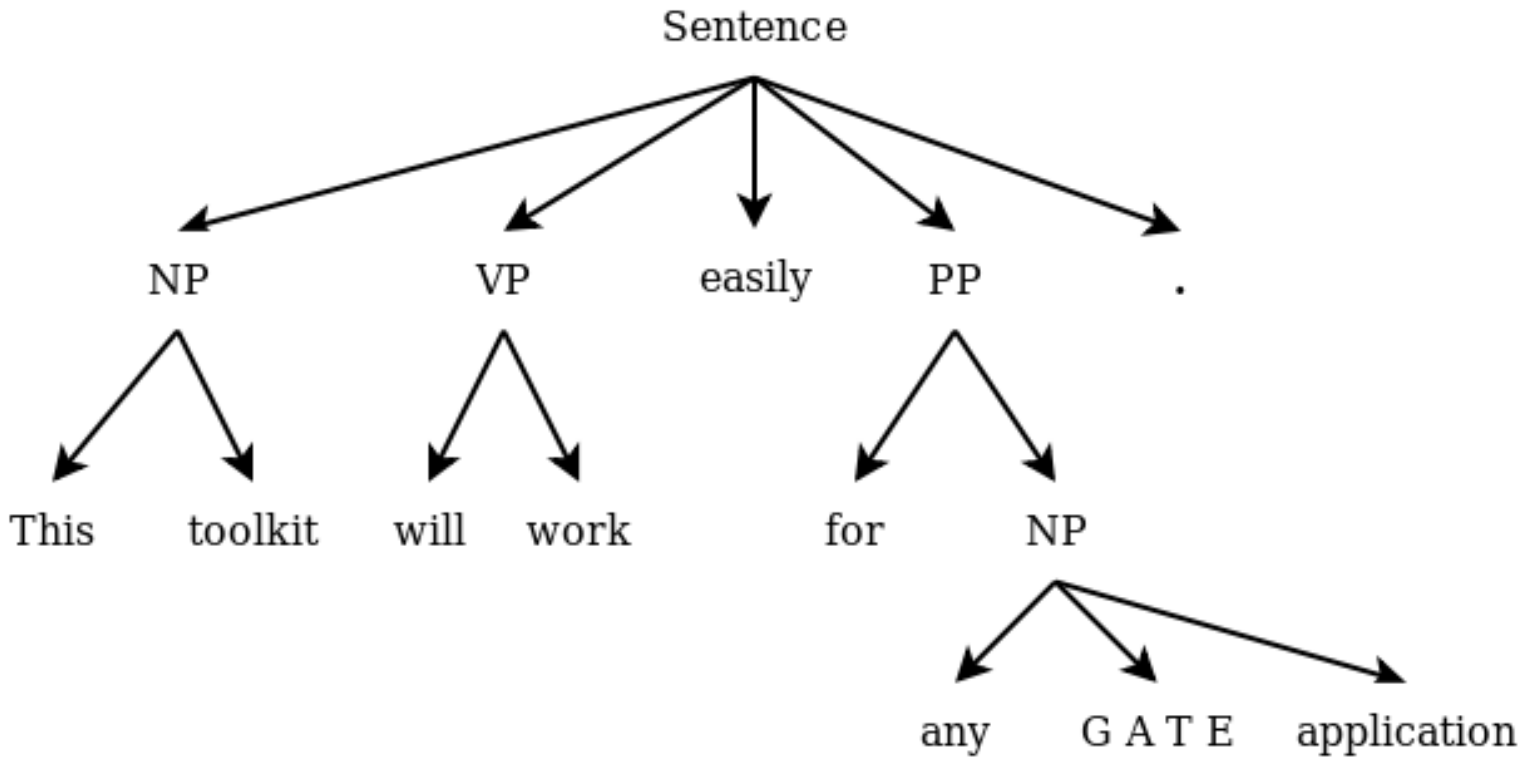
chunking-synaf-en runs the *maf-en* components and NP, VP, and PP chunkers for English; composes a simple syntactic tree from the chunks based on containment, and returns a SYNAF XML document.

annie-rdf runs ANNIE, analyses the annotations by type and features, generates RDF representing the entities according to the PROTON ontology, and returns an RDF-XML document.

Chunking example



Chunking example



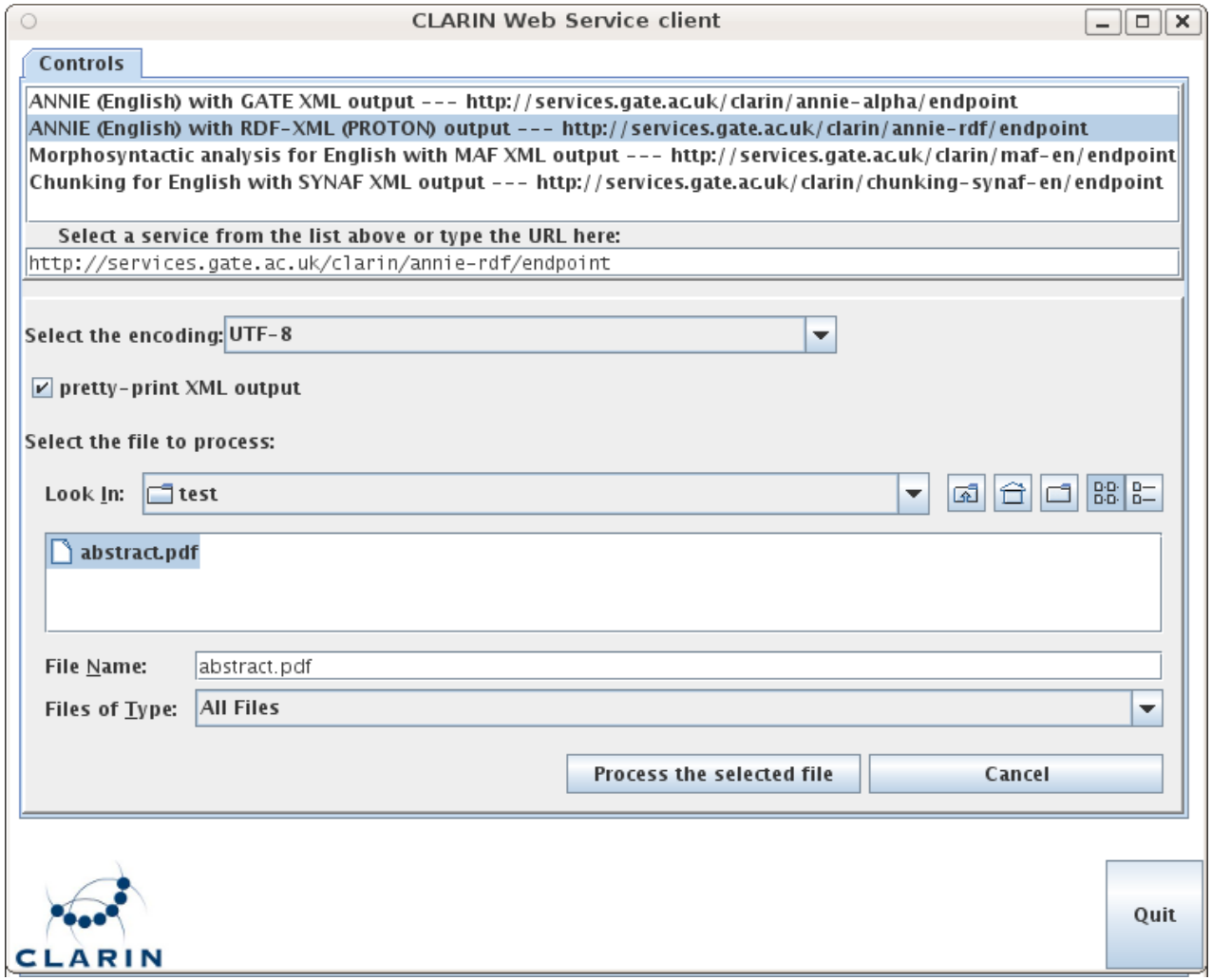
Reference client

Our reference client

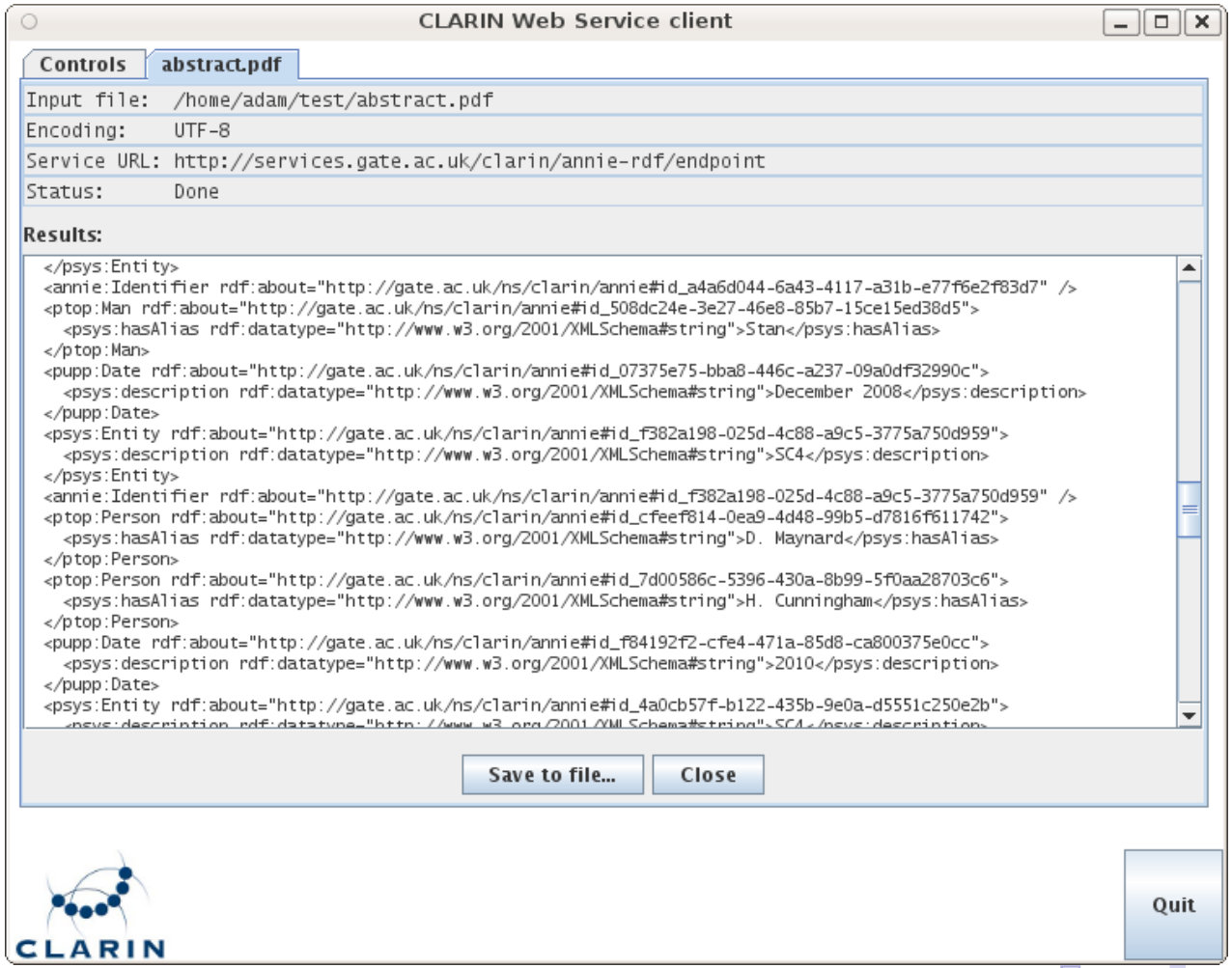
- is supplied as a ZIP file with all necessary libraries, requiring only a Java 1.5 runtime environment;
- comes with a menu of the four current services (updated as more are added);
- sends the file contents, file URL, and selected encoding to the service;
- presents the service's output and allows the user to save it to an XML file.

But developers can produce their own client(s) from the services' WSDL files.

Reference client



Reference client



Future work

- We are adding harvestable metadata for CLARIN integration.
- We want to deploy MAF services for other European languages, and we can do more (better, faster) if others can share with us
  - processing tools—especially if they are easy to integrate into GATE (a Java API works best); and
  - language resources—especially tagged corpora.

  GATE is most fully developed for English—support for other languages varies, and we always welcome contributions to improve this.

Shameless promotion

GATE is free and open (LGPL) and community-supported (Sourceforge mailing list). We also offer customization services, formal training, and certification.

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

`http://gate.ac.uk/conferences/montreal-2010/`