WebLicht – A Service Oriented Architecture for Language Resources and Tools

Erhard Hinrichs, Thomas Zastrow

University of Tübingen

Munich, October 17 2009

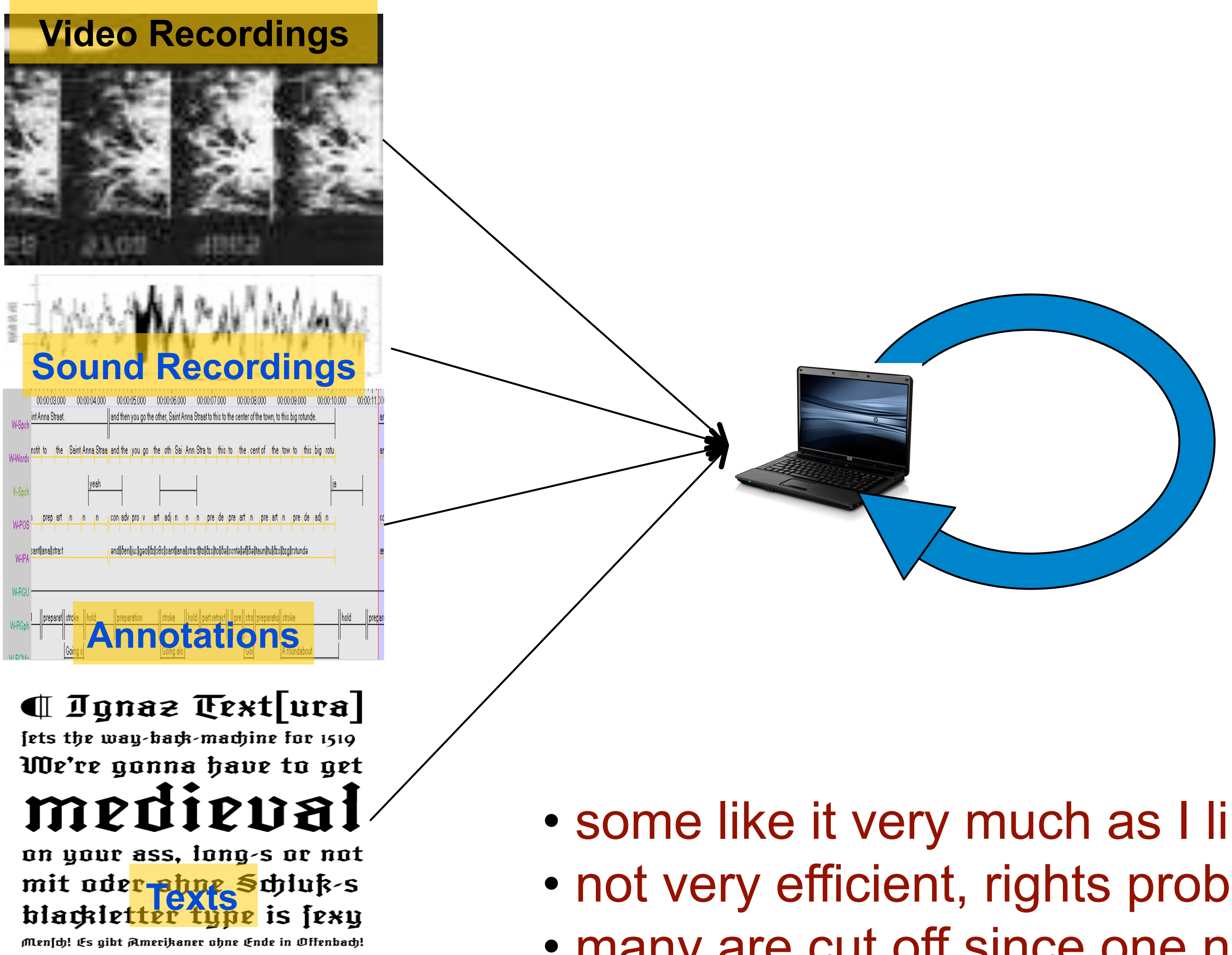
Current Situation

- Many linguistic resources (corpora, dictionaries, …) and tools (tokenizer, tagger, parser, …) are available
- Most of them are implemented to run on local machines. This can be inconvenient and error-prone ….
- …on the user side:
  - Every potential user has to download and install them on his own machine: this may cause problems with operating systems, compiler versions, missing libraries, …
  - Keep an eye on updates, (security) patches, new versions etc.

Current Situation

- … on the developers side:
  - How to publish LRT?
  - Question of license, user permissions, …
  - Combination and comparison with other tools/resources
  - Sustainability, long term support

Current Download-First Paradigm



- installing
- downloading
- adapting
- converting
- scripting
- etc etc

- some like it very much as I liked the VW Käfer
- not very efficient, rights problems, etc
- many are cut off since one needs IT skills
- cyberspace needs to overcome this scenario

One Possible Solution

- **Make LRT available on the web! –**

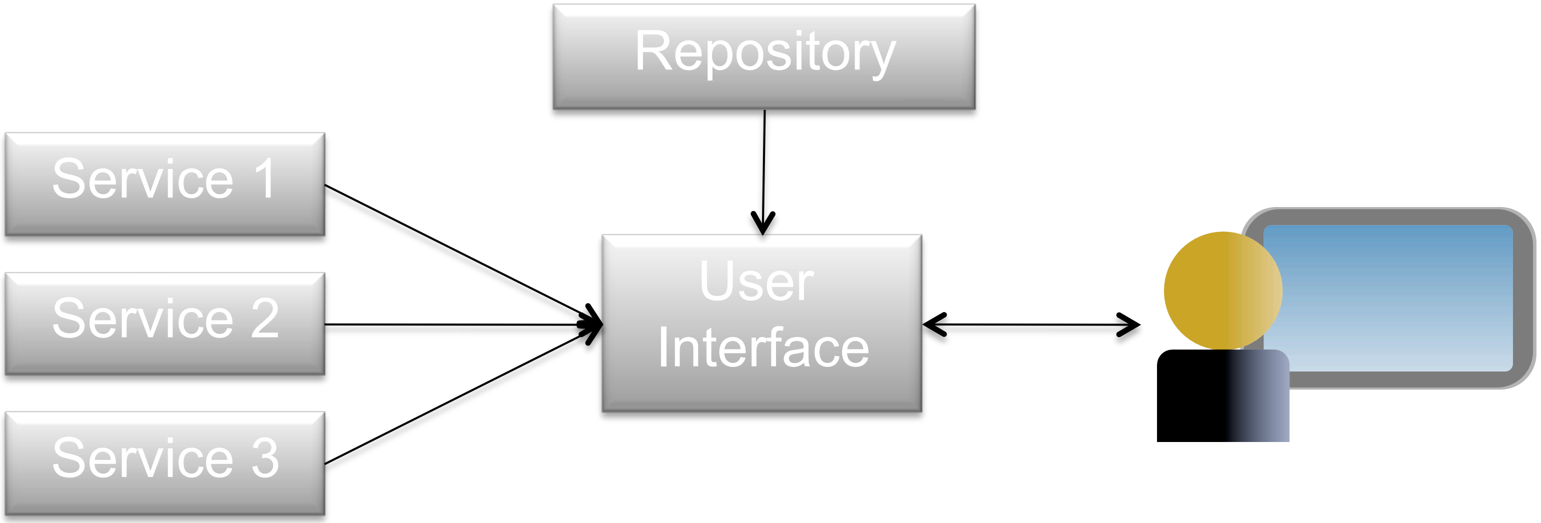
- For some kinds of LRT, its easy to put them online (make resources downloadable, offer search engines etc.)
- For other kinds, more effort is necessary (limiting access to resources, how to make tools online usable)

➔ Solution: a ***Service Oriented Architecure* (SOA)**

- WebLicht: a Service Oriented Architecture for incremental automatic annotation of text corpora
- Work started in October 2008
- Participants (September 2009):
  - BBAW Berlin
  - ASV Leipzig
  - IDS Mannheim
  - IMS Stuttgart
  - SfS Tübingen

Service Oriented Architectures

- Components of a SOA
  - ***Distributed Services***: offering functionality (resources & tools) over the (inter-)net. Mostly implemented as webservices
  - ***Repository***: stores metadata and technical information about the services
  - ***User interface***: interacts with the user and combines services and information from the repository



Repository

Service 1

Service 2

Service 3

User Interface

The Services

- Services are implemented as REST style webservices
- HTTPs POST method is used to send data from the UI to the services
- As client, *anything* which is able to use the HTTP protocol, can be used:
  - Browser
  - Commandline tools (wget, curl)
  - Programming Languages

➔ Anyone can implement his/her own interface to WebLicht

The Repository

- Implemented at the ASV Leipzig
- It offers information and a query engine for the services:
  - Which services are available?
  - How can I combine them?
  - Which input/output format does a service accept/produce?

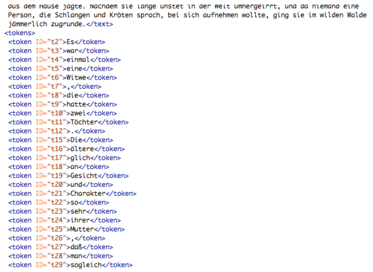
- Example: a tokenizer is already applied to a plain text, which services can be used next?

Web 2.0 Application for Tool Chaining and Execution

- Implemented at the SfS Tübingen
- Java application, deployed in Apache Tomcat
- Allows the user to
    - upload a text (plain text, MS Word, RTF or PDF files)
    - construct a text from corpora in Leipzig
    - use some hardwired example texts
- Build a chain of linguistic tools
- Executes the tool chain with the uploaded text and presents the results
- During the chaining process, it queries the repository for available services

WebLicht

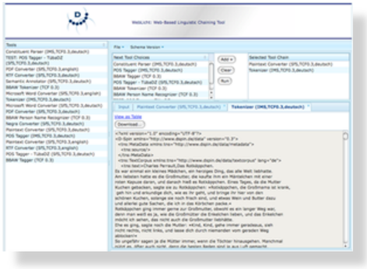
Stuttgart



```
 aus dem Hause jagte. Nachdem sie lange unstet in der Welt umhergeirrt, und da niemand eine
 Person, die Schlangen und Kröten sprach, bei sich aufnehmen wollte, ging sie im wilden Walde
 jämmerlich zugrunde.</text>
<tokens>
 <token ID="t2">Es</token>
 <token ID="t3">war</token>
 <token ID="t4">einmal</token>
 <token ID="t5">eine</token>
 <token ID="t6">Witwe</token>
 <token ID="t7">,</token>
 <token ID="t8">die</token>
 <token ID="t9">hatte</token>
 <token ID="t10">zwei</token>
 <token ID="t11">Töchter</token>
 <token ID="t12">.</token>
 <token ID="t15">Die</token>
 <token ID="t16">ältere</token>
 <token ID="t17">glich</token>
 <token ID="t18">an</token>
 <token ID="t19">Gesicht</token>
 <token ID="t20">und</token>
 <token ID="t21">Charakter</token>
 <token ID="t22">so</token>
 <token ID="t23">sehr</token>
 <token ID="t24">ihrer</token>
 <token ID="t25">Mutter</token>
 <token ID="t26">,</token>
 <token ID="t27">daß</token>
 <token ID="t28">man</token>
 <token ID="t29">sogleich</token>
```

Standard-conformant
Text Corpus Encoding

Tübingen



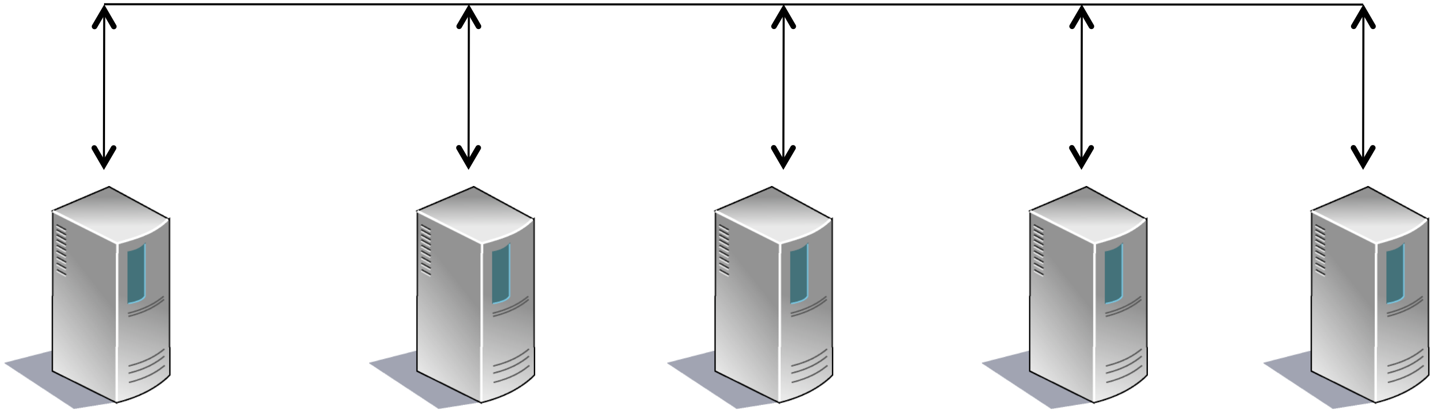
WebLicht: Web-Based Linguistic Chaining Tool
Tools
Constituent Parser (IMS,TCF0.3,deutsch)
TEST: POS Tagger - TübaDZ (SfS,TCF0.3,deutsch)
PDF Converter (SfS,TCF0.3,englisch)
RTF Converter (SfS,TCF0.3,deutsch)
Semantic Annotator (SfS,TCF0.3,deutsch)
BBAW Tokenizer (TCF 0.3)
Microsoft Word Converter (SfS,TCF0.3,englisch)
Tokenizer (IMS,TCF0.3,deutsch)
Microsoft Word Converter (SfS,TCF0.3,deutsch)
PDF Converter (SfS,TCF0.3,deutsch)
BBAW Person Name Recognizer (TCF 0.3)
Negra Converter (SfS,TCF0.3,deutsch)
Plaintext Converter (SfS,TCF0.3,deutsch)
POS Tagger (IMS,TCF0.3,deutsch)
Plaintext Converter (SfS,TCF0.3,englisch)
RTF Converter (SfS,TCF0.3,englisch)
POS Tagger - TübaDZ (SfS,TCF0.3,deutsch)
BBAW Tagger (TCF 0.3)
File
Schema Version
Next Tool Choices
Constituent Parser (IMS,TCF0.3,deutsch)
POS Tagger (IMS,TCF0.3,deutsch)
BBAW Tagger (TCF 0.3)
POS Tagger - TübaDZ (SfS,TCF0.3,deutsch)
BBAW Tokenizer (TCF 0.3)
BBAW Person Name Recognizer (TCF 0.3)
Add »
Clear
Run
Selected Tool Chain
Plaintext Converter (SfS,TCF0.3,deutsch)
Tokenizer (IMS,TCF0.3,deutsch)
Input
Plaintext Converter (SfS,TCF0.3,deutsch)
Tokenizer (IMS,TCF0.3,deutsch)
View as Table
Download...

```
<?xml version="1.0" encoding="UTF-8"?>
<D-Spin xmlns="http://www.dspin.de/data" version="0.3">
  <tns:MetaData xmlns:tns="http://www.dspin.de/data/metadata">
    <tns:source/>
  </tns:MetaData>
  <tns:TextCorpus xmlns:tns="http://www.dspin.de/data/textcorpus" lang="de">
    <tns:text>Charles Perrault,Das Rotkäppchen.
Es war einmal ein kleines Mädchen, ein herziges Ding, das alle Welt liebhatte.
Am liebsten hatte es die Großmutter; die kaufte ihm ein Mäntelchen mit einer
roten Kapuze daran, und danach hieß es Rotkäppchen. Eines Tages, da die Mutter
Kuchen gebacken, sagte sie zu Rotkäppchen: »Rotkäppchen, die Großmama ist krank,
 geh hin und erkundige dich, wie es ihr geht, und bringe ihr hier von den
schönen Kuchen, solange sie noch frisch sind, und etwas Wein und Butter dazu
und allerlei gute Sachen, die ich in das Körbchen packe.«
Rotkäppchen ging immer gerne zur Großmutter, obwohl es ein langer Weg war,
denn man weiß es ja, wie die Großmütter die Enkelchen lieben, und das Enkelchen
möcht ich sehen, das nicht auch die Großmutter liebhätte.
Ehe es ging, sagte noch die Mutter: »Kind, Kind, gehe immer geradeaus, sieh
nicht rechts, nicht links, und lasse dich durch niemanden vom geraden Weg
ablocken!«
So ungefähr sagen ja die Mütter immer, wenn die Töchter hinausgehen. Manchmal
nützt es, öfter auch nicht, denn die besten Reden sind ja aus Luft gemacht.
```

Web 2.0 Application for
Tool Chaining
and Execution

Leipzig

Repository



Stuttgart

Tübingen

Berlin

Leipzig

Mannheim

Live Presentation ….

Munich, October 17 2009