Semantic Annotations Tool For EIS Publications

...using crowdsourcing, ontology and SPARQL

Team : A Q M Saiful Islam
Seyidov Mirsattar

What this tool about?

This tool is about annotating EIS related doc, which normally contain lot's of tabular information.

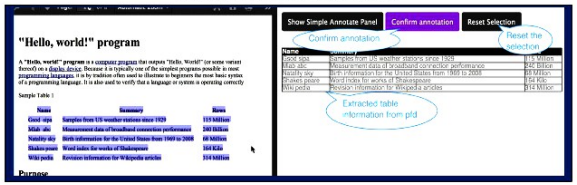
- Annotate Tables
- Transform the table information into **Data cube** vocabulary
- Store data cube into to virtuso server using SPARQL interface.

Poster

SEMANTIC ANNOTATIONS TOOL FOR EIS PUBLICATIONS

This tool is a research effort for **annotating** text and **tabular** information in a pdf document via **crowdsourcing**. It also facilitate a user to annotate tabular information using **DataCube RDF** vocabulary. This tool **transforms** selected pdf table info into data cube RDF.

ANNOTATION WINDOW



DATACUBE RDF SEGMENT OF TRANSFORMED TABLE INFO

http://example.org/ns#table1R3C2	http://eisAnnotation/#value	"Birth information for the United States from 1969 to 2008"
http://example.org/ns#table1R5C2	http://eisAnnotation/#value	"Revision information for Wikipedia articles"
http://example.org/ns#table1R1C2	http://eisAnnotation/#value	"Samples from US weather stations since 1929"
http://example.org/ns#table1R4C2	http://eisAnnotation/#value	"Word index for works of Shakespeare"
http://example.org/ns#table1R2C2	http://eisAnnotation/#value	"Measurement data of broadband connection performance"
http://example.org/ns#table1R4C3	http://eisAnnotation/#value	"164 Kilo"
http://example.org/ns#table1R2C3	http://eisAnnotation/#value	"240 Billion"
http://example.org/ns#table1R5C3	http://eisAnnotation/#value	"314 Million"
http://example.org/ns#table1R3C3	http://eisAnnotation/#value	"68 Million"
http://example.org/ns#table1R1C1	http://eisAnnotation/#value	"Good sipa"
http://example.org/ns#table1R2C1	http://eisAnnotation/#value	"Mlab abc"
http://example.org/ns#table1R3C1	http://eisAnnotation/#value	"Natality sky"
http://example.org/ns#table1R4C1	http://eisAnnotation/#value	"Shakes peare"
http://example.org/ns#table1R5C1	http://eisAnnotation/#value	"Wiki pedia"
http://example.org/ns#table1R1C3	http://eisAnnotation/#value	"115 Million "

REPRESENTATION OF CELL IN THE DATA CUBE

For example, in data cube the cell - "Birth information for the United States from 1969 to 2008", has been represented in the following way.

p	o
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://purl.org/linked-data/cube#Observation
http://eisAnnotation/#value	"Birth information for the United States from 1969 to 2008"
http://purl.org/linked-data/cube#dataSet	http://example.org/ns#table1
http://example.org/ns#table1Row	3
http://example.org/ns#table1Column	2

TECHNOLOGY USED

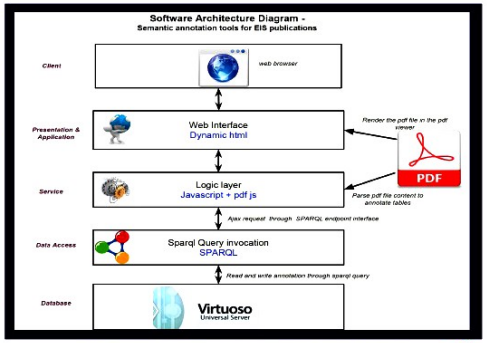
- VirtuosoUniversal Server
- SPARQL as a query
- JavaScript + Html
- Pdf.js as rendering
- Qunit as JavaScript test environment
- DataCube (http://www.w3.org/TR/vocab-data-cube/)

PHYSICAL WORK STATION

Hostname: EIS-02
Description: Physical PC at Uni Bonn
OS: Windows 7 Enterprise SP1 64bit
RAM: 16 GB (14 GB Free)
CPU: Intel Core i7-4770 @3.4GHz
HDD: (C:) 240 GB & (L:) 930 GB
IP Address: 131.220.153.88

EIS LAB

PRESENTED BY:
AQM SAIFUL ISLAM
(islam@informatik.uni-bonn.de)
SEYIDOV MIR SATTAR
(sattar.seyidov@gmail.com)



SETUP / INSTALLATION PROCESS

- Install virtuoso universal server
- Download the code from git repo
- In the downloaded directory open the index.html in any browser.
- For details set up please follow the git hub link section in the bottom.

VIRTUOSO SETTING

- Make sure that virtuoso endpoint is listening at http://localhost:8890/
- Login as admin (user: dba, pass: ***) in the backend (http://localhost:8890/conductor/)
- Update write permission settings for user : sparql
- Enable CORS for the Data Access of JS client

VM SETTINGS

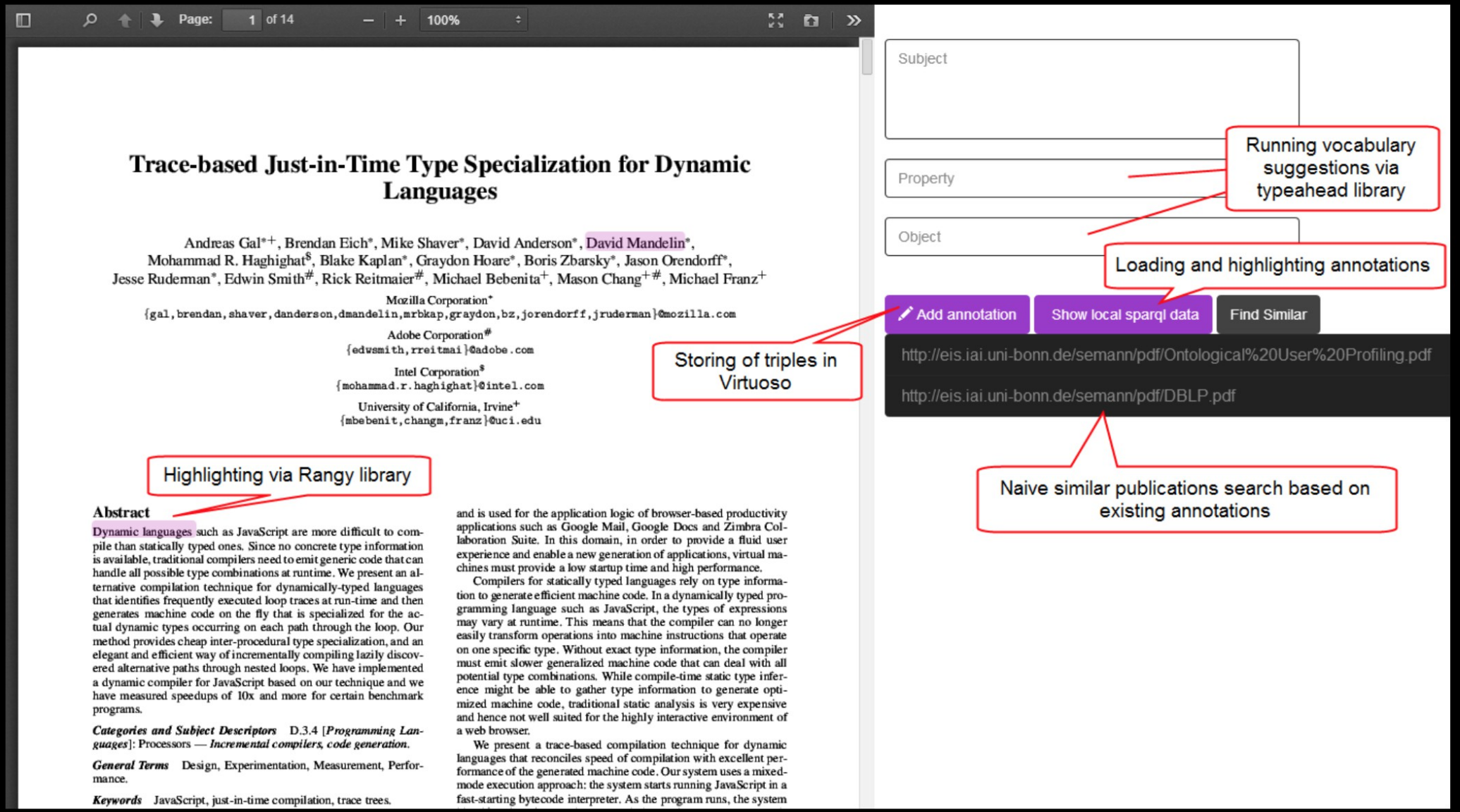
VM Name : Ubuntu-sciPubAnn
OS: Ubuntu 12.04 LTS (32 bit)
RAM: 3.9 GB
CPU: Intel® Core™ i7-4770 CPUs @3.40GHz
HDD: 4.2 GB (possible increase)

PROJECT'S GIT HUB LINKS

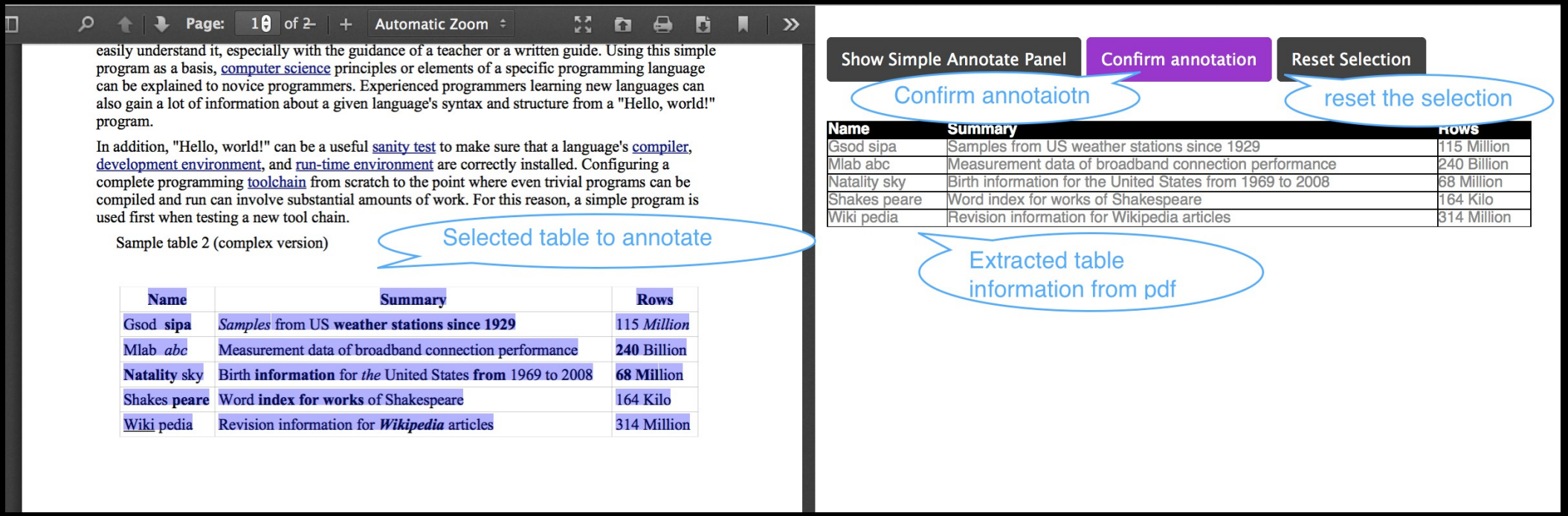
- Code link: https://github.com/saifulnipo/eis-semantic-annotation
- Documentation link : https://github.com/saifulnipo/eis-semantic-annotation/wiki
- Release note : https://github.com/saifulnipo/eis-semantic-annotation/releases

universitätbonn

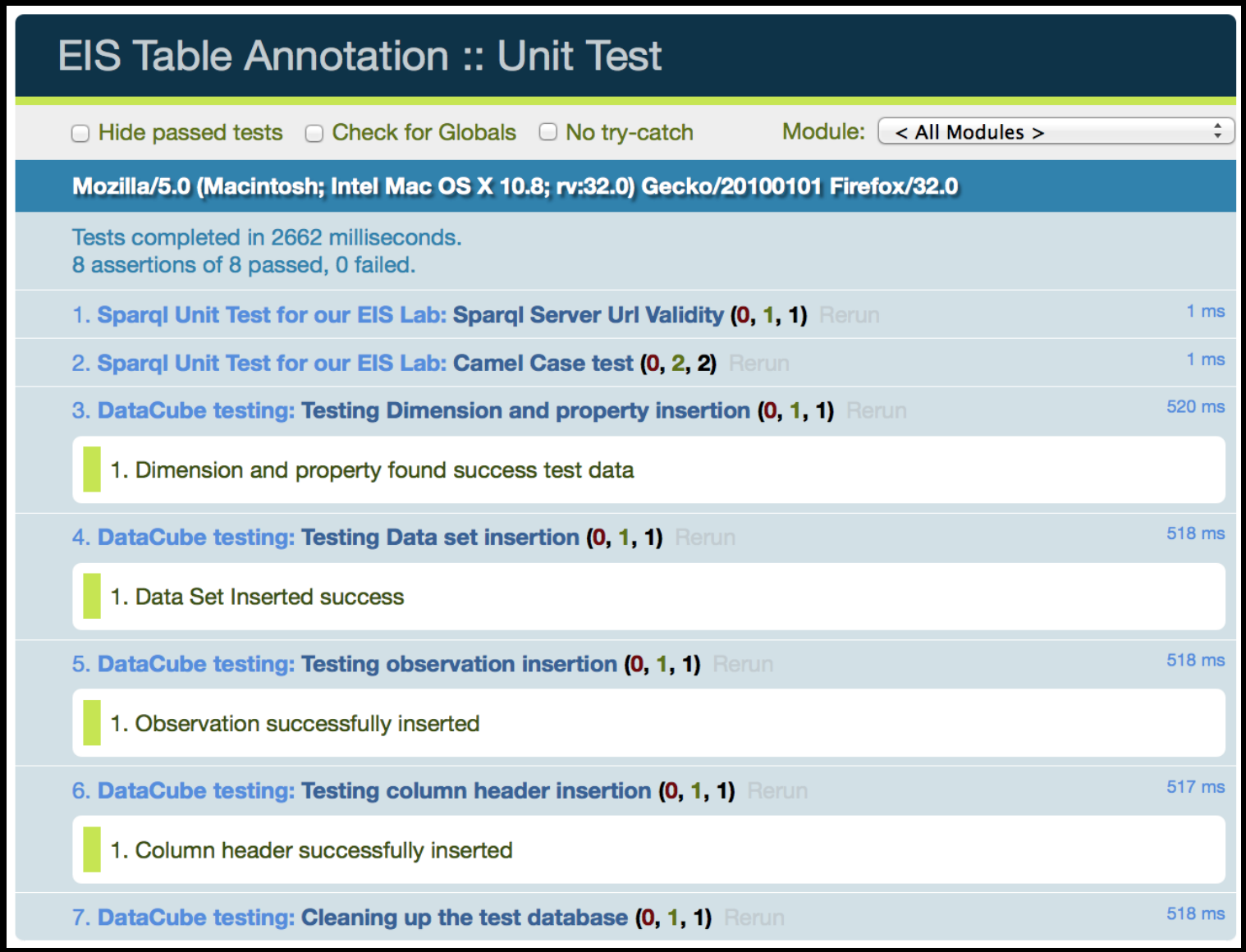
Start point


Page: 1 of 14
100%
Trace-based Just-in-Time Type Specialization for Dynamic Languages
Andreas Gal*+, Brendan Eich*, Mike Shaver*, David Anderson*, David Mandelin*,
Mohammad R. Haghighat$, Blake Kaplan*, Graydon Hoare*, Boris Zbarsky*, Jason Orendorff*,
Jesse Ruderman*, Edwin Smith#, Rick Reitmaier#, Michael Bebenita+, Mason Chang+#, Michael Franz+
Mozilla Corporation*
{gal,brendan,shaver,danderson,dmandelin,mrbkap,graydon,bz,jorendorff,jruderman}@mozilla.com
Adobe Corporation#
{edwsmith,rreitmai}@adobe.com
Intel Corporation$
{mohammad.r.haghighat}@intel.com
University of California, Irvine+
{mbebenit,changm,franz}@uci.edu
Highlighting via Rangy library
Abstract
Dynamic languages such as JavaScript are more difficult to compile than statically typed ones. Since no concrete type information is available, traditional compilers need to emit generic code that can handle all possible type combinations at runtime. We present an alternative compilation technique for dynamically-typed languages that identifies frequently executed loop traces at run-time and then generates machine code on the fly that is specialized for the actual dynamic types occurring on each path through the loop. Our method provides cheap inter-procedural type specialization, and an elegant and efficient way of incrementally compiling lazily discovered alternative paths through nested loops. We have implemented a dynamic compiler for JavaScript based on our technique and we have measured speedups of 10x and more for certain benchmark programs.
Categories and Subject Descriptors D.3.4 [Programming Languages]: Processors — Incremental compilers, code generation.
General Terms Design, Experimentation, Measurement, Performance.
Keywords JavaScript, just-in-time compilation, trace trees.
and is used for the application logic of browser-based productivity applications such as Google Mail, Google Docs and Zimbra Collaboration Suite. In this domain, in order to provide a fluid user experience and enable a new generation of applications, virtual machines must provide a low startup time and high performance.
Compilers for statically typed languages rely on type information to generate efficient machine code. In a dynamically typed programming language such as JavaScript, the types of expressions may vary at runtime. This means that the compiler can no longer easily transform operations into machine instructions that operate on one specific type. Without exact type information, the compiler must emit slower generalized machine code that can deal with all potential type combinations. While compile-time static type inference might be able to gather type information to generate optimized machine code, traditional static analysis is very expensive and hence not well suited for the highly interactive environment of a web browser.
We present a trace-based compilation technique for dynamic languages that reconciles speed of compilation with excellent performance of the generated machine code. Our system uses a mixed-mode execution approach: the system starts running JavaScript in a fast-starting bytecode interpreter. As the program runs, the system
Subject
Property
Object
Running vocabulary suggestions via typeahead library
Loading and highlighting annotations
Add annotation
Show local sparql data
Find Similar
Storing of triples in Virtuoso
http://eis.iai.uni-bonn.de/semann/pdf/Ontological%20User%20Profiling.pdf
http://eis.iai.uni-bonn.de/semann/pdf/DBLP.pdf
Naive similar publications search based on existing annotations

Result


Page: 1 of 2
Automatic Zoom
easily understand it, especially with the guidance of a teacher or a written guide. Using this simple program as a basis, computer science principles or elements of a specific programming language can be explained to novice programmers. Experienced programmers learning new languages can also gain a lot of information about a given language's syntax and structure from a "Hello, world!" program.
In addition, "Hello, world!" can be a useful sanity test to make sure that a language's compiler, development environment, and run-time environment are correctly installed. Configuring a complete programming toolchain from scratch to the point where even trivial programs can be compiled and run can involve substantial amounts of work. For this reason, a simple program is used first when testing a new tool chain.
Sample table 2 (complex version)
Selected table to annotate
Name
Summary
Rows
Gsod sipa
Samples from US weather stations since 1929
115 Million
Mlab abc
Measurement data of broadband connection performance
240 Billion
Natality sky
Birth information for the United States from 1969 to 2008
68 Million
Shakes peare
Word index for works of Shakespeare
164 Kilo
Wiki pedia
Revision information for Wikipedia articles
314 Million
Show Simple Annotate Panel
Confirm annotation
Reset Selection
Confirm annotaiotn
reset the selection
Name
Summary
Rows
Gsod sipa
Samples from US weather stations since 1929
115 Million
Mlab abc
Measurement data of broadband connection performance
240 Billion
Natality sky
Birth information for the United States from 1969 to 2008
68 Million
Shakes peare
Word index for works of Shakespeare
164 Kilo
Wiki pedia
Revision information for Wikipedia articles
314 Million
Extracted table information from pdf

Unit Test Result



Demo

Issues & limitations (Demo)

Implemented Features

- Extend the implementation from text to table annotation
- Suggested table selection mechanism
- Accommodate the UI for new implementation
- Add a application set up script when the application run first time
- Extend our wiki doc to support Mac and Ubunty installation.

Lessons learn

- PDF.js is not friendly at all
- Get more support from Virtuso (**CORS**).
- JavaScript test unit (**Qunit**) is not simple unit test its quite different.
- Handling a **team** is very different/difficult in University and real world professional environment.
- Making the **Poster** was a interesting experience.

Future work

- Will continue as part of thesis work
- We addressed some issue here, will continue worked on that
- Put it in web for public use.
- Get the user feedback and take under consideration of constructive suggestion.

Thanks for your time

Q & A.....