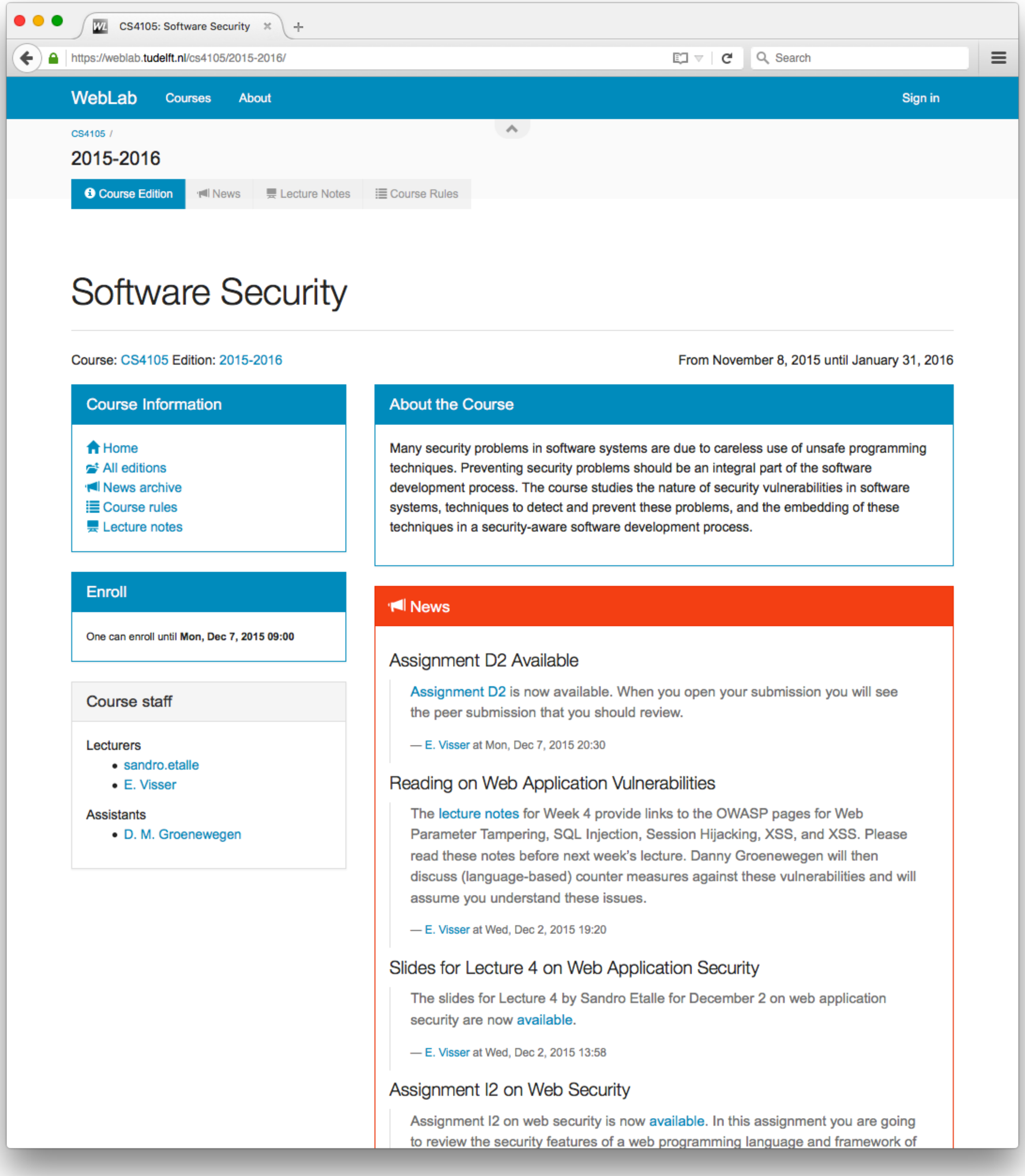
Guest lecture CS4105 Software Security

# Language-based Security in WebDSL

Danny Groenewegen



2015-2016

# Software Security

Course: CS4105 Edition: 2015-2016

From November 8, 2015 until January 31, 2016

## Course Information

- Home
- All editions
- News archive
- Course rules
- Lecture notes

## Enroll

One can enroll until **Mon, Dec 7, 2015 09:00**

## Course staff

Lecturers

- sandro.etalle
- E. Visser

Assistants

- D. M. Groenewegen

## About the Course

Many security problems in software systems are due to careless use of unsafe programming techniques. Preventing security problems should be an integral part of the software development process. The course studies the nature of security vulnerabilities in software systems, techniques to detect and prevent these problems, and the embedding of these techniques in a security-aware software development process.

## News

### Assignment D2 Available

> Assignment D2 is now available. When you open your submission you will see the peer submission that you should review.
>
> — E. Visser at Mon, Dec 7, 2015 20:30

### Reading on Web Application Vulnerabilities

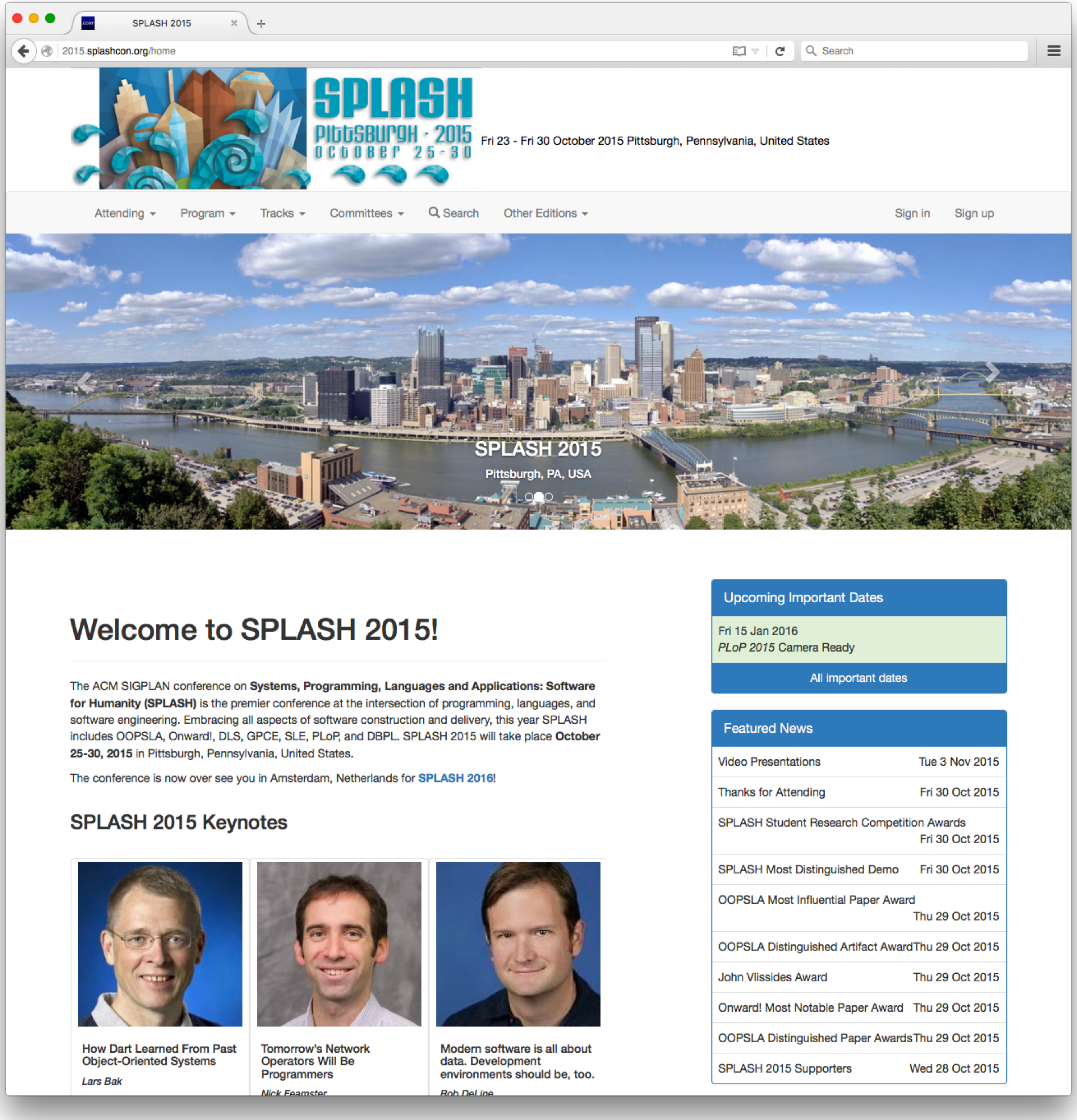
> The lecture notes for Week 4 provide links to the OWASP pages for Web Parameter Tampering, SQL Injection, Session Hijacking, XSS, and XSS. Please read these notes before next week's lecture. Danny Groenewegen will then discuss (language-based) counter measures against these vulnerabilities and will assume you understand these issues.
>
> — E. Visser at Wed, Dec 2, 2015 19:20

### Slides for Lecture 4 on Web Application Security

> The slides for Lecture 4 by Sandro Etalle for December 2 on web application security are now available.
>
> — E. Visser at Wed, Dec 2, 2015 13:58

### Assignment I2 on Web Security

> Assignment I2 on web security is now available. In this assignment you are going to review the security features of a web programming language and framework of



Fri 23 - Fri 30 October 2015 Pittsburgh, Pennsylvania, United States

# Welcome to SPLASH 2015!

The ACM SIGPLAN conference on **Systems, Programming, Languages and Applications: Software for Humanity (SPLASH)** is the premier conference at the intersection of programming, languages, and software engineering. Embracing all aspects of software construction and delivery, this year SPLASH includes OOPSLA, Onward!, DLS, GPCE, SLE, PLoP, and DBPL. SPLASH 2015 will take place **October 25-30, 2015** in Pittsburgh, Pennsylvania, United States.

The conference is now over see you in Amsterdam, Netherlands for **SPLASH 2016**!

## SPLASH 2015 Keynotes

How Dart Learned From Past Object-Oriented Systems

*Lars Bak*

Tomorrow's Network Operators Will Be Programmers

*Nick Feamster*

Modern software is all about data. Development environments should be, too.

*Rob DeLine*

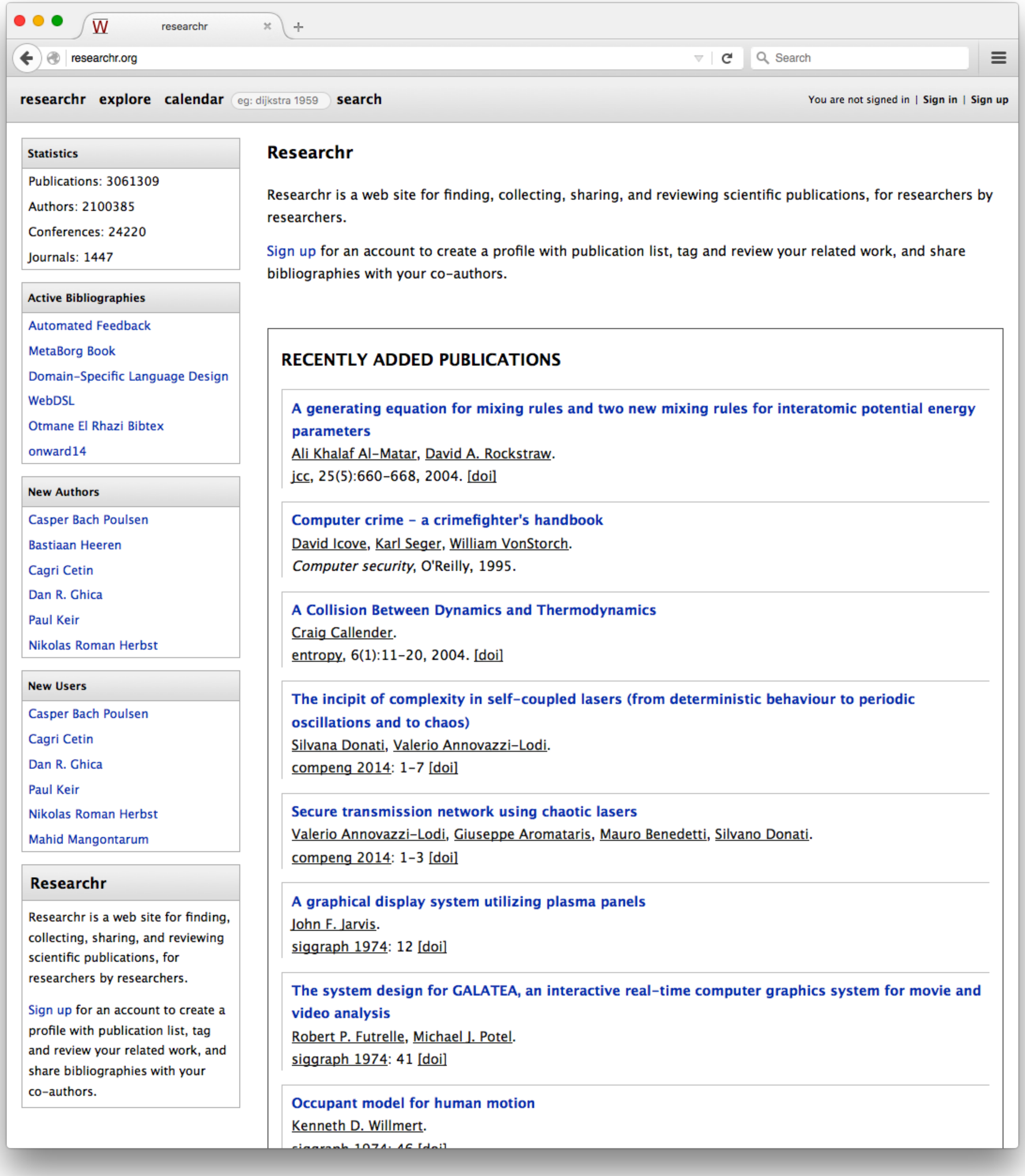
### Upcoming Important Dates

Fri 15 Jan 2016
*PLoP 2015* Camera Ready

All important dates

### Featured News

| | |
|---|---|
| Video Presentations | Tue 3 Nov 2015 |
| Thanks for Attending | Fri 30 Oct 2015 |
| SPLASH Student Research Competition Awards | Fri 30 Oct 2015 |
| SPLASH Most Distinguished Demo | Fri 30 Oct 2015 |
| OOPSLA Most Influential Paper Award | Thu 29 Oct 2015 |
| OOPSLA Distinguished Artifact Award | Thu 29 Oct 2015 |
| John Vlissides Award | Thu 29 Oct 2015 |
| Onward! Most Notable Paper Award | Thu 29 Oct 2015 |
| OOPSLA Distinguished Paper Awards | Thu 29 Oct 2015 |
| SPLASH 2015 Supporters | Wed 28 Oct 2015 |



researchr explore calendar eg: dijkstra 1959 search

You are not signed in | Sign in | Sign up

**Statistics**

Publications: 3061309

Authors: 2100385

Conferences: 24220

Journals: 1447

**Active Bibliographies**

- Automated Feedback
- MetaBorg Book
- Domain-Specific Language Design
- WebDSL
- Otmane El Rhazi Bibtex
- onward14

**New Authors**

- Casper Bach Poulsen
- Bastiaan Heeren
- Cagri Cetin
- Dan R. Ghica
- Paul Keir
- Nikolas Roman Herbst

**New Users**

- Casper Bach Poulsen
- Cagri Cetin
- Dan R. Ghica
- Paul Keir
- Nikolas Roman Herbst
- Mahid Mangontarum

**Researchr**

Researchr is a web site for finding, collecting, sharing, and reviewing scientific publications, for researchers by researchers.

Sign up for an account to create a profile with publication list, tag and review your related work, and share bibliographies with your co-authors.

## Researchr

Researchr is a web site for finding, collecting, sharing, and reviewing scientific publications, for researchers by researchers.

Sign up for an account to create a profile with publication list, tag and review your related work, and share bibliographies with your co-authors.

### RECENTLY ADDED PUBLICATIONS

**A generating equation for mixing rules and two new mixing rules for interatomic potential energy parameters**
Ali Khalaf Al-Matar, David A. Rockstraw.
jcc, 25(5):660-668, 2004. [doi]

**Computer crime - a crimefighter's handbook**
David Icove, Karl Seger, William VonStorch.
*Computer security*, O'Reilly, 1995.

**A Collision Between Dynamics and Thermodynamics**
Craig Callender.
entropy, 6(1):11-20, 2004. [doi]

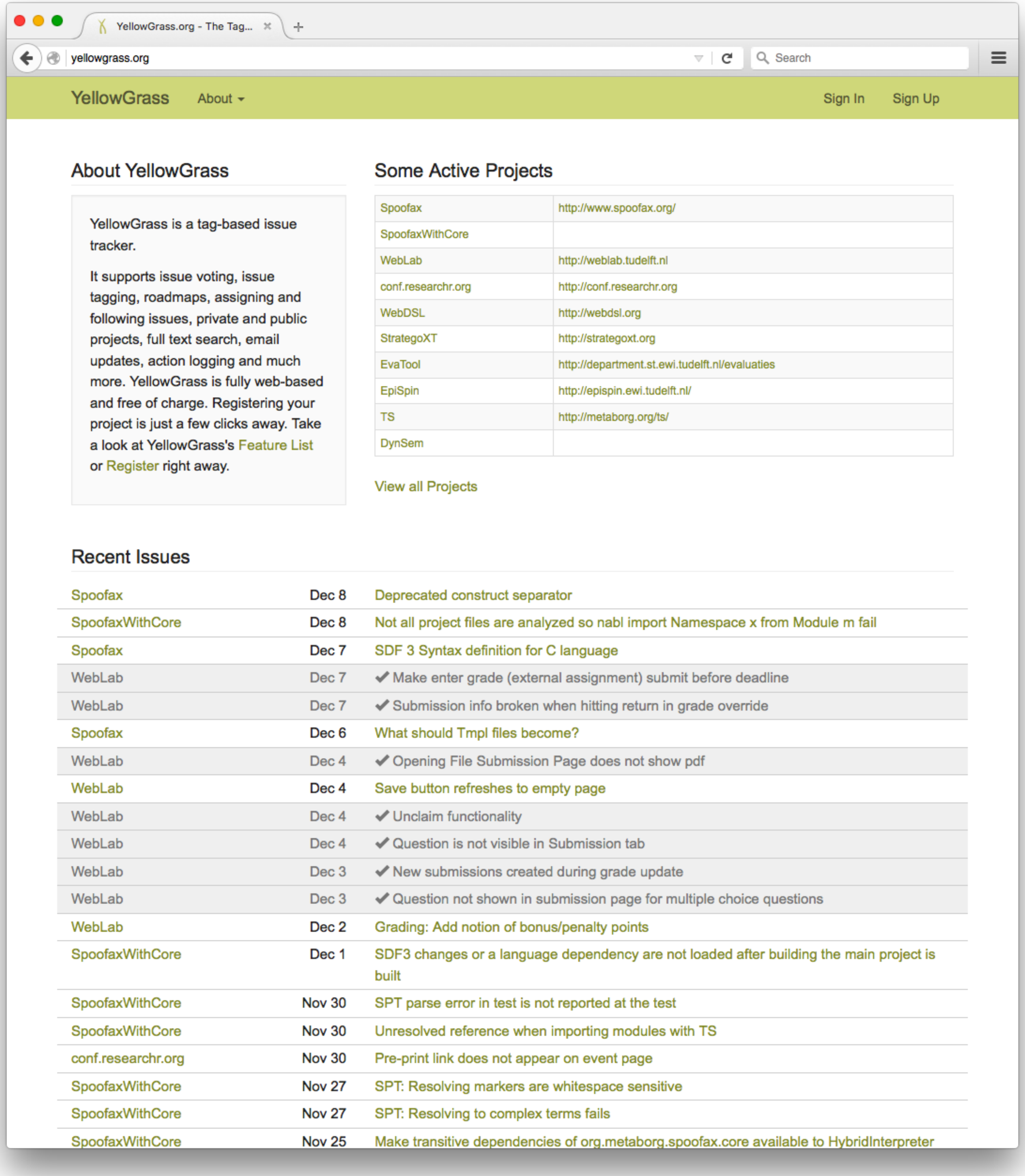
**The incipit of complexity in self-coupled lasers (from deterministic behaviour to periodic oscillations and to chaos)**
Silvana Donati, Valerio Annovazzi-Lodi.
compeng 2014: 1-7 [doi]

**Secure transmission network using chaotic lasers**
Valerio Annovazzi-Lodi, Giuseppe Aromataris, Mauro Benedetti, Silvano Donati.
compeng 2014: 1-3 [doi]

**A graphical display system utilizing plasma panels**
John F. Jarvis.
siggraph 1974: 12 [doi]

**The system design for GALATEA, an interactive real-time computer graphics system for movie and video analysis**
Robert P. Futrelle, Michael J. Potel.
siggraph 1974: 41 [doi]

**Occupant model for human motion**
Kenneth D. Willmert.
siggraph 1974: 46 [doi]



## About YellowGrass

YellowGrass is a tag-based issue tracker.

It supports issue voting, issue tagging, roadmaps, assigning and following issues, private and public projects, full text search, email updates, action logging and much more. YellowGrass is fully web-based and free of charge. Registering your project is just a few clicks away. Take a look at YellowGrass's Feature List or Register right away.

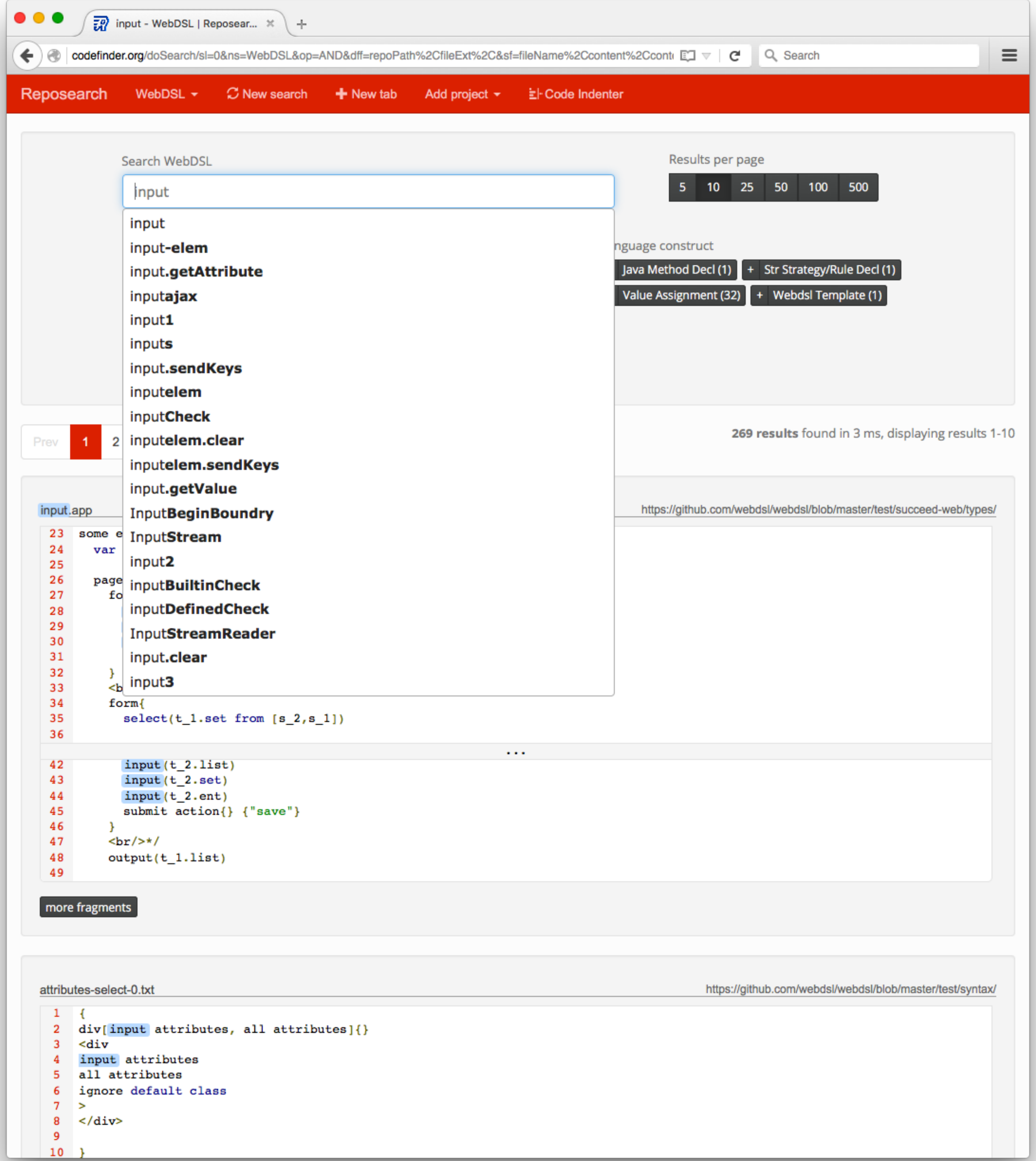
## Some Active Projects

| | |
|---|---|
| Spoofax | http://www.spoofax.org/ |
| SpoofaxWithCore | |
| WebLab | http://weblab.tudelft.nl |
| conf.researchr.org | http://conf.researchr.org |
| WebDSL | http://webdsl.org |
| StrategoXT | http://strategoxt.org |
| EvaTool | http://department.st.ewi.tudelft.nl/evaluaties |
| EpiSpin | http://epispin.ewi.tudelft.nl/ |
| TS | http://metaborg.org/ts/ |
| DynSem | |

View all Projects

## Recent Issues

| | | |
|---|---|---|
| Spoofax | Dec 8 | Deprecated construct separator |
| SpoofaxWithCore | Dec 8 | Not all project files are analyzed so nabl import Namespace x from Module m fail |
| Spoofax | Dec 7 | SDF 3 Syntax definition for C language |
| WebLab | Dec 7 | ✓ Make enter grade (external assignment) submit before deadline |
| WebLab | Dec 7 | ✓ Submission info broken when hitting return in grade override |
| Spoofax | Dec 6 | What should Tmpl files become? |
| WebLab | Dec 4 | ✓ Opening File Submission Page does not show pdf |
| WebLab | Dec 4 | Save button refreshes to empty page |
| WebLab | Dec 4 | ✓ Unclaim functionality |
| WebLab | Dec 4 | ✓ Question is not visible in Submission tab |
| WebLab | Dec 3 | ✓ New submissions created during grade update |
| WebLab | Dec 3 | ✓ Question not shown in submission page for multiple choice questions |
| WebLab | Dec 2 | Grading: Add notion of bonus/penalty points |
| SpoofaxWithCore | Dec 1 | SDF3 changes or a language dependency are not loaded after building the main project is built |
| SpoofaxWithCore | Nov 30 | SPT parse error in test is not reported at the test |
| SpoofaxWithCore | Nov 30 | Unresolved reference when importing modules with TS |
| conf.researchr.org | Nov 30 | Pre-print link does not appear on event page |
| SpoofaxWithCore | Nov 27 | SPT: Resolving markers are whitespace sensitive |
| SpoofaxWithCore | Nov 27 | SPT: Resolving to complex terms fails |
| SpoofaxWithCore | Nov 25 | Make transitive dependencies of org.metaborg.spoofax.core available to HybridInterpreter |



input

input
input-elem
input.getAttribute
inputajax
input1
inputs
input.sendKeys
inputelem
inputCheck
inputelem.clear
inputelem.sendKeys
input.getValue
InputBeginBoundry
InputStream
input2
inputBuiltinCheck
inputDefinedCheck
InputStreamReader
input.clear
input3

**269 results** found in 3 ms, displaying results 1-10

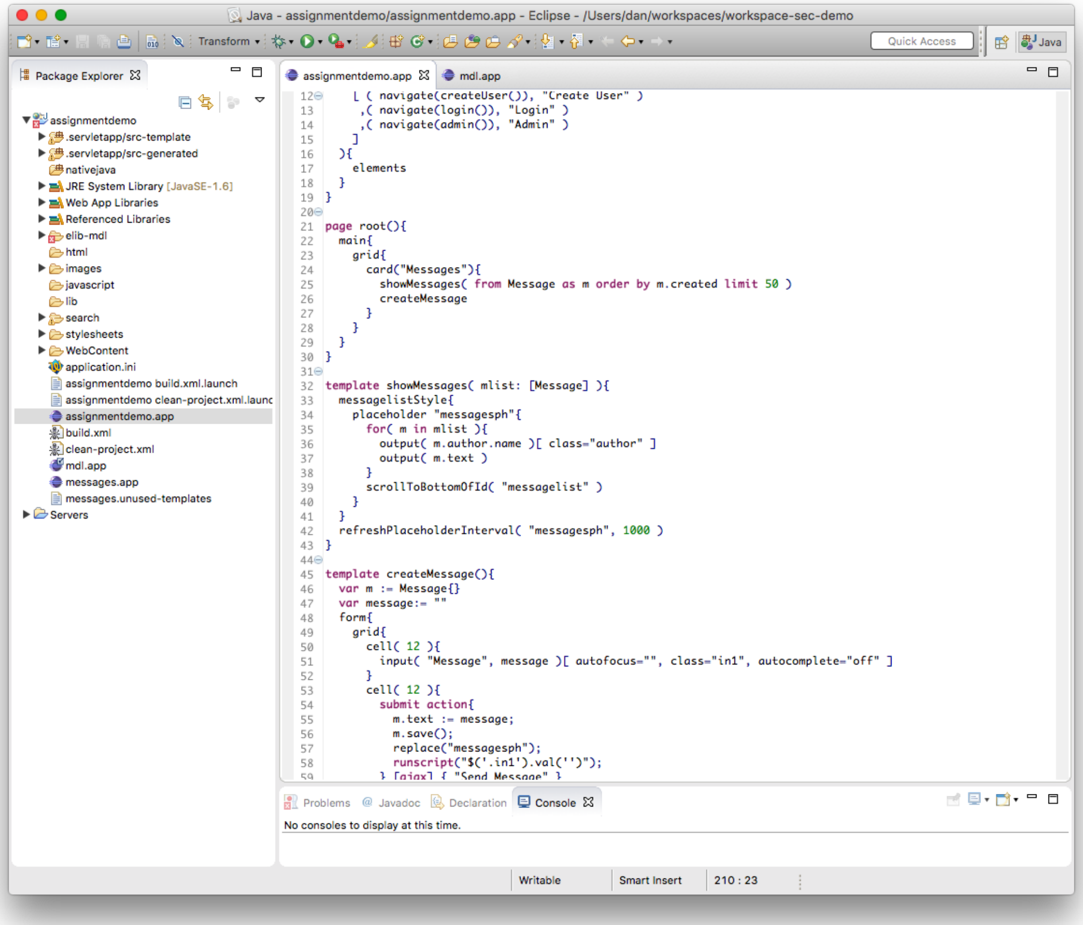
input.app https://github.com/webdsl/webdsl/blob/master/test/succeed-web/types/

```
some e
  var

  page
    fo




    }
    <b
    form{
      select(t_1.set from [s_2,s_1])

...
      input(t_2.list)
      input(t_2.set)
      input(t_2.ent)
      submit action{} {"save"}
    }
    <br/>*/
    output(t_1.list)

```

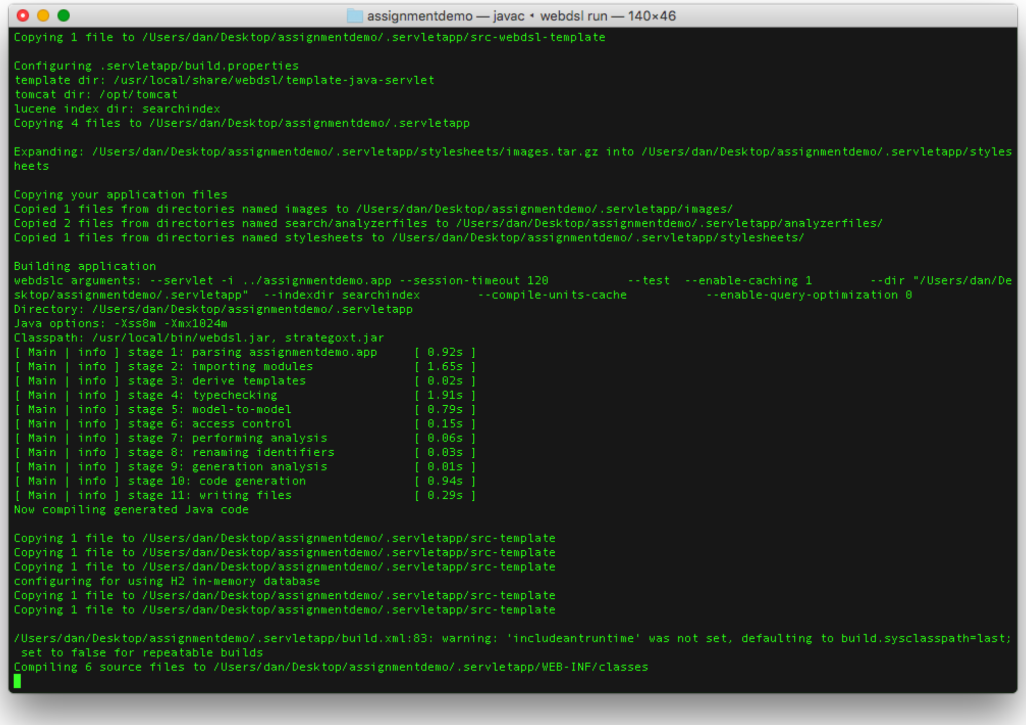
attributes-select-0.txt https://github.com/webdsl/webdsl/blob/master/test/syntax/

```
{
div[input attributes, all attributes]{}
<div
input attributes
all attributes
ignore default class
>
</div>

}
```

# WebDSL



```
    [ ( navigate(createUser()), "Create User" )
     ,( navigate(login()), "Login" )
     ,( navigate(admin()), "Admin" )
    ]
  ){
    elements
  }
}

page root(){
  main{
    grid{
      card("Messages"){
        showMessages( from Message as m order by m.created limit 50 )
        createMessage
      }
    }
  }
}

template showMessages( mlist: [Message] ){
  messagelistStyle{
    placeholder "messagesph"{
      for( m in mlist ){
        output( m.author.name )[ class="author" ]
        output( m.text )
      }
      scrollToBottomOfId( "messagelist" )
    }
  }
  refreshPlaceholderInterval( "messagesph", 1000 )
}

template createMessage(){
  var m := Message{}
  var message:= ""
  form{
    grid{
      cell( 12 ){
        input( "Message", message )[ autofocus="", class="in1", autocomplete="off" ]
      }
      cell( 12 ){
        submit action{
          m.text := message;
          m.save();
          replace("messagesph");
          runscript("$('.in1').val('')");
        } [ajax] { "Send Message" }
```



```
Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-webdsl-template

Configuring .servletapp/build.properties
template dir: /usr/local/share/webdsl/template-java-servlet
tomcat dir: /opt/tomcat
lucene index dir: searchindex
Copying 4 files to /Users/dan/Desktop/assignmentdemo/.servletapp

Expanding: /Users/dan/Desktop/assignmentdemo/.servletapp/stylesheets/images.tar.gz into /Users/dan/Desktop/assignmentdemo/.servletapp/styles
heets

Copying your application files
Copied 1 files from directories named images to /Users/dan/Desktop/assignmentdemo/.servletapp/images/
Copied 2 files from directories named search/analyzerfiles to /Users/dan/Desktop/assignmentdemo/.servletapp/analyzerfiles/
Copied 1 files from directories named stylesheets to /Users/dan/Desktop/assignmentdemo/.servletapp/stylesheets/

Building application
webdslc arguments: --servlet -i ../assignmentdemo.app --session-timeout 120           --test  --enable-caching 1        --dir "/Users/dan/De
sktop/assignmentdemo/.servletapp"  --indexdir searchindex        --compile-units-cache           --enable-query-optimization 0
Directory: /Users/dan/Desktop/assignmentdemo/.servletapp
Java options: -Xss8m -Xmx1024m
Classpath: /usr/local/bin/webdsl.jar, strategoxt.jar
[ Main | info ] stage 1: parsing assignmentdemo.app     [ 0.92s ]
[ Main | info ] stage 2: importing modules              [ 1.65s ]
[ Main | info ] stage 3: derive templates               [ 0.02s ]
[ Main | info ] stage 4: typechecking                   [ 1.91s ]
[ Main | info ] stage 5: model-to-model                 [ 0.79s ]
[ Main | info ] stage 6: access control                 [ 0.15s ]
[ Main | info ] stage 7: performing analysis            [ 0.06s ]
[ Main | info ] stage 8: renaming identifiers           [ 0.03s ]
[ Main | info ] stage 9: generation analysis            [ 0.01s ]
[ Main | info ] stage 10: code generation               [ 0.94s ]
[ Main | info ] stage 11: writing files                 [ 0.29s ]
Now compiling generated Java code

Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-template
Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-template
Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-template
configuring for using H2 in-memory database
Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-template
Copying 1 file to /Users/dan/Desktop/assignmentdemo/.servletapp/src-template

/Users/dan/Desktop/assignmentdemo/.servletapp/build.xml:83: warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last;
 set to false for repeatable builds
Compiling 6 source files to /Users/dan/Desktop/assignmentdemo/.servletapp/WEB-INF/classes
```

## IDE

- check, compile, run
- inline errors markers
- reference resolving
- content completion

## Command-line compiler

- check, compile, run

# Entity - Persisted Object

```
entity Message{
  author: User
  text: WikiText
}

entity User{
  name: String (id)
  password: Secret
  admin: Bool
}
```

- Compiler generates Java class with Hibernate ORM annotations for mapping to database tables

# Function Code

```
var m := Message{}
m.text := "test";
m.author := securityContext.principal;
m.save();
```

- Object-oriented
- Changes to persisted entity automatically saved
- New entities .save() or assign to a persisted entity property

# Page Definitions

```
page account(){
  main{
    grid{
      card( "Login" ){
        logintemplate
      }
      card( "Logout" ){
        logout
      }
    }
  }
}
```

```
page user( u: User ){
  "User page of "
  output( u.name )
}
```

- arguments loaded automatically from database
- rendering escapes HTML
- url is page name and arguments
- built-in page cache

# Template Definitions

```
template cell( i: Int ){
  div[  class = "mdl-cell mdl-cell--"+i+"-col"
      , all attributes ]{
    elements
  }
}

template maingridcard(title: String){
  main{
    grid{
      card(title){
        elements
      }
    }
  }
}
```

- reusable page fragments
- elements are the nested elements at the call

# Forms

```
template createMessage(){
  var m := Message{}
  form{
    input( m.text )
    submit action{
      m.author := securityContext.principal;
      m.save();
    } { "Send Message" }
  }
}
```

- not just rendering
- databind of inputs
- request is one database transaction

# Data Validation

- validate rules on entity property, in form, or in submit action
- if any validate fails the transaction is aborted and error message is rendered

```
template logintemplate() {
  var name: String
  var pass: Secret
  var stayLoggedIn := false
  form {
    input( "Name", name )
    input( "Password", pass )
    submit signinAction() { "Login" }
  }
  action signinAction() {
    validate(authenticate(name,pass),
             "The login credentials are not valid.");
    return root();
  }
}
```

# Access Control

```
principal is User with credentials name, password

access control rules

rule page root(){ true }
rule page createUser(){ true }
rule page account(){ true }
rule page user( u: User ){ u == principal }
rule page admin(){ principal.admin }
```

- principal refers to an entity and enables access control
- access control checks woven into pages by compiler
- default deny access (such as no rule for page)
- generates default authentication templates and functions

# Common Vulnerabilities

- Missing access control on URLs
- Request tampering
- SQL injection
- XSS
- CSRF

# Missing access control

Solution

- Access control as language feature
- Boilerplate code for ac checks generated by compiler
- Deny by default

Discussion

- Implementation errors avoided, but can still have design flaws. Policies can become complex.

# Request tampering

Solution

- Entity data and function code only server-side.
- Input name generation built-in (details on next slide)
- Inputs restricted by their type and other annotations (e.g. allowed=[e1,e2,e3] to specify limited options for dropdowns)
- Data validation rules for additional restrictions
- Form actions not a separate server entry point, goes through access control check of page.

# Input name generation

- Each template can access it's dynamic template id, this is used to create input names.
- Runtime generates template id based on control flow path.
- Form input and submit actions are checked to (still) be valid at the server.
- Example: the condition that caused a form to be available changed -> action fails.
- CSRF: add a secret and user-unique part to template id

Discussion

- Custom input components could be implemented incorrectly, e.g. instead of using id having a fixed name

# SQL Injection

HQL Example:

```
session.createQuery(
"select g from Message as g where g.text= :param0" )
.setParameter("param0", prop).list();
```

```
session.createQuery(
  "FROM accounts WHERE custID='" +
  request.getParameter("id") +
  "'");
```

- API helps but still allows incorrect usage

# SQL Injection

Solution

- Queries part of language syntax, generate code with correct API usage
- Hide access to lower level query execution.

```
page user( u: User ){
  maingridcard("User page for " + u.name){
    showMessages( from Message as m
                  where m.author = ~u
                  order by m.created )
  }
}
```

Discussion

- Can still get access with extensibility features, calling custom Java code.

# XSS

Solution

- Generated rendering code applies escaping

Discussion

- Custom libraries and javascript code with XSS issues
- Valid usage of unescaped data, e.g. trusted user in a CMS creates page content that embeds a slideshare widget.

# Conclusion

- A better programming language can do a lot to avoid security vulnerabilities in web programming.
- Trade-off expressivity and security.
- In practice, applications often require integration with other code such as invoking web services, invoking libraries, and including javascript widgets.
- Language cannot do it all, security depends on whole stack, not just the application code.
- Deployment requires knowledge of platform

# Assignment I-2 Demo