

JAWS: A Javascript API for the Efficient Testing and Integration of Semantic Web Services

David A. Ostrowski

System Analytics and Environmental Sciences

Research and Advanced Engineering

Ford Motor



Research & Advanced Engineering

Overview

- Introduction (Focus / Overview)
- Review environment
- Present major constructs
- Example - Use Case
- Conclusion



Introduction

- Semantic Web Services (SWS)
 - High Potential
- Leverage current software
- Decouple process from application development
 - Javascript API
 - Work collaboration (controlled)



Goals

- Web 2.0
- Integration
- Deployment (Compatibility)
- Protection
- Simplicity



SWS Environment

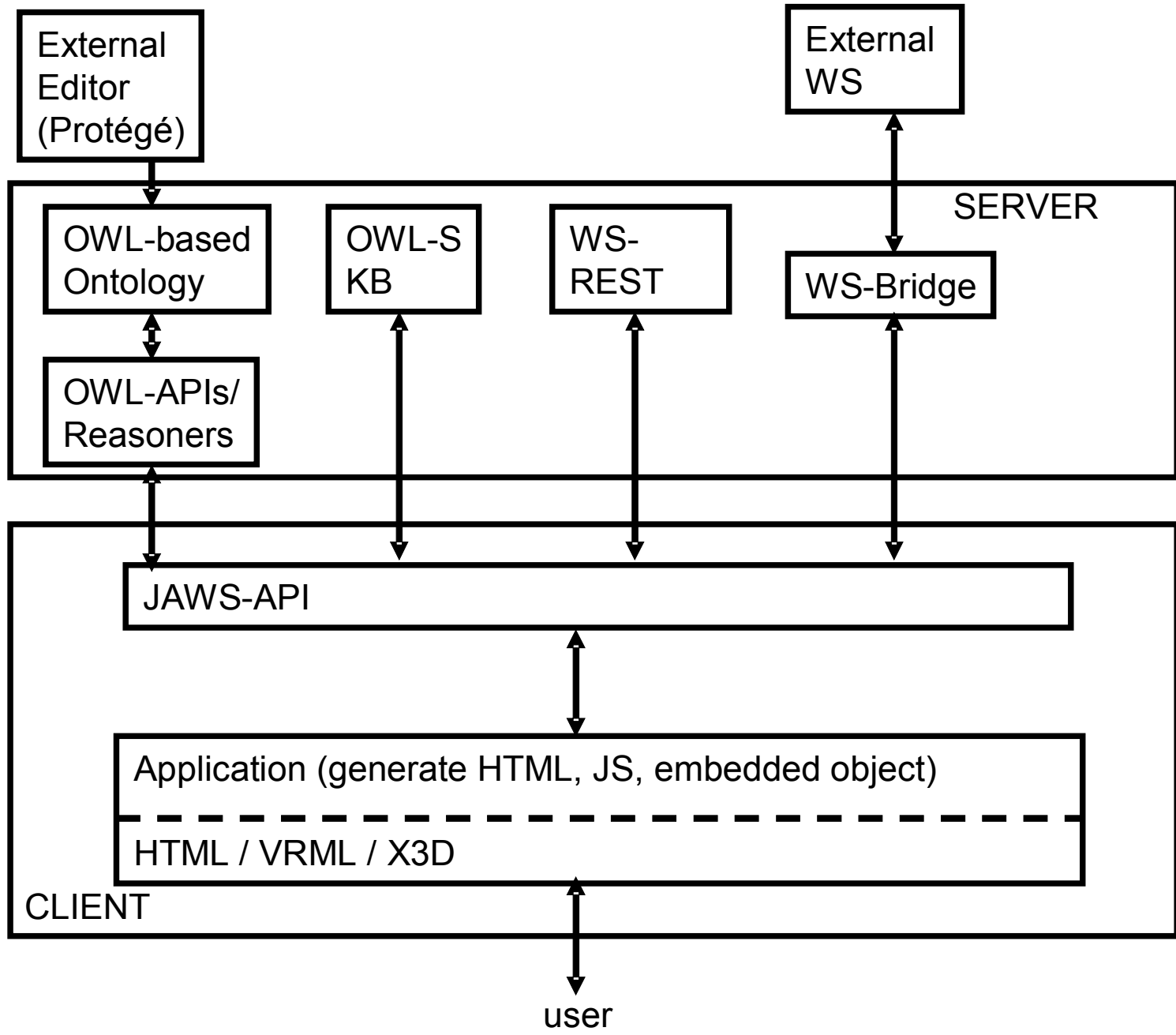
- Two separate knowledge bases
 - The first KB supports description of application domain
 - Second maintains semantic-based definition of web services
- Support
 - interaction data source(s)
 - web services



Technologies

- AJAX
 - Redefining attribute of JS
- REST
 - Natural extension to (relatively) uncomplicated networking of XHR
- Dynamic Typing
 - Scripting language feature
 - More important in semantic web since data is not anticipated





Advantages of Javascript

- Ubiquitous to Web
- Complementary nature
- Ease of use (scripting lang att.)
- Everyone knows it (or at least has used it to some degree)
- More flexibility than advertised



Challenges

- Differences between O-O and RDF
 - beyond OWL-S represents a substantial challenge
- Large Data Stores
 - Not our problem - yet
- Cross-Domain
 - Bridge with process (inefficient but easy)
 - Migrate what we can to same server (perfect world)



API (3 Categories)

- Applied to OWL Taxonomy
- OWL-S
- Invocation WS



Applied to OWL Taxonomy

- ***setRes()***

- returns an object providing methods to support the loading and subsequent discovery of SWS resources
- Two main input arguments are adapter and host (defaults are regular OWL/RDF format and localhost)
- ex. `var c = setRes(adapter, host)`



OWL Taxonomy

- ***getRes()***
 - Loads SWS resources (.OWL) or makes available (large scale data resources)
 - Ex: *c.getRes(ResourceName)*
 - onError handling performed by *size()*
 - Ex: *c.size()*



OWL taxonomy

- Discovery
 - Simple location to expert reasoning

`currentRes.find(keyword_or_expression)`

`currentRes.query(format, queryString)`

`currentRes.reasoning(format, expression)`



OWL-S data stores

- ***setOWLS()***

- Mapping of all four OWL-S based class definitions.
- Accepts a URI for the services description class file
- returns a Javascript object representing the data
- The object, referenced by hashing in one of the class names
- ***ex: var s = setOWLS("uri")***



Web Service Invocation

- WS contains all web services ref by name (associative array) and then its methods
- WS("SWS_name").refMethods()
- WS("SWS_name").refMethods_CB()



Case Study

- Demonstrate use through the development of an numerical matchmaking app.
- Discovery, dynamic testing and invocation of mathematically based web services.



Case Study

- `Var r = setres(
http://srlxpm3q7h41.srl.ford.com) // no
adapter necessary`
- `r.getRes("mathOnto.owl")`
- `r.size()`
- `var arr = r.find("optimization")`



Case Study

- ```
for(i = 0;i = arr.length();i++){
 document.write(arr[i].className,
 arr[i].childName, arr[i].uriProperty)
}

var s = setOWLs(arr[0].URIproperty)
```



# Case Study

- When a class is identified of interest, then its properties can be located via a method call
- `Var s = setOWLs(arr[0].URIproperty) LP_solver and mixed_mixed_int_opt` would be referenced as:
- ***output = WS("LP\_solver").mixed\_int\_opt(params);***



# USE CASE Scenario

- Use the API in a manufacturing simulation application
- This software allows for the design of manufacturing workstation layouts.
- We are interested in using a service to generate a suggested optimal design

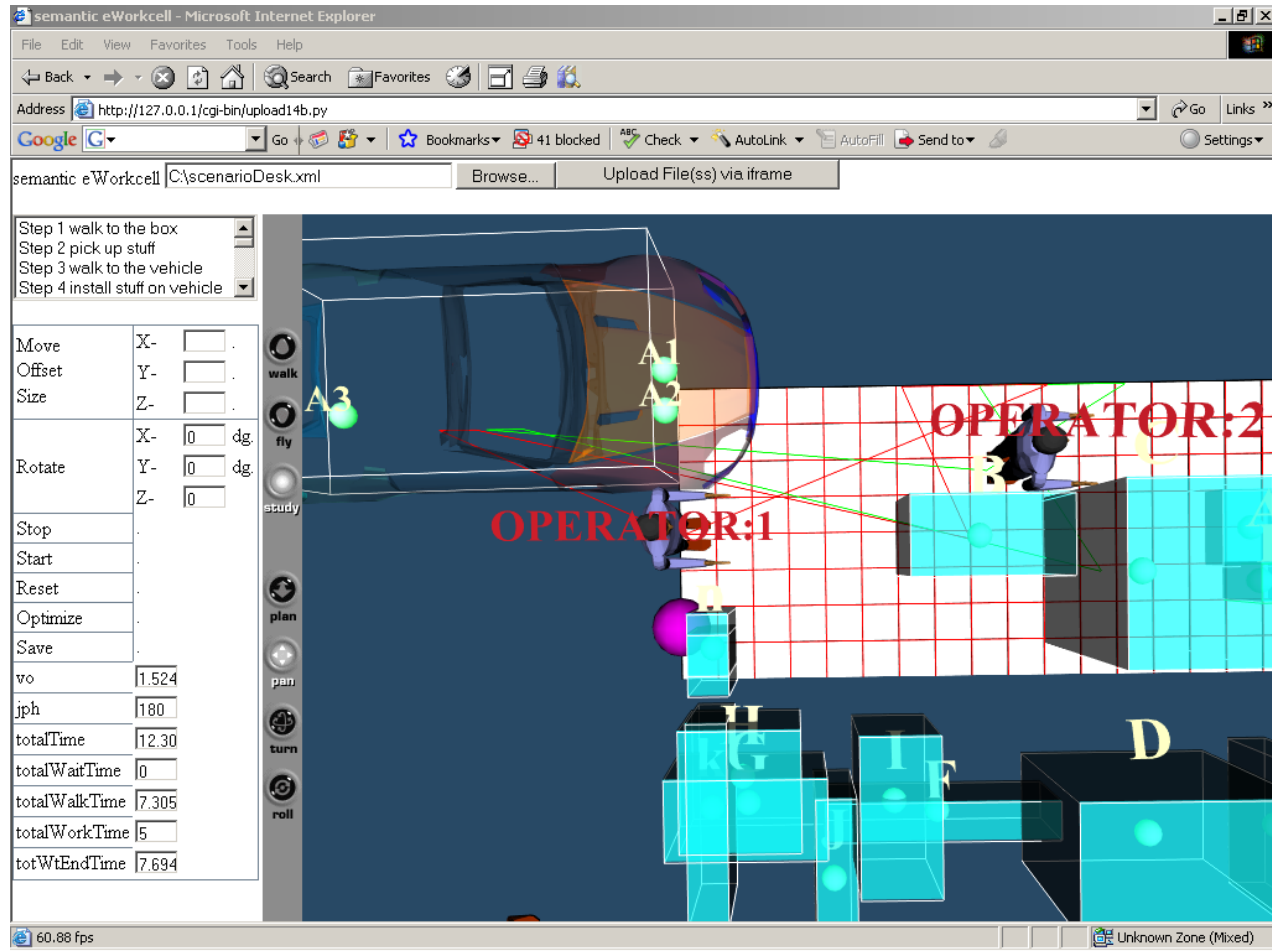


# Use Case Scenario

- Here, we incorporated our API to support semi-automatic process of SWS discovery and integration
- Step One: enter a keyword for ontology search
- Step Two: select among the recommended alternatives
- Step Three: run service with demo data
- Step Four: incorporate our service in app



# e-Workcell Interface



# e-Workcell Interface (w/ use of API)

semantic eWorkcell - Microsoft Internet Explorer

Address: http://localhost/cgi-bin/sework4b\_SW5demo2.py

Semantic eWorkcell | Browse... | Upload File(ss) via iframe

optimize | find | demo | demo2

|        |    |                                |     |
|--------|----|--------------------------------|-----|
| Move   | X- | <input type="checkbox"/>       | .   |
| Offset | Y- | <input type="checkbox"/>       | .   |
| Size   | Z- | <input type="checkbox"/>       | .   |
| Rotate | X- | <input type="text" value="0"/> | dg. |
|        | Y- | <input type="text" value="0"/> | dg. |
|        | Z- | <input type="text" value="0"/> |     |

Read File  
Stop  
Start  
Reset  
Optimize  
Save

|               |       |
|---------------|-------|
| vo            | 1.047 |
| jph           | 80    |
| totalTime     | 43.91 |
| totalWaitTime | 0.193 |
| totalWalkTime | 22.72 |
| totalWorkTime | 21    |
| totWtEndTime  | 1.082 |

```
kstart=271 10.42624629 10.42624629 0.00000000
10.42624629big_time = 33.42624629remainder =
23.00000000 NUOVO MINIMO A kstart=410 10.42624629
10.42624629 0.00000000 10.42624629 big_time =
33.42624629 remainder = 23.00000000
```

simplex - mead  
mixed-integer linear program

82.96 fps

Microsoft PowerPoint - [S... | semantic eWorkcell - ... | Command Prompt - pytho... | Cortona Console

11:34 AM



# Conclusion

- Presented a use of JS to decouple the process of SWS discovery, testing and integration
- Rapid development approach to incorporate the semantic web – now
- Support a means of integration of these technologies into industry



# Future Work

- Expand the ontology definition
- Incorporate more algorithms as SWS
- Incorporate demo data into each SWS in order to support this approach



# Future Work

- Develop this system into support of expert reasoning
- Utilize reasoning capabilities to support a higher integration into the enterprise



# Future Work

- Taking inspiration from XHR wrapping techniques applied in SAJAX (simple ajax) perhaps Javascript can be taken to an even higher level by configuring and regenerating script languages or even more strongly typed Java/ C# (good or bad?)



Thank you.

**VOLVO**



**Research & Advanced Engineering**

# 2007 Ford Shelby Cobra GT500

- 5.4 Litre (supercharged)
- Trans 6sp man.
- Horsepower 450+
- Torque 450+ (conservative est. ((they do not call it '500' for nothing)) )
- Fastest Mustang Ever.
- See me for customer discount.....Danke



# Q & A

**contact:  
dostrows@ford.com**

