



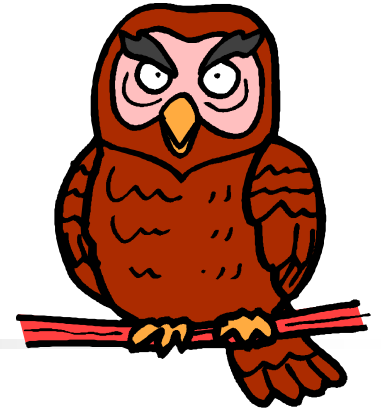
F-OWL: *An OWL Reasoner in Flora-2*

**Youyong Zou, Harry Chen,
Tim Finin, Lalana Kagal**

<http://fowl.sourceforge.net/>



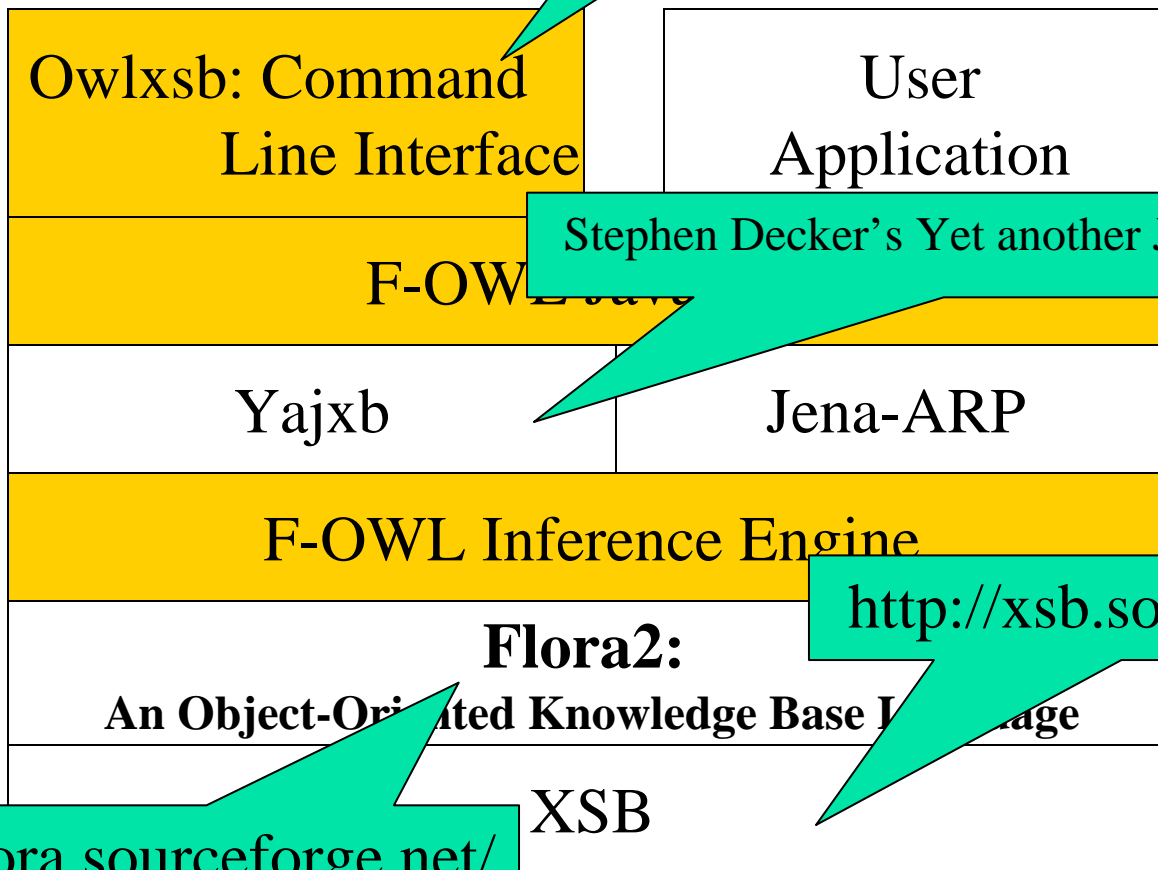
Feature



- Supports RDF and OWL-Full
- Supports RDF/N-Triple query
- Supports Dynamic Import
- Provides a Java API
- Tested with the RDF and OWL test cases

F-OWL Design

<http://fowl.sourceforge.net/>



Stephen Decker's Yet another Java-XSB Bridge

<http://xsb.sourceforge.net/>

<http://flora.sourceforge.net/>



F-OWL Inference Engine

- A set of rules that reason over the data model of RDF-S & OWL
 - Implemented in Flora-2
- A set of rules that map XML DataTypes into XSB terms
- A set of rules that perform ontology consistency checks
- A set of rules that “glue” together the upper Java API calls to the lower layer Flora-2/XSB rules



F-OWL Java API

- Core class: fowlEngine
 - Public method
 - init(): starts XSB engine
 - startFlora(): loads Flora-2 modules
 - startFOWL(): loads F-OWL modules
 - rule(): executes a specified rule
 - load(): imports OWL(RDF/XML) from a specified URI
 - query(): execute N-Triple query




owlxsb: F-OWL's CLI

- Users can
 - import OWL docs into the KB
 - modify stored triples
 - execute queries over the stored N-Triples
 - execute RDF/OWL test cases
 - manage Namespace mapping table
 - ... also access f-owl documentation

“I hate CLI” --> I want GUI

- Read & parse RDF/OWL Manifest files
- Fetch RDF/OWL test files at runtime
- Extensive log report



The screenshot shows a window titled "fowlTest" with two buttons: "All Testing..." and "Testing 1/210". The window is divided into three sections: "Test File", "Question?", and "Result".

Test File

```
-->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:about="http://example/q?abc=1&#38;def=2">
    <rdf:value>xxx</rdf:value>
  </rdf:Description>
</rdf:RDF>
```

Question?

```
# <http://www.w3.org/Consortium/Legal/copyright-software.html>
#
# Author: Dan Connolly
#
# $Id: test001.nt,v 1.2 2001/09/06 21:07:19 barstow Exp $
#
#####
<http://example/q?abc=1&def=2> <http://www.w3.org/1999/02/22
```

Result

```
Test Type: test.PositiveParserTest
Test Success!
Positive!

Test Time: 2131 mSeconds
```

F-OWL on Sourceforge

http://fowl.sourceforge.net/

F-OWL is an open source project hosted on sourceforge

The screenshot shows a Netscape browser window displaying the SourceForge project page for F-OWL. The browser's address bar shows the URL <http://sourceforge.net/projects/fowl/>. The page features a navigation menu with links for 'my sf.net', 'software map', 'foundries', 'about sf.net', and 'My Favorites'. The main content area is titled 'Project: F-OWL: Summary' and includes a search bar, a list of navigation links (Summary, Admin, Home Page, Forums, Tracker, Bugs, Support, Patches, RFE, Lists, Tasks, Docs, CVS, Files), and a description of the project. The description states that F-OWL is implemented in Flora-2/XSB and is an inference engine for the standard Web Ontology Language (OWL). It also lists the project's development status, environment, intended audience, license, natural language, operating system, programming language, and topic. A table of 'Latest File Releases' is provided at the bottom, showing two releases: 'fowl' and 'fowl-lib', both dated September 24, 2003. The left sidebar contains 'SF.net Resources' and a 'Most Active' list. The right sidebar features a 'Developer Info' section with project administrators and developers listed.

Project: F-OWL: Summary

Summary | Admin | Home Page | Forums | Tracker | Bugs | Support | Patches | RFE | Lists | Tasks | Docs | CVS | Files |

Implemented in Flora-2/XSB, F-OWL is an inference engine for the standard Web Ontology Language OWL. A set of Java API is provided for applications to assert and query OWL ontology statements.

Developer Info

Project Admins:
hchen1
zouyyong

Developers:
3 [View Members]

Latest File Releases

Package	Version	Date	Notes / Monitor	Download
fowl	fowl-0.41	September 24, 2003		Download
fowl-lib	fowl-lib-0.41	September 24, 2003		Download

F-OWL tests “Test Cases”



- Use of OWL Namespace: *Check OWL Syntax*
- Entailment Tests: *Load Premises file, Prove Conclusion file is TRUE*
- Non-Entailment Tests: *Load Premises file, Prove Conclusion file is FALSE*
- OWL for OWL Tests: *Prove conclusion is TRUE*
- Consistency Tests: *Load the file, Run consistency checker, Prove the file can be satisfied.*
- Inconsistency Tests: *Prove the file cannot be satisfied via consistency checker.*
- Import Entailment/Level Tests: *load imported ontology, check for import loop!*



Test Results

- Published at the W3C web site

<http://www.w3.org/2003/08/owl-systems/test-results-out.html>

- F-OWL fails on:

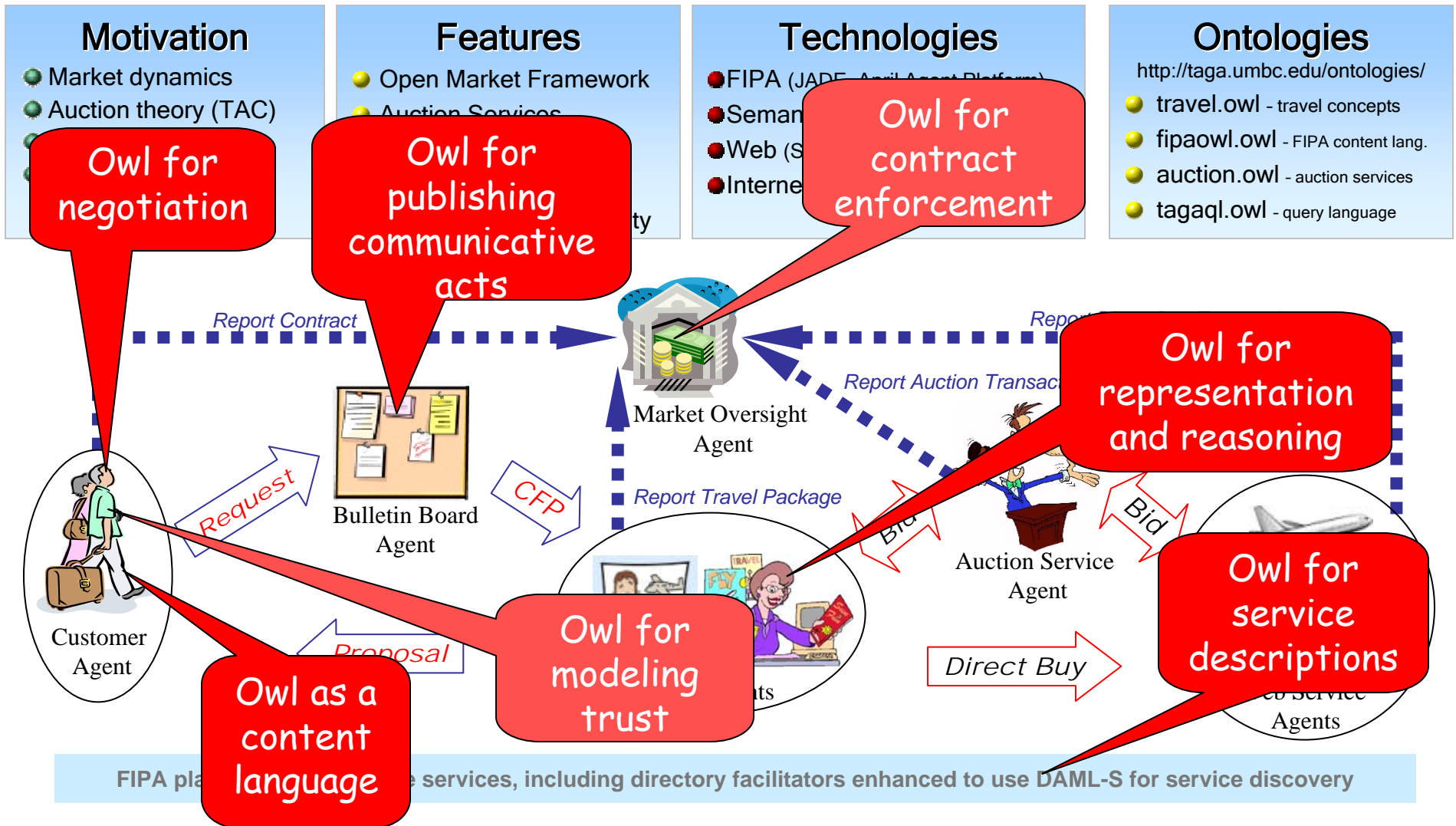
- Some large test files (e.g., food and wine)
- OWL DL Files with Complex XSD datatype:
for example: “invR-N-times-M-to-1”



Uses...

- F-OWL is / will be used in four Ph.D. dissertations at UMBC
 - Pervasive computing (Harry Chen)
 - Policy language (Lalana Kagal)
 - OWL+Bayesian reasoning (Zhongli Ding)
 - Multiagent systems (Youyong Zou)
- Motivation: all need an OWL reasoner as part of a hybrid reasoning system
- Prolog provides a good host environment

Travel Agent Game in Agentcities



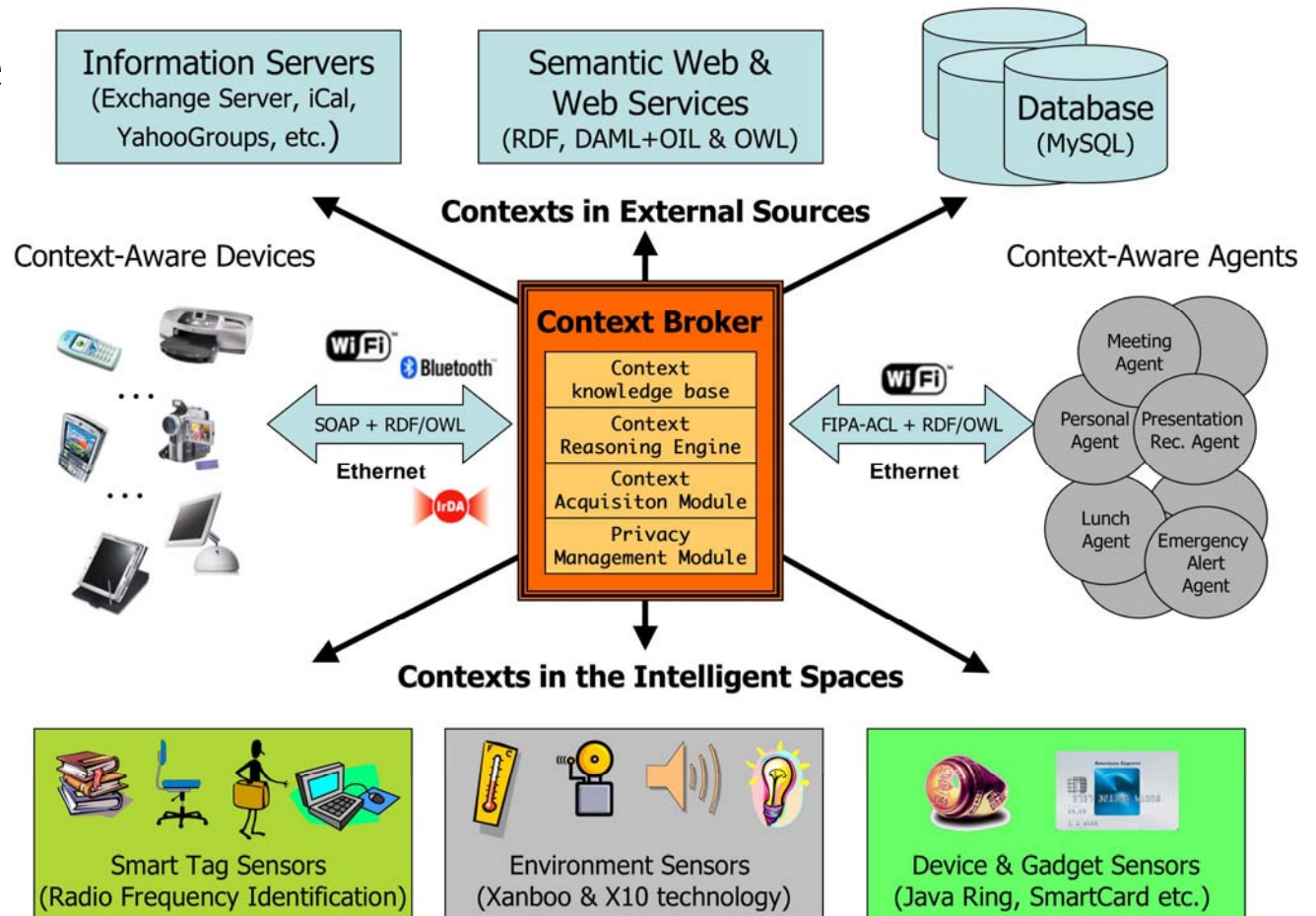
礼 Rei Policy Language

- A declarative policy language for describing policies over actions
- Based on deontic concepts (permission, prohibitions, obligations and dispensations) and speech acts (delegate, request, etc.)
- Represented in RDF and OWL
- Current policy reasoner in XSB with a custom policy IDE in Eclipse

UMBC Cobra

OWL usage

- Ontologies
- Content language
- F-OWL reasoner
- REI policy language
- DAML-Time components



<http://cobra.umbc.edu/>

Bayes OWL

- Probabilistic extension of OWL to support uncertain ontology representation and reasoning
- Approaches
 - Extend OWL for probabilistic annotation
 - Translate OWL ontology to Bayesian network (OWL-BN)
 - Probabilistic mappings between individual OWL-BN
 - Treat ontology reasoning (within and across ontologies) as Bayesian inference
- Plan for next year
 - Preliminary investigation on probabilistic concept mapping between OWL-BNs

Bayes OWL

- Probabilistic extension of OWL to support uncertain ontology representation and reasoning
- Approaches
 - Extend OWL for probabilistic annotation
 - Translate OWL ontology to Bayesian network (OWL-BN)
 - Probabilistic mappings between individual OWL-BNs
 - Treat ontology reasoning (within and across ontologies) as Bayesian inference



Future Work



- Better Support XSD data types
- Include additional rules to reasoning over OWL-Full
- Provide Web-base GUI to F-OWL
- Support additional query languages
 - RDQL/RQL
- Storing triples in MySQL
- Optimizations