Tools for DAML-Based Services, Document Templates, and Query Answering

Knowledge Systems, Artificial Intelligence Lab (KSL)
Stanford University

Investigators: Deborah McGuinness, Richard Fikes, Sheila McIlraith, Paulo Pinheiro da Silva,

November 2004
KSL DAML Language, Services, and Tools: Overall Program Summary

Language:
- Problem: Markup Languages lack representational power, precise semantics, & predictable inference tools
- Solution: Design (DAML-ONT, DAML+OIL, OWL), (DAML-S, OWL-S, SWSL), OWL-QL, SWRL; Form Joint Committee, WebOnt, SWSA, Semantic Web Best Practices, to design, integrate, and disseminate
- Research Approach: Leverage Web languages and Description Logics to generate OWL; leverage DAML/OWL for services ontologies and provide tools to use them; leverage query languages, description logics, and state of the art web languages to provide query and rule languages compatible with emerging standards.

Services:
- Problem: Provide language and reasoning tools for Web Service discovery, invocation, composition, & interoperation
- Solution: Proposed the Semantic Web Service vision, seminal paper with 200+ citations to date. **Tools:** Automated reasoning tools for Web Services: Golog tool for DAML WS composition; Petri-Net tool for WS analysis, simulation and verification; SDS semantic discovery, translation & explanation tool, adding OWL-S to BPEL4WS.
- Research Approach: Interleaved development of ontologies and tools, influencing & informed by industry standards.

Tools:
- **Explanation/Inference Web:** Applications determine answers from web input but may provide users (humans and agents) with little information defending their answers thus not giving users information that would justify trust. Our work provides a design and infrastructure for knowledge provenance, proofs, explanations, and trust and is used in DARPA’s PAL, ARDA’s NIMD, Trento’s JSAT, SRI’s SNARK, etc.
- **Hybrid Reasoning/JTP:** Many applications do not provide effective deductive question answering from semantic web information services. JTP provides a hybrid reasoning infrastructure for deductive question answering using powerful reasoners, web resources, and rich ontological information. Used by PAL, NIMD, AQUAINT, Ultralog, …
- **Question Answering/DQL & OWL-QL:** Semantic Web applications may require powerful and tunable query languages to enable customizable and accurate question answering. DQL and OWL-QL provide a question answering language and our client/server implementation is in use in PAL and NIMD and is a candidate standard query language for the semantic web.
**Markup Language**: Extend expressive power of existing web markup languages by:

- generating **DAML-ONT** (walkthru, spec, ex, daml-ont/oil, semantics)
- evolving to **DAML+OIL** (ref, model-theoretic semantics, axiomatic semantics, annotated markup, ex. Ontologies), submitting to W3C, outreach docs: Ontologies 101, Ontologies come of age, …
- evolving to W3C recommendation **OWL** (Overview, Guide, Reference, Semantics)

**Query & Rules Language**: Provide languages compatible with expressive markup languages supporting questions and rules by:

- generating **DQL** and evolving to **OWL-QL** from Joint Committee, input to DAWG
- generating **SWRL** from Joint Committee and submitting to W3C
**PROBLEM:** make network-accessible programs and devices interoperable, enabling automation of Web service (WS) discovery, invocation, composition, & interoperation.

**SOLUTION:** Interleaved development of:

- **an ontology** for describing Web services; (in collaboration w/ Coalition)
  - Development of the OWL-S (formerly DAML-S) ontology was iterative, influenced by several factors including: the evolution of the OWL (formerly DAML+OIL) language, evolving WS industry standards (including WSDL and BPEL4WS), program experience working with early versions of our ontology.
  - Generated DAML-S Coalition, OWL-S Coalition, SWSL, SWSA

- **SW Tools and automated reasoning** techniques that manipulate the ontology to perform automated WS discovery, invocation, composition and interoperation
  - DAML-S Editor (transitioned to SRI for OWL editor)
  - Golog tool for DAML WS composition; 1st such tool; 80+ citations*
  - Petri-Net tool for WS analysis, simulation and verification; 70+ citations*
  - ** SDS semantic discovery, translation, and explanation tool: shows value-add of OWL-S for BPEL4WS, run-time dynamic binding and integration of WS, paves the way for mixed initiative WS composition.

*http://scholar.google.com*
** Inference Web aims to support the proof layer of the semantic web layer cake and enable explanations with provenance and trust information.

Approach: represent multiple types of information manipulations as inferences using OWL and the *Proof Markup Language (PML)*.

It provides infrastructure for encoding knowledge provenance

*PML* has been developed to represent answer justifications as inferences supporting:
- Multiple justifications
- Justification interoperability
- Multi-language assertions in answer justifications
- Provenance information tracking
- Trust value computation

*IWBase* enables users and applications to create and maintain and reuse justification-related meta-information

*IWTrustNet* will provide services for computing local (user-specific) and global trust values for answers

In addition to PML, IWBase and IWTrustNet, Inference Web provides tools for browsing, abstracting and explaining proofs in PML in multiple presentation formats.

Inference Web and PML are in use at SRI and USC in DARPA’s PAL, at IBM and Batelle in ARDA’s NIMD, IBM in UIMA, at Trento in JSAT and IWTrust, accepted for Sapient, proposed for ASSIST,…
• Hybrid reasoning architecture
  – General purpose first-order logic theorem prover
  – Suite of special-purpose reasoners

• Special-purpose reasoner for OWL query-answering
  – Infers property values of classes and individuals as KB is loaded
    • In effect, caches answers to potential queries
    • Uses linked lists of property values to reduce redundancy

• Application: query-answering service for OWL Web sites
  – Would be a knowledge server for a site’s OWL markup

• Success Measure - Usage: used in Ultralog, Aquaint, PAL, NIMD, accepted for Sapient, …
OWL Query Language (DQL / OWL-QL): Problem and Progress

• Query language for deductive query-answering
  – From knowledge represented in OWL on the Semantic Web
  – Supports a client-server query-answering dialogue
  – The server may derive answers to queries
    • Answers may take an unpredictable amount of time to compute
    • There may be an unpredictable number of answers
  – The knowledge may be in multiple knowledge bases
  – The knowledge bases need not be specified by the client
    • As in a “Query-Answering Google”
  – Answers may only be known to exist
    • Bindings can be to “blank nodes”
  – Provides foundation for Query Manager architecture for PAL, NIMD
Stanford Languages, Services, & Tools: Milestones and Accomplishments

• 2000
  – DAML-ONT initial release (KSL one of 3 editors)
  – DAML-ONT axiomatic semantics (all authors from KSL)
  – OIL and initial integration with DAML-ONT
  – Semantic Web Service vision, rudimentary first-order logic WS ontology

• 2001
  – DAML+OIL design and submission to W3C (Stanford author/editor of 4 documents)
  – DAML+OIL deductive query-answering system; Upgraded to OWL in FY03
  – JTP hybrid architecture, component library, and reasoning system (extended with special purpose reasoners for OWL, Time, Contexts, and upgraded to better support programs like Ultralog)
  – DAML-S 0.5,0.6 ontology, upgraded to 0.7 in ‘02, upgraded to 0.9 in ‘03
  – Golog WS Composition Tool
  – initial DAML-S editor (upgraded in ‘02, transferred to SRI in ‘03)

• 2002
  – First Proof Markup Language specification and API; (upgraded and proof generation services added in ‘03), in use in hybrid reasoner - JTP
  – IWBrowser, augmented with full provenance support, IWBase first centralized, later distributed with database support and services in ‘03.
  – PDDL-to-DAML-S compiler
• 2003
  – OWL-S 1.0 (upgraded and submitted to W3C in 2004)
  – SDS semantic WS discovery and translation tool, enhanced with explanation through inference web in 2004
  – Inference ML specification, PML Abstractor introduced, in use by SNARK,
  – DQL Web-browser client to enable humans to query a DQL server, upgraded to OWL and in use by SRI, USC, …, input to DAWG
  – Gene Ontology distributed and in broad usage
• 2004
  – OWL becomes W3C Recommendation
  – IWExplainer, IWHandler, PML Checker services, are introduced
  – PML fully integrated with text analytics (IBM’s UIMA)
  – TrustNet Design and prototype implementation
  – Protégé SWRL Rule Editor
### Stanford KSL Remaining Issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solidifying OWL Legacy</td>
<td>Semantic Web Best Practices Notes, Outreach, publication, packaging demos, documentation, …</td>
</tr>
<tr>
<td>OWL not expressive enough for WS description</td>
<td>Address limitations imposed by OWL on OWL-S through SWSL ontology work, while maintaining OWL-S legacy.</td>
</tr>
<tr>
<td>Promotion &amp; maintenance of OWL-S legacy</td>
<td>Ongoing OWL-S Coalition work. SWSL activities.</td>
</tr>
<tr>
<td>Extending Proof/Trust/Explanations</td>
<td>PML API, explainer, and browser user manual, document applications, extend user base, build more services, complete IWTrustNet implementation and populate example trust network,</td>
</tr>
<tr>
<td>Extending OWL editors with explanations</td>
<td>Explanation facilities with SWOOP, Pellet, Protégé, Racer, …</td>
</tr>
<tr>
<td>OWL-S &amp; WSMO</td>
<td>Work on integration</td>
</tr>
</tbody>
</table>
• Specs and Committee Work:
  – OWL is a W3C Recommendation!!!!!!!
  – DAML+OIL integrates OIL (DLs) and DAML-ONT
  – OWL-S and SWRL are W3C submitted notes
  – SWSL will be submitted to W3C as a note
  – W3C WebOnt and Semantic Web Best Practices formed
  – OMG Call for OWL MOF model response for SCL and OWL

• Software, Ontologies, Operational Specification Demonstrations:
  – Inference Web in use by Stanford, SRI, USC, IBM, … for explaining semantic web applications
  – OWL-QL implementation in use by ARDA NIMD, DARPA PAL, …
  – OWL-S Editor transferred to SRI
  – Semantic Discovery Services with Explanation in evaluation by France Telecon
  – Pedagogical prototypes: KSL Wine Agent (used in Naval Institute, …), OWL Guide wine/food ontologies, GO,
  – Protégé Rule Editor and integration with OWL Plugin

• Papers: dozens of papers in AIJ, AAAI, DL, IEEE, IJCAI, ISWC, Standards Submissions, Handbook of Description Logic Book, Semantic Web (SWWS 2001), …
• Government transition and high level outreach
  – Semantic Web for the Military User, SWANS, NGA, Counter-terrorism responses, HORUS, NSF cyber-infrastructure, NSF Virtual Solar Terrestrial Observatory, Raytheon DARPA XG and other policy work, …

• Commercial Influence, Interest
  – Acquired Companies: Applied Semantics, Guru,
  – Funded Companies: Network Inference
  – Formed Companies: Radar Networks, Tumri, Buildfolio,
  – Transitioned to existing public companies: Cisco, VerticalNet, General Motors, ArvinMeritor, France Telecon,
  – KeyNotes/Invited Talks: Bell Labs, Cisco, France Telecom, Genentech, IGM, Sony, Sun, VerticalNet,

• Changing the World
  – OWL Recommendation
  – OWL-S and SWRL submissions
  – Public Company significant usage: Cisco, VerticalNet, Sony, France Telecom, IBM Research, Batelle, General Motors, ArvinMeritor
  – Started ISWC series (first co-chaired by us at Stanford, last co-chaired by us)
  – Organized 10+ workshops related to semantic web,
  – Seminal papers: Semantic Web Services Vision, Ontologies Ontologies 101, Web Service Composition, PML,…
  – Invited Talk activity at events like Genetech’s Future information technology, Top Quadrant series, SWANs,
Stanford KSL Summary

- OWL Language provides core representational power with precise semantics enabling reasoning and interoperability
- OWL-S submitted, in use, starting point for SWRL, has multiple tools supporting it, it provides a solution to automation of web service discovery, invocation, composition, and interoperation
- Proofs and Provenance enable explanations and trust for answers from semantic web applications. Proof Markup Language and Inference Web suite enables the proof and trust layers
- Deductive query-answering is a core Semantic Web capability. Hybrid reasoning technology enables Semantic Web deductive query-answering
- KSL Tool Suite provides tools for hybrid reasoning, explanation, trust, service composition, discovery.
Stanford KSL Transition/Handoff

• Where are the results of your work available?
  – Did you influence specs (where?)
    • Fikes & McGuinness; *An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001)*; World Wide Web Committee (W3C) Note 18; December 2001.
    • DQL Specification, lead author
    • OWL-QL Specification, lead author
  – Did you build software (where is it, what is its status)
    • JTP (http://ksl.stanford.edu/software/JTP)
    • DQL Web browser client (http://ksl.stanford.edu/projects/dql/)
  – Did you write papers (where, how many)
  – Did you contribute to a commercial company?
  – Did you contribute to a DoD pilot or product?
  – How did you change the world? Who is using what you developed?
    • University of West Florida in the UltraLog program
    • SRI in the CALO program
    • SAIC in Phase I of AQUAINT
• What is the take-away message from your program?
  – Summarize your program’s problem, solution, approach, and outputs
• What technical problems were there and when/how did you overcome them?
• Are there any metrics that are relevant to your program? How did you measure technical progress and success? What were your intermediate goals?
• Did you meet your original or revised programmatic goals?
• What technical problems were there and when/how did you overcome them?
  – Integrating OWL, FOL, and domain-specific reasoners
    • Developed the JTP hybrid reasoning architecture

• Are there any metrics that are relevant to your program? How did you measure technical progress and success? What were your intermediate goals?
  – Use of JTP and OWL-QL by other contractors in other programs

• Did you meet your original or revised programmatic goals?
  – JTP used in UltraLog, CALO, AQUAINT, and NIMD
Over the course of the time you have been funded by the DAML program, what have you accomplished year-by-year?

- What were the concrete products of your work?
  - Homework assignments, ontologies, specifications, software, reports, papers, etc.
- What was shown in your various demos?
- What can you point to of your work in specifications?
KSL DAML Language, Services, and Tools: Overall Program Summary

• Language:
  – Problem: Existing Markup Languages have limited representational power, lack precise semantics, and lack predictable inference tools
  – Solution: Design (DAML-ONT, DAML+OIL, OWL), (OWL-S, SWSL), (DQL, OWL-QL,) SWRL, Form Joint Committee, WebOnt, SWSA, Semantic Web Best Practices,
  – Research Approach: Leverage Web languages and Description Logics to generate DAML-ONT, DAML+OIL, OWL; leverage DAML/OWL for services ontologies and provide tools to use them; leverage query languages, description logics, and state of the art web languages to provide query and rule languages compatible with state of the art emerging standards.

• Services:
  – Problem:
  – Solution:
  – Research Approach:

• Topic/Tools:
  – Explanation/Inference Web: Applications determine answers from web input but may provide users (humans and agents) with little information defending their answers thus not giving users information that would justify trust. Our work provides a design and infrastructure for knowledge provenance, proofs, explanations, and trust.
  – Hybrid Reasoning/JTP: Many applications do not provide effective deductive question answering from semantic web information services. Our work on JTP – Java Theorem Prover - provides a hybrid reasoning infrastructure for deductive question answering using powerful reasoners, web resources, and rich ontological information.
  – Question Answering/OWL-QL: Semantic Web applications may require powerful and tunable query languages to enable customizable and accurate question answering. Our work designing DQL and later OWL-QL provides such a question answering language and our client implementation provides an implementation that is in use in our work and in other DARPA and ARDA programs and is a candidate standard query language for the semantic web.
(Program Name)
Overall Program Summary

BASIC PROBLEM
• To make network-accessible programs and devices seamlessly interoperable. More specifically, to enable automation of Web service (WS) discovery, invocation, composition, and interoperation.

TECHNICAL SOLUTION
• Interleaved development of:
  – an ontology for describing Web services; (in collaboration w/ Coalition)
  – automated reasoning techniques that manipulate the ontology to perform automated WS discovery, invocation, composition and interoperation;
• informed by and influencing existing and evolving industry WS standards.

ELEMENTS OF PROGRAM AND APPROACH
• KSL (McIlraith) has been a leader in developing the Semantic Web Services vision, both within and outside the DAML program.
• Seminal paper “Semantic Web Services” has received 200+ citations to date*
• OWL-S: Member submission to W3C (11/2004); 250+ OWL-S citations to date*
• Automated reasoning tools for Web Services:
  – Golog tool for DAML WS composition; 1st such tool; 80+ citations*
  – Petri-Net tool for WS analysis, simulation and verification; 70+ citations*
  – SDS semantic discovery & translation tool: shows value-add of OWL-S for BPEL4WS.

* http://scholar.google.com
TECHNICAL PROBLEM AND EVOLVING APPROACH

Ontology for Web Services:

Problem: Develop a means of describing Web service properties and functionality in sufficient detail to enable automated WS discovery, invocation, composition, verification and interoperation, and in a language with a well-defined semantics, that is ideally decidable and tractable.

Evolving Approach:

• Development of the OWL-S (formerly DAML-S) ontology was iterative, influenced by several factors including: the evolution of the OWL (formerly DAML+OIL) language, evolving WS industry standards (including WSDL and BPEL4WS), program experience working with early versions of our ontology.
• Founding member of DAML-S Coalition (02/01).
• DAML-S 0.5, 0.6, 0.7, 0.9 releases (04/01, 12/01, 10/02, 05/03).
• OWL-S 1.0, 1.1 Beta, 1.1 releases (11/03, 07/04, 11/04).
• Founding member of SWSL (09/02) – joint EU/US WS ontology initiative.
• W3C member submission of OWL-S (11/2004).
• Tools to support ontology use: DAML-S to PDDL compiler, DAML-S editor.
• Tools that demonstrate the value of OWL-S. (See next slide)
TECHNICAL PROBLEM AND EVOLVING APPROACH

Automating Web Services Tasks:

Problem: Develop reasoning tools to support automated WS discovery, invocation, composition, interoperation and verification.

Evolving Approach:
Development of tools to support WS automation, adapted to evolving DAML & industry languages and standards.

- Golog tool for DAML WS composition; first DAML Web service composition tool; 200+ + 80+ citations*.
- DAML-to-PDDL compiler for WS composition.
- DAML-S Editor. Transitioned to SRI for profession development of OWL-S Editor.
- Petri-Net tool for WS analysis, simulation and verification; 70+ citations*.
- SDS semantic discovery and translation tool, adding OWL-S to BPEL4WS. Spurred by desire to demonstrate value-added of OWL-S to recent industry standards this tool performs run-time dynamic binding and integration of WS.
- Augmentation of SDS with Inference Web explanations: paves the way for mixed initiative WS composition.
TECHNICAL PROBLEMS & HOW THEY WERE OVERCOME

Problem: DAML+OIL/OWL not sufficiently expressive for WS ontology development, particularly process model development.
Solution: Added supporting syntax to OWL-S. Described semantics by appealing to richer language. SWSL is addressing this more deeply by preserving the OWL-S ontology but addressing short-comings in a richer language.

Problem: Evolution of DAML language (DAML-ONT, DAML+OIL, OWL) and lack of tools and reasoners until well into the program made development of DAML-S/OWL-S difficult.
Solution: We invested the necessary time to continually adapt and build our own tools.

Solution: We invested the necessary time to continually adapt. Some software now obsolete.

Problem: Semantic Web Services do not yet exist, making it difficult to test our software.
Solution: We constructed internal test beds.

METRICS FOR TECHNICAL PROGRESS & SUCCESS.
Large number of paper citations; software requests; invited key-notes; interest from industry (Sony, France Telecom, Cisco, IBM Research); industry speaking requests; program, conference, journal and committee invitations, are all indicators of the impact and success of our work.

MEET ORIGINAL AND REVISED PROGRAMMATIC GOALS?
• Yes. See comments on outstanding items.
YEAR-BY-YEAR ACCOMPLISHMENTS

Concrete Products (all shown in PI meeting demos)*:

2000: Semantic Web Service vision, rudimentary first-order logic WS ontology
2001: DAML-S 0.5, 0.6 ontology, Golog WS Composition Tool, initial DAML-S editor
2002: DAML-S 0.7 ontology, Golog WS Composition Tool (V2), PDDL-to-DAML-S compiler, DAML-S editor (V2)
2003: DAML-S 0.9, OWL-S 1.0, SDS semantic WS discovery and translation tool
2004: OWL-S 1.1, OWL-S Member submission to W3C, SDS *explainable* semantic discovery and translation for Web services tool

* Details of these tools are listed previously.

KSL Work in WS Specifications: All versions of DAML-S, OWL-S and SWRL.
(Program Name)
Transition/Handoff

AVAILABILITY OF RESULTS
Spec Influence: OWL-S, SWSL and as a consequence indirectly WSMO, WSDL
Software: Golog tool for DAML WS composition; 200+ + 80+ citations* (AOR).
Petri-Net tool for WS analysis, simulation and verification; 70+ citations* (AOR).
SDS semantic discovery & translation tool, OWL-S + BPEL4WS (semwebcentral).
PDDL-DAML-S compiler (AOR), DAML-S Editor (transitioned to SRI for industry-level reimplementaiton as
OWL-S Editor). (AOR = available on request)
Papers: dozen papers in WWW, KR, IEEE Intelligent Systems, ISWC, as well a variety of reputable
workshops, symposia. Several are highly cited (http://scholar.google.com)
Commercial companies: Cisco, Sony, France Telecom and IBM Research have all used DAML-S/OWL-S
because of our work.

How did you change the world? Who is using what you developed?
- KSL (McIlraith) has been a leader in developing the Semantic Web Services vision, both within and
outside the DAML program.
- Seminal paper “Semantic Web Services” has received 200+ citations to date*.
- OWL-S: Member submission to W3C (11/2004); 250+ OWL-S citations to date*.
- Golog tool for DAML WS composition was the 1st WS composition tool. Described in “Semantic Web
Services” (200+ citations) & another technical paper (80+ citations*).
- Several companies (listed above) are using our tools.
- Program co-chair ISWC04, co-organizer of 6+ workshops related to semantic WS, 3 key note lectures at
conferences/workshops on the topic of semantic web services, 2 invited presentations at industry-
centric conferences on Web services
- 6+ lectures to industry including Cisco, Sony, France Telecom, Bell Labs, IBM.
<table>
<thead>
<tr>
<th>Issue</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWL not expressive enough for WS description</td>
<td>Address limitations imposed by OWL on OWL-S through SWSL ontology work, while maintaining OWL-S legacy.</td>
</tr>
<tr>
<td>Promotion &amp; maintenance of OWL-S legacy</td>
<td>Ongoing OWL-S Coalition work. SWSL activities.</td>
</tr>
</tbody>
</table>
BASIC PROBLEM
• To make network-accessible programs and devices seamlessly interoperable. More specifically, to enable automation of Web service (WS) discovery, invocation, composition, and interoperation.

TECHNICAL SOLUTION
• Interleaved development of:
  – an **ontology** for describing Web services; (in collaboration w/ Coalition)
  – **automated reasoning** techniques that manipulate the ontology to perform automated WS discovery, invocation, composition and interoperation;
• informed by and influencing existing and evolving industry WS standards.

ELEMENTS OF PROGRAM AND APPROACH
• KSL has been a leader in the developing the Semantic Web Services vision, both within and outside the DAML program.
• Seminal paper “Semantic Web Services” has received 200+ citations to date*.
• OWL-S: Member submission to W3C (11/2004); 250+ OWL-S citations to date*.
• Automated reasoning tools for Web Services:
  – Golog tool for DAML WS composition; 1st such tool; 80+ citations*.
  – Petri-Net tool for WS analysis, simulation and verification; 70+ citations*.
  – SDS semantic discovery and translation tool, adding OWL-S to BPEL4WS.

* http://scholar.google.com
Overall Program Summary

• What is the basic problem you are trying to solve?
  – Enable effective deductive query-answering from Semantic Web information sources

• What was the technical solution strategy?
  – Use hybrid reasoning for deductive query-answering
  – Develop a query language suitable for reasoners and the Semantic Web

• What were the basic elements of the research and program approach?
  – Develop the JTP hybrid reasoner for OWL
  – Develop the OWL-QL candidate standard query language for Semantic Web client-server query-answering dialogues
• What technical problems were there and when/how did you overcome them?
  – Integrating OWL, FOL, and domain-specific reasoners
    • Developed the JTP hybrid reasoning architecture

• Are there any metrics that are relevant to your program? How did you measure technical progress and success? What were your intermediate goals?
  – Use of JTP and OWL-QL by other contractors in other programs

• Did you meet your original or revised programmatic goals?
  – JTP used in UltraLog, CALO, AQUAINT, and NIMD
• Over the course of the time you have been funded by the DAML program, what have you accomplished year-by-year?
  – What were the concrete products of your work?
    • DAML+OIL deductive query-answering system (FY01)
      – Upgraded to OWL in FY03
    • JTP hybrid architecture, component library, and reasoning system (FY01 – ongoing)
    • DQL Web-browser client to enable humans to query a DQL server (April 2003)
  – What can you point to of your work in specifications?
    • Richard Fikes and Deborah McGuinness; *An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001)*; World Wide Web Committee (W3C) Note 18; December 2001.
    • DQL Specification (March 2003)
    • OWL-QL Specification (July 2004)
Where are the results of your work available?

- Did you influence specs (where?)
  - Richard Fikes and Deborah McGuinness; *An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001)*; World Wide Web Committee (W3C) Note 18; December 2001.
  - DQL Specification, lead author
  - OWL-QL Specification, lead author

- Did you build software (where is it, what is its status)
  - JTP (http://ksl.stanford.edu/software/JTP)
  - DQL Web browser client (http://ksl.stanford.edu/projects/dql/)

- Did you write papers (where, how many)

- Did you contribute to a commercial company?
- Did you contribute to a DoD pilot or product?
- How did you change the world? Who is using what you developed?
  - University of West Florida in the UltraLog program
  - SRI in the CALO program
  - SAIC in Phase I of AQUAINT
(Program Name)
Remaining Issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yellow for challenges still working on and expected to be met in the near term</td>
</tr>
<tr>
<td></td>
<td>Red for challenges that will probably remain unresolved in the near future.</td>
</tr>
</tbody>
</table>

- What do you still have to do that is relevant to your original programmatic vision?
- What big problems are still out there?
• Take-away message:
  – OWL Language provides core representational power with precise semantics enabling reasoning and interoperability
  – OWL-S
  – Proofs and Provenance enable explanations and trust for answers from semantic web applications
  – Proof Markup Language
  – Deductive query-answering is a core Semantic Web capability
  – Hybrid reasoning technology enables Semantic Web deductive query-answering
Inference Web: Understanding and Trusting Web Answers

KSL Stanford University

Deborah L. McGuinness, Paulo Pinheiro da Silva and Richard Fikes

November 2004
Inference Web
Overall Program Summary

- Basic problem: To enable web applications to explain their answers including provenance and trust information
- Technical solution strategy: To represent multiple ways of information manipulation as inference. To use OWL to encode inference and inference-related information
- Basic elements: the Inference Web infrastructure including the Proof Markup Language (PML) Specification
PML has been developed to represent answer justifications as inferences
- To allow answers to have multiple justifications
- To enable justification interoperability
- To allow assertions in answer justifications to be represented in any language
- To provide support for tracking provenance information
- To enable the computation of trust values for answers

IWBase enables users and applications to create and maintain and reuse justification-related meta-information

IWTrustNet will provide services for computing local (user-specific) and global trust values for answers

In addition to PML, IWBase and IWTrustNet, Inference Web provides tools for browsing, abstracting and explaining proofs in PML.
Technical problems report
- PML spec has been through three major revisions
- IWBase has moved from a centralized to a distributed architecture and from a manual to a manual/automatic registration of information sources
- Explanation presentation moved from a passive browsing of justifications to an interactive dialog
- Tools have been created to create explanation tactics

Metrics
- Understandability is a major metrics for Inference Web explanations
- IWBrowser usability was an issue for measuring IW explanation understandability (before having the IW Explainer)
- IW Explainer interactive interface enables an understandability study.

Programmatic goals: Infrastructure is ready for use and its foundation is mostly implemented, tested and already in use
Inference Web
Milestones and Accomplishments

• 2002
  - First PML specification and API
  - IWBrowser, centralized IWBase are introduced
  - PML in use by JTP
• 2003
  - First PML specification revision and API
  - IWBase becomes distributed and includes database support; proof generation services and internal services for IWBase are introduced
  - Inference ML specification is introduced
  - PML Abstractor is introduced
  - PML in use by SNARK, ISI Mediator
  - 6 publications and technical reports
• 2004
  - Second PML specification revision and API
  - IWExplainer, IWHandler, PML Checker services are introduced
  - First PML Abstractor revision
  - Explanation tactic editor becomes functional
  - PML in use by SDS, IBM's UIMA
  - 7 publications and technical reports
• IW includes the Prool Markup Language specification (in OWL) and InferenceML specs
• iw.stanford.edu includes the following:
  – PML and InferenceMetaLanguage specifications
  – IW web service prototypes
  – Some IW software for downloading (PML API, IWBrowser and IWBase Domain cell)
  – 10 publications, 5 tech. Reports, 8 posters, 12 presentations
  – 1 public mailing list in SemWebCentral
## Remaining Issues

<table>
<thead>
<tr>
<th>Issue</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yellow for challenges still working on and expected to be met in the near term</td>
</tr>
<tr>
<td></td>
<td>Testing IWExplainer usability</td>
</tr>
</tbody>
</table>

- What do you still have to do that is relevant to your original programmatic vision?
- What big problems are still out there?
• Users may prefer to use agents that can explain how they produce answers including the sources they used
• A user may prefer answers derived from sources either trusted by the user or trusted by someone trusted by the user
• Inference Web enables answer understanding and trust and it is ready for use
• What technical problems were there and when/how did you overcome them?
• Are there any metrics that are relevant to your program? How did you measure technical progress and success? What were your intermediate goals?
• Did you meet your original or revised programmatic goals?
• Hybrid reasoning architecture
  – General purpose first-order logic theorem prover
  – Suite of special-purpose reasoners

• Special-purpose reasoner for OWL query-answering
  – Infers property values of classes and individuals as KB is loaded
    • In effect, caches answers to potential queries
    • Uses linked lists of property values to reduce redundancy

• Application: query-answering service for OWL Web sites
  – Would be a knowledge server for a site’s OWL markup

• Measures: Usage: used in Ultralog, Aquaint, PAL, NIMD, …
• Query language for deductive query-answering
  – From knowledge represented in OWL on the Semantic Web
  – Supports a client-server query-answering dialogue
  – The server may derive answers to queries
    • Answers may take an unpredictable amount of time to compute
    • There may be an unpredictable number of answers
  – The knowledge may be in multiple knowledge bases
  – The knowledge bases need not be specified by the client
    • As in a “Query-Answering Google”
  – Answers may only be known to exist
    • Bindings can be to “blank nodes”
• What technical problems were there and when/how did you overcome them?
  – Integrating OWL, FOL, and domain-specific reasoners
    • Developed the JTP hybrid reasoning architecture
• Are there any metrics that are relevant to your program? How did you measure technical progress and success? What were your intermediate goals?
  – Use of JTP and OWL-QL by other contractors in other programs
• Did you meet your original or revised programmatic goals?
  – JTP used in UltraLog, CALO, AQUAINT, and NIMD
Over the course of the time you have been funded by the DAML program, what have you accomplished year-by-year?

- What were the concrete products of your work?
  - Homework assignments, ontologies, specifications, software, reports, papers, etc.
- What was shown in your various demos?
- What can you point to of your work in specifications?
Inference Web
Technical Progress

• Technical problems report
  – PML spec has been through three major revisions
  – IWBase has moved from a centralized to a distributed architecture and from a manual to a manual/automatic registration of information sources
  – Explanation presentation moved from a passive browsing of justifications to an interactive dialog
  – Tools have been created to create explanation tactics

• Metrics
  – Understandability is a major metrics for Inference Web explanations
  – IWBrowser usability was an issue for measuring IW explanation understandability (before having the IW Explainer)
  – IW Explainer interactive interface enables an understandability study.

• Programmatic goals: Infrastructure is ready for use and its foundation is mostly implemented, tested and already in use
• Over the course of the time you have been funded by the DAML program, what have you accomplished year-by-year?
  – What were the concrete products of your work?
    • Homework assignments, ontologies, specifications, software, reports, papers, etc.
  – What was shown in your various demos?
  – What can you point to of your work in specifications?
Over the course of the time you have been funded by the DAML program, what have you accomplished year-by-year?

- What were the concrete products of your work?
  - DAML+OIL deductive query-answering system (FY01)
    - Upgraded to OWL in FY03
  - JTP hybrid architecture, component library, and reasoning system (FY01 – ongoing)
  - DQL Web-browser client to enable humans to query a DQL server (April 2003)

- What can you point to of your work in specifications?
  - Richard Fikes and Deborah McGuinness; An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL (March 2001); World Wide Web Committee (W3C) Note 18; December 2001.
  - DQL Specification (March 2003)
  - OWL-QL Specification (July 2004)
TECHNICAL PROBLEM AND EVOLVING APPROACH

Ontology for Web Services:

**Problem:** Develop a means of describing Web service properties and functionality in sufficient detail to enable automated WS discovery, invocation, composition, verification and interoperation, and in a language with a well-defined semantics, that is ideally decidable and tractable.

**Evolving Approach:**

- Development of the OWL-S (formerly DAML-S) ontology was iterative, influenced by several factors including: the evolution of the OWL (formerly DAML+OIL) language, evolving WS industry standards (including WSDL and BPEL4WS), program experience working with early versions of our ontology.
- Founding member of DAML-S Coalition (02/01).
- DAML-S 0.5, 0.6, 0.7, 0.9 releases (04/01, 12/01, 10/02, 05/03).
- OWL-S 1.0, 1.1 Beta, 1.1 releases (11/03, 07/04, 11/04).
- Founding member of SWSL (09/02) – joint EU/US WS ontology initiative.
- Tools to support ontology use: DAML-S to PDDL compiler, DAML-S editor.
- Tools that demonstrate the value of OWL-S. (See next slide)
TECHNICAL PROBLEM AND EVOLVING APPROACH

Automating Web Services Tasks:

**Problem:** Develop reasoning tools to support automated WS discovery, invocation, composition, interoperation and verification.

**Evolving Approach:**
Development of tools to support WS automation, adapted to evolving DAML & industry languages and standards.

- Golog tool for DAML WS composition; first DAML Web service composition tool; 200+ + 80+ citations*.
- DAML-to-PDDL compiler for WS composition.
- DAML-S Editor. Transitioned to SRI for profession development of OWL-S Editor.
- Petri-Net tool for WS analysis, simulation and verification; 70+ citations*.
- SDS semantic discovery and translation tool, adding OWL-S to BPEL4WS. Spurred by desire to demonstrate value-added of OWL-S to recent industry standards this tool performs run-time dynamic binding and integration of WS.
- Augmentation of SDS with Inference Web explanations: paves the way for mixed initiative WS composition.
TECHNICAL PROBLEMS & HOW THEY WERE OVERCOME

**Problem:** DAML+OIL/OWL not sufficiently expressive for WS ontology development, particularly process model development..

**Solution:** Added supporting syntax to OWL-S. Described semantics by appealing to richer language. SWSL is addressing this more deeply by preserving the OWL-S ontology but addressing short-comings in a richer language.

**Problem:** Evolution of DAML language (DAML-ONT, DAML+OIL, OWL) and lack of tools and reasoners until well into the program made development of DAML-S/OWL-S difficult.

**Solution:** We invested the necessary time to continually adapt and build our own tools.

**Problem:** Evolution of DAML-S/OWL-S ontology made development of WS tools difficult.

**Solution:** We invested the necessary time to continually adapt. Some software now obsolete.

**Problem:** Semantic Web Services do not yet exist, making it difficult to test our software

**Solution:** We constructed internal test beds.

METRICS FOR TECHNICAL PROGRESS & SUCCESS.

Large number of paper citations; software requests; invited key-notes; interest from industry (Sony, France Telecom, Cisco, IBM Research); industry speaking requests; program, conference, journal and committee invitations, are all indicators of the impact and success of our work.

MEET ORIGINAL AND REVISED PROGRAMMATIC GOALS?

- Yes. See comments on outstanding items.