

Accessing Information and Services on the DAML-Enabled Web

David Martin

Grit Denker, Jerry R. Hobbs,
Srini Narayanan, Richard Waldinger

SRI International



Outline

- What is DAML?
- Observations
 - Semantic Web Queries, Requests, Requirements
 - The Big Picture (Web architecture and evolution)
- Work-in-progress at SRI (and elsewhere)
 - Theories & inference on the Semantic Web
 - **DAML-S**: DAML for Services
 - **Coming soon!**



What is DAML?

- ❑ A DARPA program
- ❑ An input to the W3C Semantic Web activity
 - Will likely be proposed as the draft of a working group recommendation
- ❑ A markup language

www.daml.org



Characteristics of DAML

- ❑ Based on XML & RDF(S)
 - Beyond RDF: properties of properties, equivalence and disjointness of classes, more constraints, etc.
Feature comparison: <https://www.daml.org/language/features.html>
 - Layered approach
XML => RDF(S) => DAML+OIL => (DAML-L) => DAML-S
- ❑ Semantics for Web resources from Knowledge Representation concepts
 - DAML+OIL: can be regarded as a description logic
 - Ontologies
 - Logical rules & inference
- ❑ DAML-S: Extension to Services



Queries, Requests, Requirements

“Find a reference to the latest paper on SHOE with James Hendler as a co-author.”

- ❑ **All concepts identified unambiguously (by URIs)**
- ❑ **Ontologies potentially (likely) distributed**
- ❑ **Reasoning about time, etc.**
- ❑ **“Latest”, etc.: no closed-world assumption**



Queries, Requests, Requirements

“Find a reference to a paper having 2 authors, both of which have been PIs for a DARPA project.”

- ❑ Meta-data for a single response is (very likely) distributed**



Queries, Requests, Requirements

“If Daniel Dennett has published a book about consciousness, request the Stanford Library to hold it for me.”

- ❑ Combine information queries with service requests**



Queries, Requests, Requirements

**“Get me a flight to Washington DC, and
reserve a room near the airport”**

□ Composition of services



Queries, Requests, Requirements

“Execute the refinancing of my mortgage, using the following parameters and providers: ...”

- ❑ Multiple participants in a single transaction**



Requirements

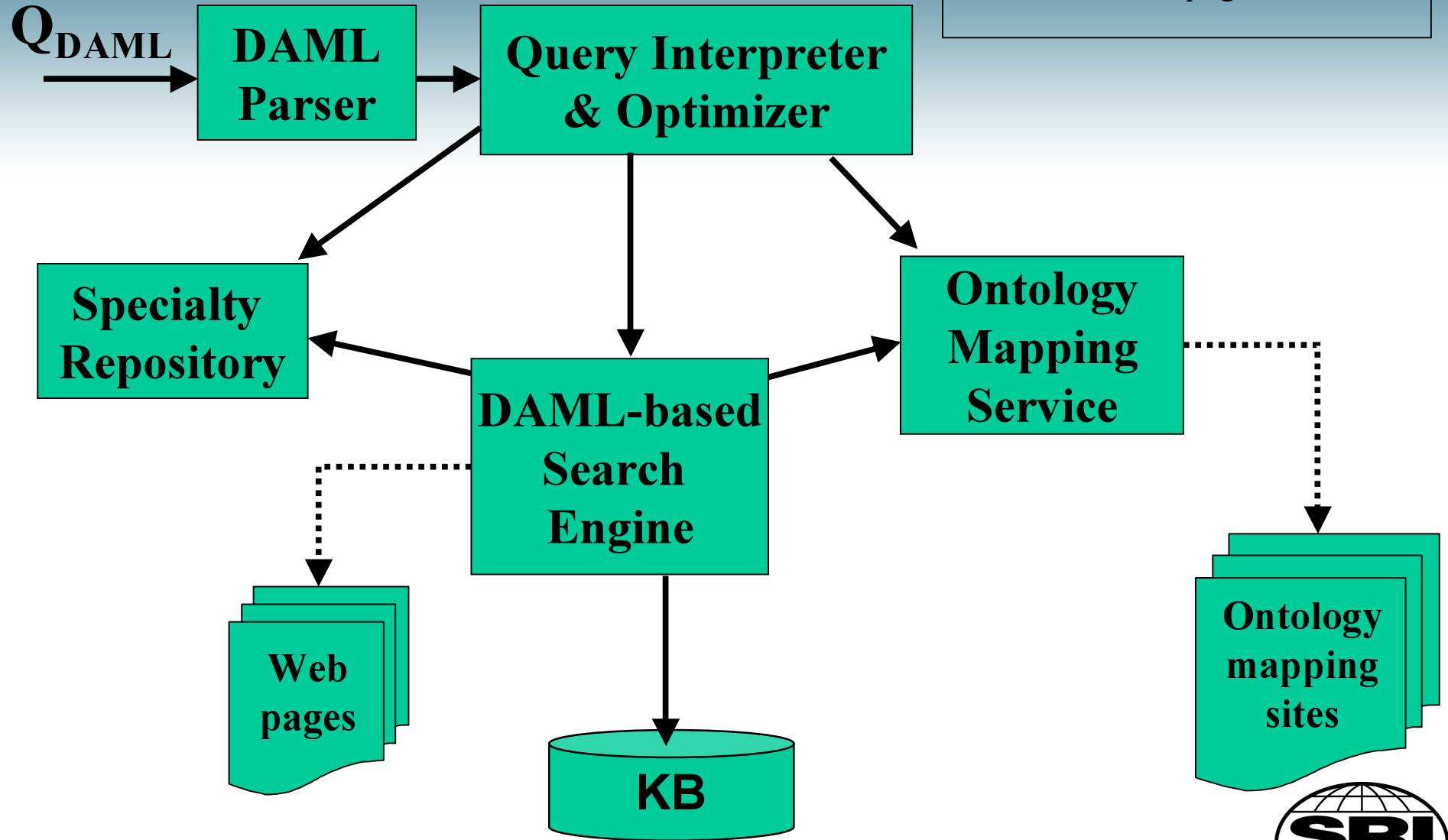
- ❑ All concepts identified unambiguously (by URIs)
- ❑ Ontologies potentially (very likely) distributed
- ❑ Reasoning about time, etc.
- ❑ “Latest”, etc.: no closed world assumption
- ❑ Data for a single response is (very likely) distributed
- ❑ Combines information plus service requests
- ❑ Composition of services
- ❑ Multiple participants in a single transaction

- ❑ **Ontology mappings needed**

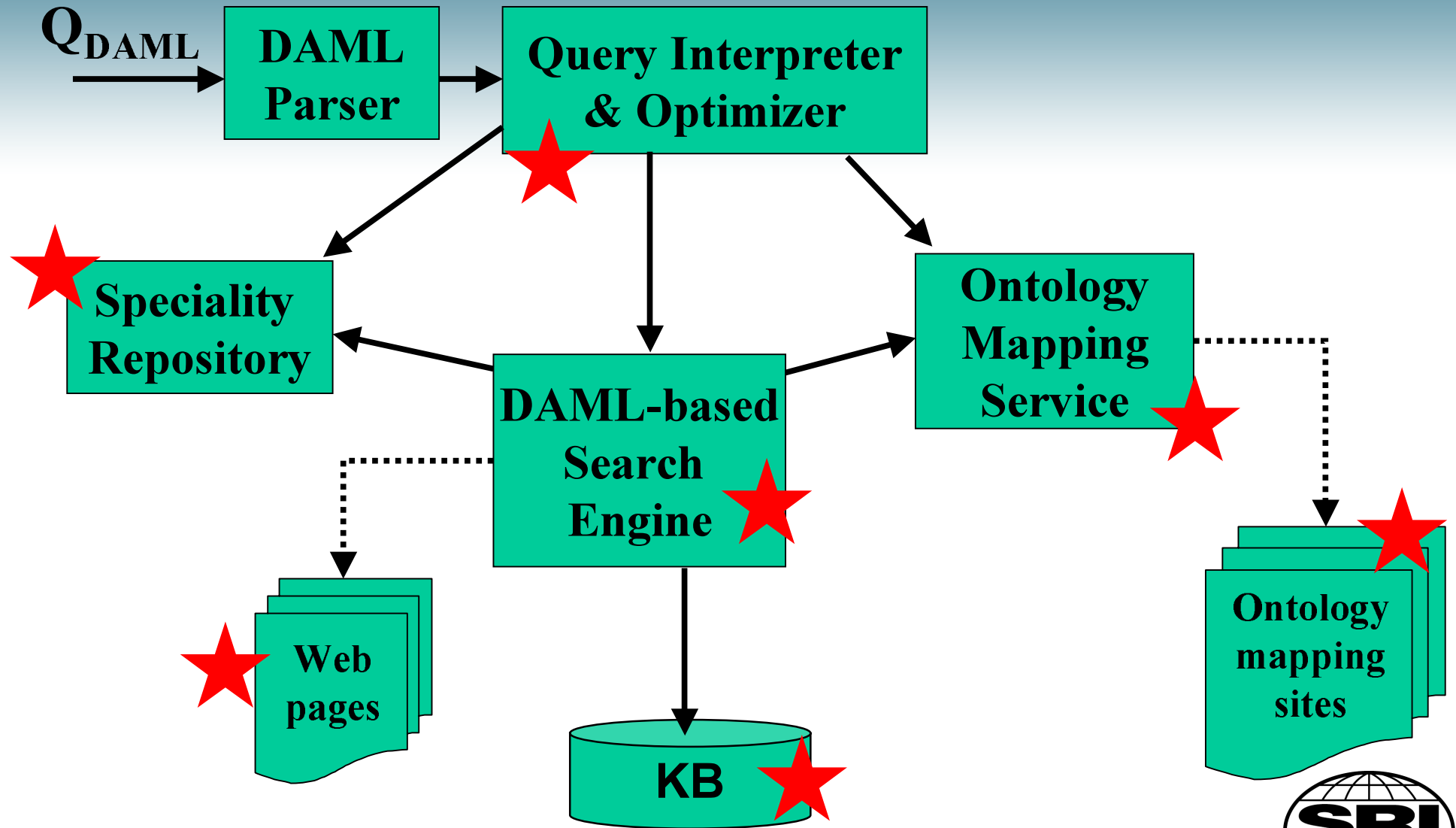


Basic Architecture

—> Flow of control
- - -> Web page access



Where can **logical rules & inference** be used?

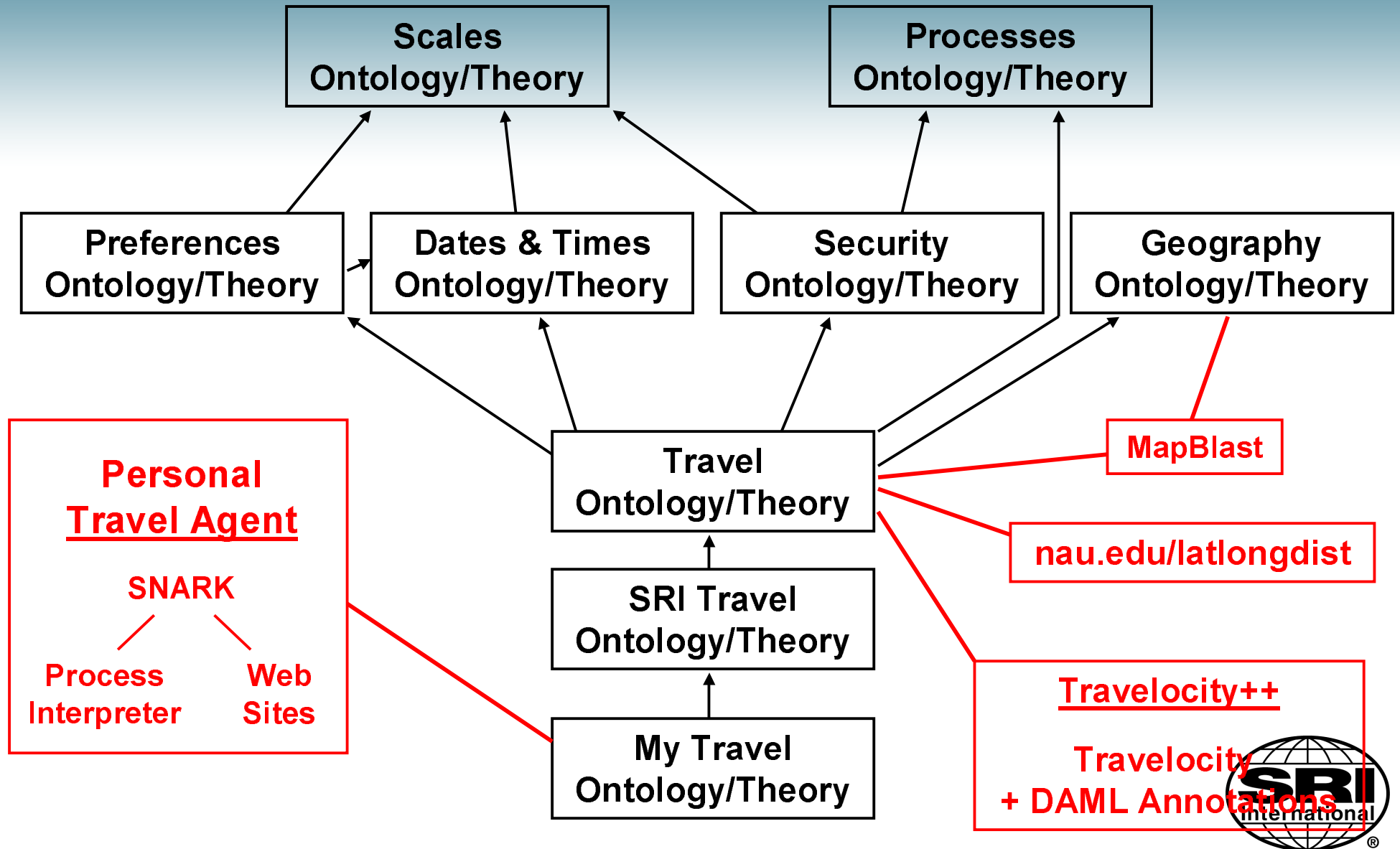


Rules, Theories, Inference

- ❑ Needed everywhere
 - Class declarations
 - Expressing and answering (decomposing) queries
 - Service descriptions, advertisements
 - Service requests, possibly service bindings
 - Composing services
 - Expressing and using ontology mappings
 - Background knowledge
- ❑ Challenge: balance expressive power, high performance, desire for simplicity



Using Distributed Theories



Linking Ontologies

Travelocity++

Linking Axioms:

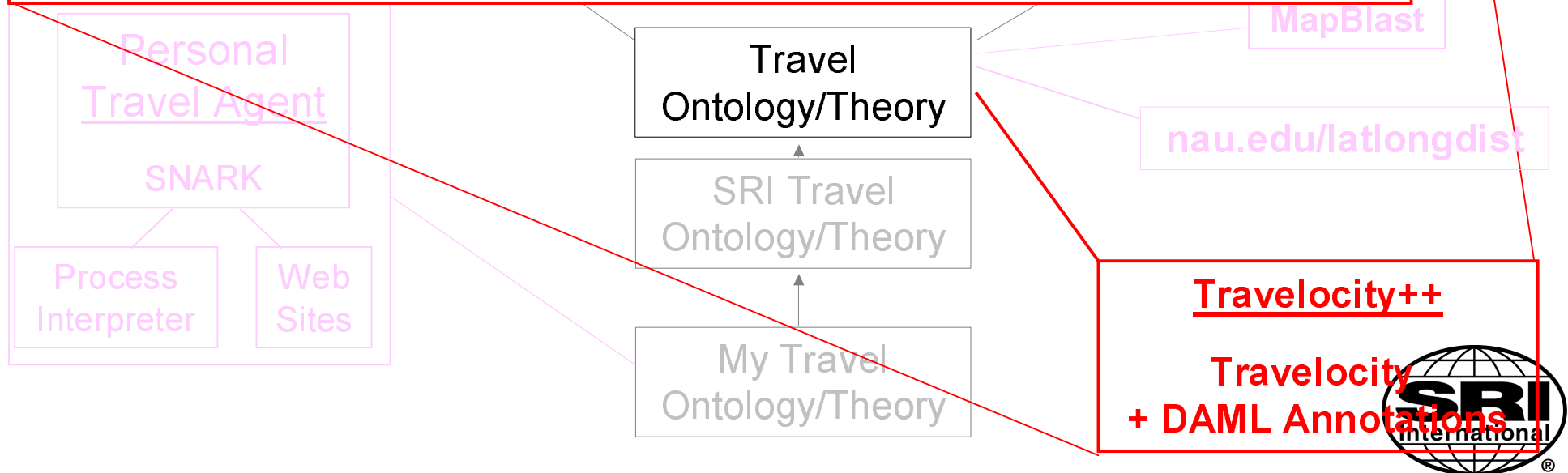
(implies

(Travelocity:Get_Itinerary ?OrigAPC ?DestAPC ?DepTime ?ArrTime
?Flt# ?Fare)

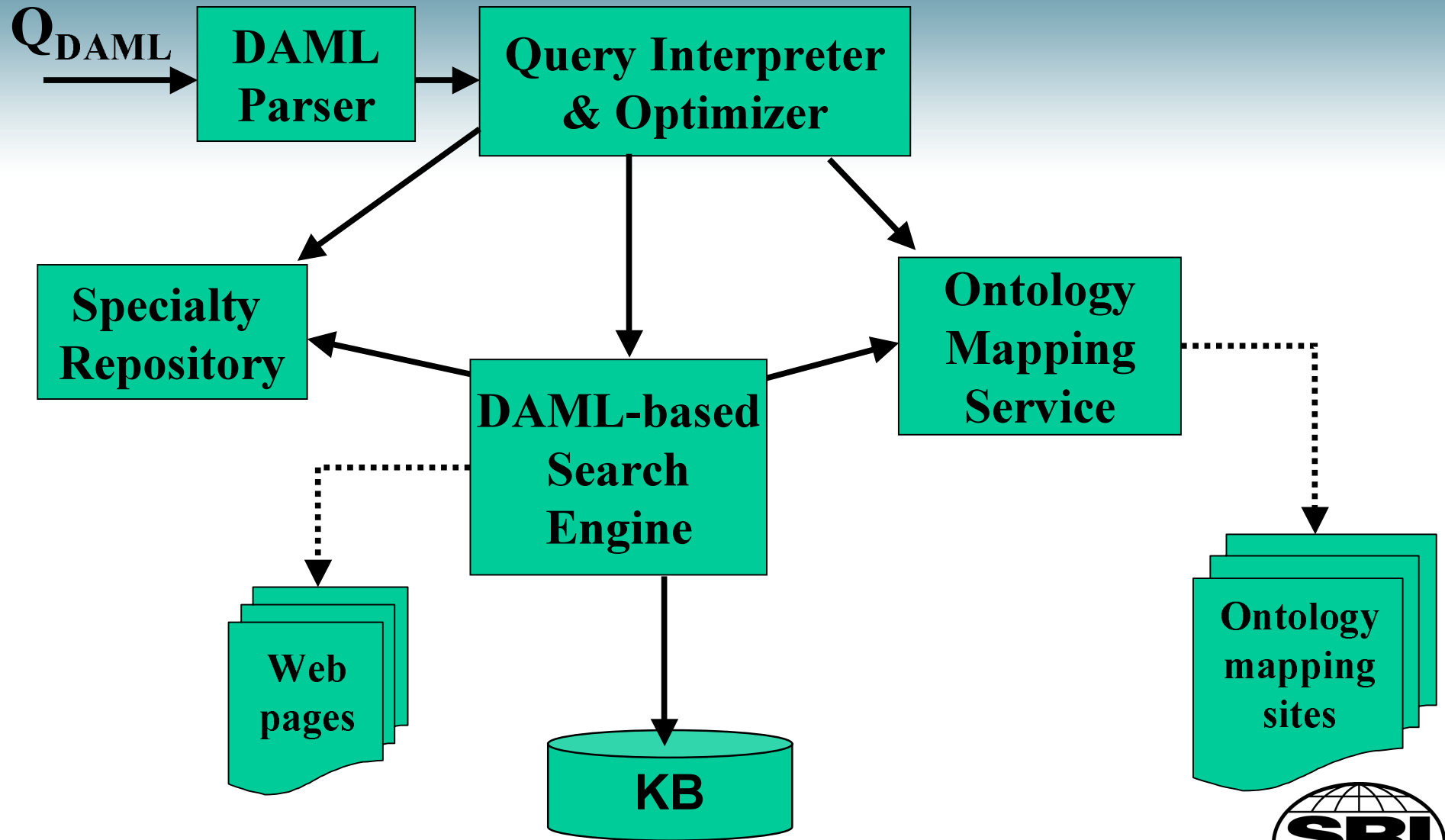
(and (DepCity ?Flt (city-of ?OrigAPC)) (DestCity ?Flt (city-of ?DestAPC))
(DepTime ?Flt ?DepTime) (ArrTime ?Flt ?ArrTime)
(FlightNo ?Flt ?Flt#)(Price ? Flt ?Fare)))

.....

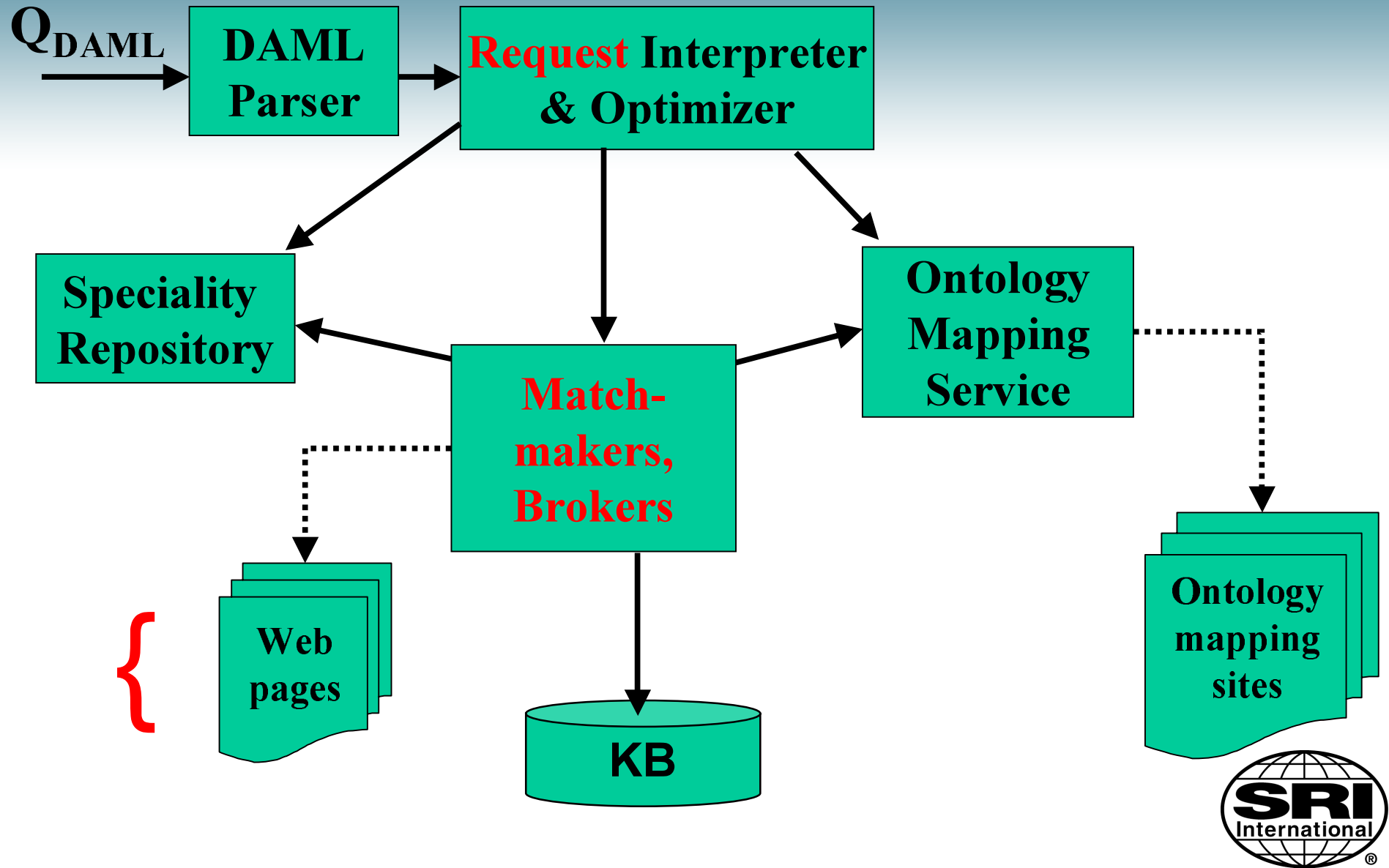
Protocol: OAA, IDL, JINI



How do **services** change the picture?



How do **services** change the picture?



DAML-S: Goals

- ❑ Full automation of service use
 - DAML markups provide enough info for an agent to find, select, and use a service never before encountered
- ❑ Service requests handled seamlessly with information queries
 - Allow for composition of both
 - Many components & tools can work for both
 - Search & selection, ontology translation, ...
- ❑ Support inference in selecting and using services

Joint work with Stanford, CMU, BBN, Nokia

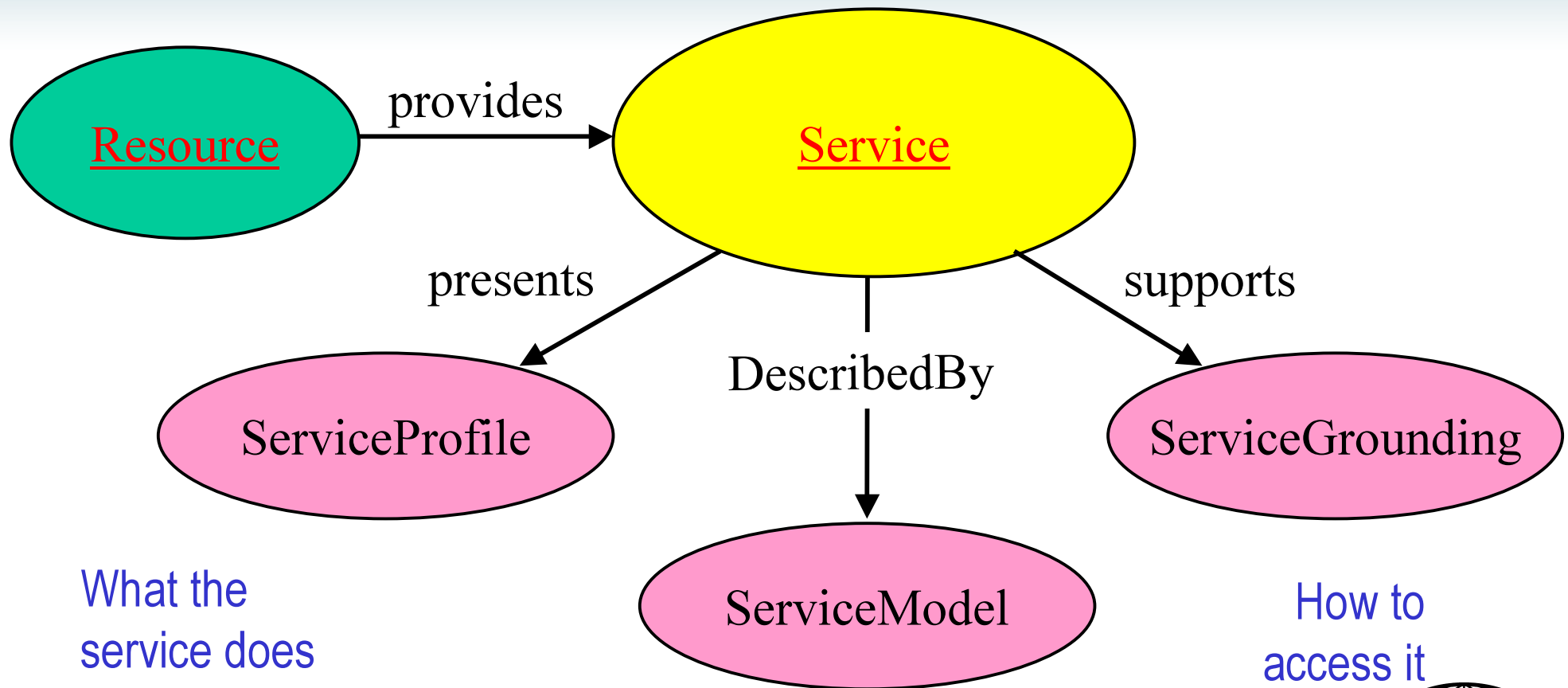


Upper Ontology for Services

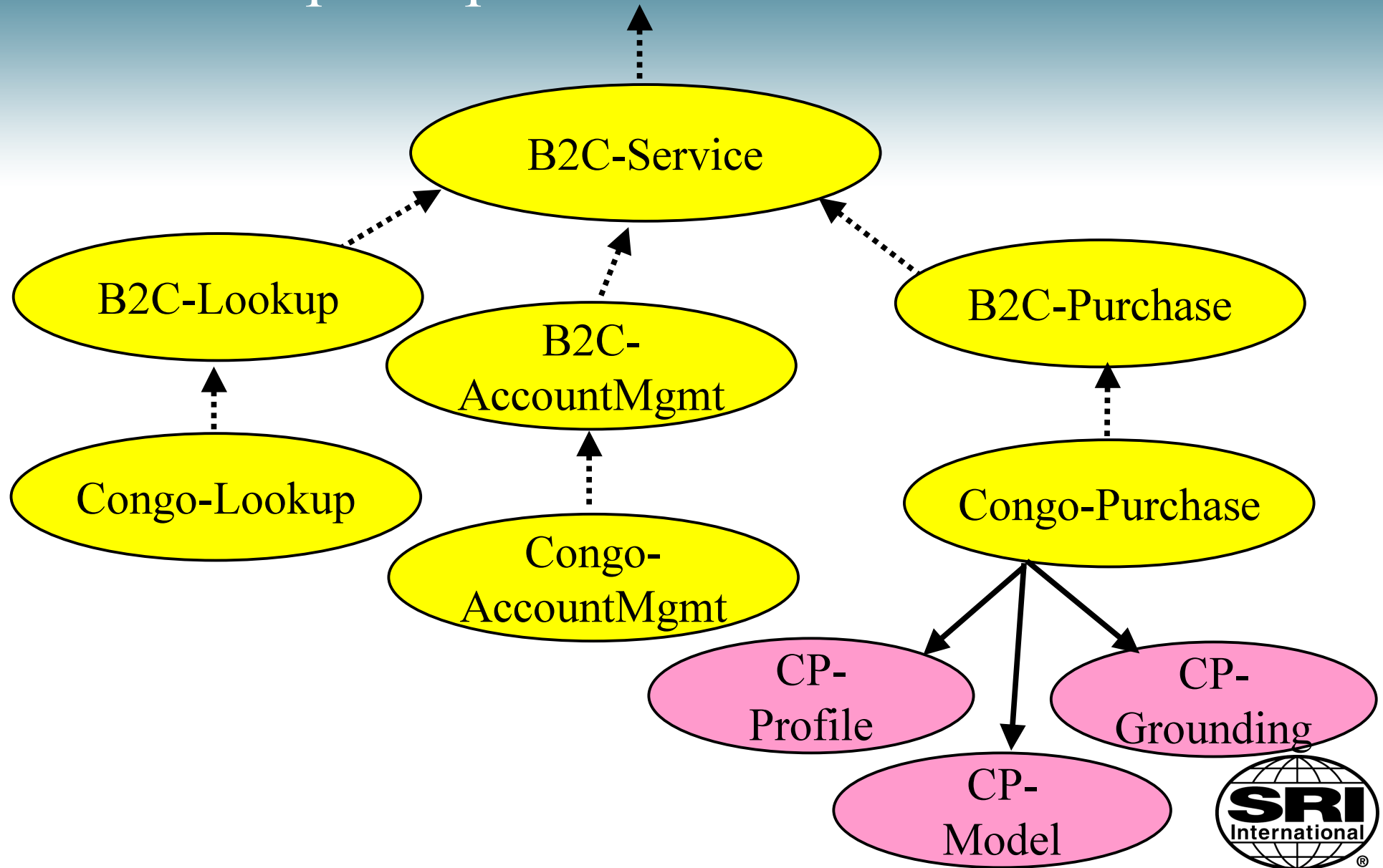
- ❑ A foundation for creating many service ontologies
 - But not a mandate for specific ontologies
 - May provide deeper ontologies for Meta-services
- ❑ Can be specialized in many different ways
- ❑ No one “official hierarchy” of services
 - But agents always know how to get started
 - Top-level specs provide consistency
- ❑ Existing taxonomies can be mapped



Service Ontology: Top-level Classes



Example: Specialization for a B2C Site



Service Profile

“What does it do?”

- ❑ Requirements for use; results of use
 - “Black box” view: Information needed to find and select a service
 - Inputs, outputs, preconditions, effects, ...
 - “Binding rules” for inputs, outputs
 - “Roles” involved
- ❑ May vary for different service classes
- ❑ Can employ logical rules
- ❑ Analogous to procedure header, DB schema



B2C Purchase: Profile

- ❑ Input: ItemDescription (several forms), PriceRange, AcctName, Passwd, CreditCard#, Shipping-address, ...
- ❑ Input usage rule:
 $\text{Exists}(\text{Acct}) \Rightarrow \text{Defined}(\text{CreditCard\#}, \text{Shipping-Address})$
- ❑ Precondition:
 $\text{Exists}(\text{Acct}) \mid \text{CanCreate}(\text{Acct})$
- ❑ Output: 'Succeed' + Receipt | 'Cancel' | 'Fail'
- ❑ Effect: 'Succeed' \rightarrow ShippingOrderPlaced



Service Model

“How does it work?”

- ❑ Semantic description of a service
 - “Glass box” view
 - Detailed characterization of what it does
- ❑ May vary for different service classes
- ❑ Can employ logical rules
- ❑ Analogous to procedure body (but abstract)



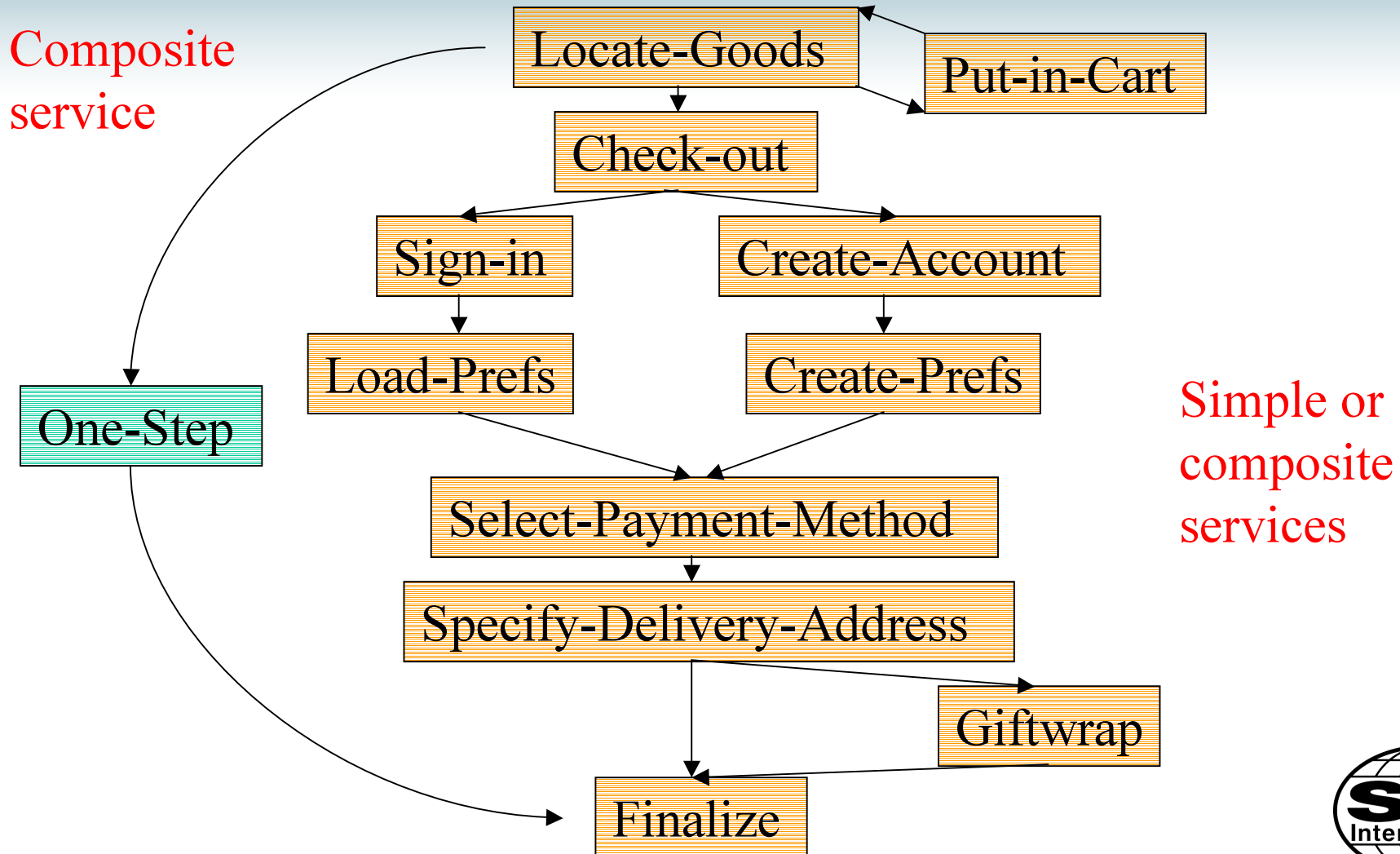
Process Model

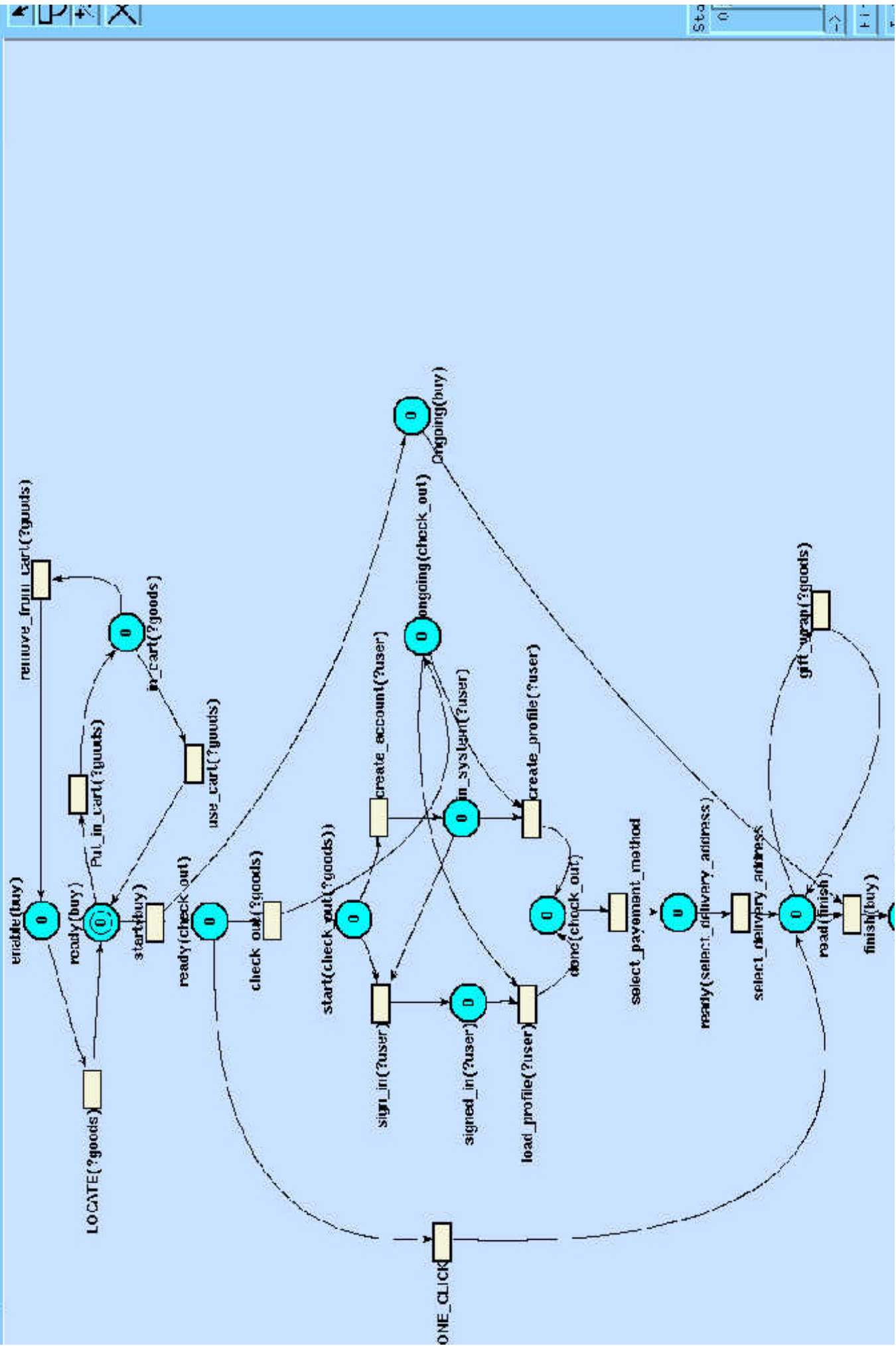
(SubClass of ServiceModel)

- ❑ Shared knowledge to coordinate service's "roles"
- ❑ Expresses state changes
- ❑ Describes sequences of possible interactions for an extended transaction
- ❑ Executable semantics
- ❑ Draws on work in AI planning, workflow, ...
- ❑ Can be used for task planning, scheduling, reachability analysis, etc.



B2C Purchase: ProcessModel





CongoPurchaseProcessModel.daml (sketch)

```
<process>
  <name> purchase </name>
  <sequence>
    <process> locate_goods (?goods)</process>
    <alternative>
      <alt1><task> One_step </task></alt1>
      <alt2> <sequence>
        <task> put_in_cart(?goods)</task>
        <process>
          <name> check_out_info</name>
          <alternative>
            <alt1> <sequence>
              <task> Sign_in (?user)</task>
              <task> Load_profile(?user.profile) </task>
            </sequence></alt1>
            <alt2> <sequence>
              <process> Create_Account (?user)</process>
              <process> Create_profile(?user.profile) </process>
            </sequence></alt2>
          </alternative>
        </process>
        <task> Select_Payment_Method </task>
        <process> Select_Delivery_Address </process>
      </sequence></alt2>
    </alternative>
    <task> Finalize </task>
  </sequence>
</process>
```



Service Grounding

“How is it used?”

- ❑ Implementation-specific details for accessing the service
- ❑ Message formatting, transport mechanisms, protocols, serializations of all types
- ❑ Service Model + Grounding give everything needed for using the service
- ❑ Examples: HTTP forms, SOAP, KQML, CORBA IDL, OAA ICL, Java RMI



B2C-Purchase: Grounding

- ❑ Transport: [Secure HTTP](#)
- ❑ Protocol: [HTTP Forms](#)
- ❑ Address: <https://buybot.congo.com:4040/initsub.htm>
- ❑ Type Serializations
 - [ItemDescription \(keywords\)](#): Set of DAML literals
 - [PriceRange](#): pair of monetary units, ISO 5678
 - [CreditCard](#):
<https://transcredit.com/S1.daml#SecureTransferFormat>



Recap of Upper Ontology for Services

- ❑ Profile supports service selection
- ❑ Model + Grounding support execution, monitoring, composition, ...
- ❑ Profile and Model are abstract; Grounding makes it concrete
- ❑ There can be multiple of each
 - One-to-one correspondence not required



Summary

- ❑ The Semantic Web will be big
- ❑ It will support a wide variety of (mixed) queries and requests, in a semantically-grounded way
- ❑ KB representational techniques, ontologies, axioms, reasoning are likely to be important elements
- ❑ Services can be advertised, found, executed, monitored, and composed using DAML-S
- ❑ Search engines & portals will evolve; ontology translation services will become essential
- ❑ Interesting new challenges for distributed DB/KB technology and Web architecture



Status

- DAML-S coming soon to

<http://www.daml.org/services/daml-s>

- Joint work with Stanford KSL, CMU, BBN,
Nokia

- SRI's DAML work is described at

<http://www.ai.sri.com/daml>

