The Semantic Web: XML, RDF, OWL, and Description Logic

Presented by Joe Kopena
tjkopena@cs.drexel.edu

March 19, 2007

Simple First Order Statement

- **A basic concept to be expressed, taken from engineering design:**

  All motors are engineering artifacts. They take electrical energy as input and produce rotational energy as output.

- **That concept in first order logic, assuming some background ontology:**

$$\forall x \ \textbf{Motor}(x) \supset \textbf{Artifact}(x).$$
$$\forall x \ \textbf{Motor}(x) \supset [\exists y \ \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)].$$
$$\forall x \ \textbf{Motor}(x) \supset [\exists y \ \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].$$

- **Note that this does not preclude other inputs and outputs.**

Inference

- **Given the statements:**

$$\forall x \ \textbf{Motor}(x) \supset \textbf{Artifact}(x).$$
$$\forall x \ \textbf{Motor}(x) \supset [\exists y \ \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)].$$
$$\forall x \ \textbf{Motor}(x) \supset [\exists y \ \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].$$

- **And the fact:**

$$\textbf{Motor}(LEGO43362).$$

- **We can derive the following:**

$$\textbf{Artifact}(LEGO43362).$$
$$\exists y \ \textbf{input}(LEGO43362, y) \wedge \textbf{ElectricalEnergy}(y).$$
$$\exists y \ \textbf{output}(LEGO43362, y) \wedge \textbf{RotationalEnergy}(y).$$

Classification

- **If we know the following, what can we infer?**

$$\textbf{Artifact}(LEGO - NXT).$$
$$\textbf{input}(LEGO - NXT, Input1).$$
$$\textbf{output}(LEGO - NXT, Output1).$$
$$\textbf{ElectricalEnergy}(Input1).$$
$$\textbf{RotationalEnergy}(Output1).$$

- **Nothing; the statements provide necessary but not sufficient criteria for membership in Motor.**

Necessary and Sufficient Conditions

- **Change the form of the original statements a bit:**

$$\forall x \ \textbf{Motor}(x) \supset \textbf{Artifact}(x) \wedge$$
$$[\exists y \ \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)] \wedge$$
$$[\exists y \ \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].$$

- **And alter the semantics, changing ($\supset$ to $\equiv$):**

$$\forall x \ \textbf{Motor}(x) \equiv \textbf{Artifact}(x) \wedge$$
$$[\exists y \ \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)] \wedge$$
$$[\exists y \ \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].$$

Class Membership

- **The altered definition:**

$$\begin{aligned}\forall x \;\; \textbf{Motor}(x) \equiv \textbf{Artifact}(x) \wedge \\ [\exists y \;\; \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)] \wedge \\ [\exists y \;\; \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].\end{aligned}$$

- **And the facts:**

$$\begin{aligned}&\textbf{Artifact}(LEGO - NXT) \wedge \\ &\textbf{input}(LEGO - NXT, Input1) \wedge \textbf{output}(LEGO - NXT, Output1) \wedge \\ &\textbf{ElectricalEnergy}(Input1) \wedge \textbf{RotationalEnergy}(Output1).\end{aligned}$$

- **Then imply:** $\textbf{Motor}(LEGO - NXT)$**.**

Description Logic Concept

- **The original Motor concept in FOL:**

$$\forall x \ \textbf{Motor}(x) \supset \textbf{Artifact}(x) \wedge$$
$$[\exists y \ \textbf{input}(x, y) \wedge \textbf{ElectricalEnergy}(y)] \wedge$$
$$[\exists y \ \textbf{output}(x, y) \wedge \textbf{RotationalEnergy}(y)].$$

- **The Motor concept in description logic syntax with necessary conditions:**

$$\textbf{Motor} \sqsubseteq \textbf{Artifact} \sqcap$$
$$\exists \textbf{input}.\textbf{ElectricalEnergy} \sqcap$$
$$\exists \textbf{output}.\textbf{RotationalEnergy}.$$

Description Logic Concept

- The Motor concept with necessary and sufficient conditions:

$$\begin{aligned}\textbf{Motor} \equiv\ & \textbf{Artifact} \sqcap \\ & \exists \textbf{input}.\textbf{ElectricalEnergy} \sqcap \\ & \exists \textbf{output}.\textbf{RotationalEnergy}.\end{aligned}$$

- Description logic focuses on objects and relations between them
  - Similar feel, & roots in, object oriented modeling and programming
- The most common, closely related, inferences:
  - *Subsumption*: Is a given class a subclass of another given class?
    - TBox reasoning—determining the relationships in a terminology
  - *Membership*: Is a given object a member of a given class?
    - ABox reasoning—classified a collection of individuals

A Typical Description Logic

- **Language constructs and semantics for the DL $\mathcal{ALEN}$**
  - **These are the most commonly used concept constructs**

Name	Notation	Interpretation
Top-Concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom-Concept	$\bot$	$\emptyset$
Primitive Negation	$\neg \mathbf{A}$	$\Delta^{\mathcal{I}} \setminus \mathbf{A}$
Intersection	$\mathbf{C} \sqcap \mathbf{D}$	$\mathbf{C} \cap \mathbf{D}$
Value Restriction	$\forall \mathbf{r}.\mathbf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in \mathbf{R}^{\mathcal{I}} \rightarrow b \in \mathbf{C}^{\mathcal{I}}\}$
Full Existential Restriction	$\exists \mathbf{r}.\mathbf{C}$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in \mathbf{R}^{I} \wedge b \in \mathbf{C}^{\mathcal{I}}\}$
Unqualified At-Most	$\leq n\ \mathbf{R}$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathbf{R}^{\mathcal{I}}\} \mid \leq n\}$
Unqualified At-Least	$\geq n\ \mathbf{R}$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathbf{R}^{\mathcal{I}}\} \mid \geq n\}$
Unqualified Exactly	$= n\ \mathbf{R}$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathbf{R}^{\mathcal{I}}\} \mid = n\}$

The Resource Description Framework (RDF)

- **RDF provides an XML-based language for describing individuals**
- **The Motor-NXT example in RDF:**
  - `eng` **and** `flow` **are the ontology namespaces**

```
<eng:Artifact rdf:about="#LEGO-NXT">
  <eng:input><flow:ElectricalEnergy /></eng:input>
  <eng:output><flow:RotationalEnergy /></eng:output>
</eng:Artifact>
```

- **"Produces" the following tuples:**

	Predicate	Subject	Object	
<	rdf:type	eng:Artifact	#LEGO-NXT	>
<	eng:input	#LEGO-NXT	#anonymous1	>
<	rdf:type	#anonymous1	flow:ElectricalEnergy	>
<	eng:output	#LEGO-NXT	#anonymous2	>
<	rdf:type	#anonymous2	flow:RotationalEnergy	>

The Ontology Web Language (OWL)

- **OWL provides a language for and in RDF for describing DL classes**
- **The necessary and sufficient Motor concept in OWL syntax:**

```
<owl:Class rdf:about="#Motor">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="&eng;#Artifact" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="&eng;#input" />
      <owl:hasValue rdf:resource="&flow;#ElectricalEnergy" />
    </owl:Restriction>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&eng;#output" />
      <owl:hasValue rdf:resource="&flow;#RotationalEnergy" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```