Assembly Representations for Design Repositories and the Semantic Web

Joseph Kopena
joe@plan.mcs.drexel.edu

December 2002
Outline

• Overview
• Representation
• Application
• Semantic Web
• Future Plans
Problem

- Managing and utilizing engineering design repositories
  - For example, the National Design Repository

- Current techniques/practices insufficient
  - Often based on name or product line, manual categorization, etc.
• **Approach**

• **Apply knowledge representation techniques to model and reason on artifacts**

• **Emphasis on modeling/reasoning on function**

![Diagram showing the workflow of the approach]

1. **Input Artifacts**
2. **Extract Semantics**
3. **Build Knowledge Base**
4. **Categorize/Browse**

![Legend and iconography related to the diagram]
Information Model

- Contact graph:

\[ \forall u \in V, v \in V \quad (u, v) \in E \equiv \text{pointset}(u) \cap \text{pointset}(v) \neq \emptyset \]
• Contact graph in triple model/conceptual graph:

\[ \forall u, v \langle \text{contact } u \ v \rangle \equiv \text{pointset}(u) \cap \text{pointset}(v) = \emptyset \]

\[ \langle \langle \text{contact Brick Axle} \rangle \ 
\langle \text{contact Axle Gear16} \rangle \rangle \]
• **Assembly hierarchy:**

![Diagram showing assembly hierarchy]
• Combined assembly hierarchy and contact graph:
**Representation: Features**

- Capture abstract feature information
  - Type, basic components, mapping to geometry

\[
\text{CylindricalFeature} \subseteq \text{AssemblyFeature} \cap \exists \text{shape. Cylinder.}
\]

\[
\text{Shaft} \subseteq \text{Male} \cap \text{CylindricalFeature}.
\]

\[
\text{Cylinder} \subseteq \text{GeometricEntity} \cap \exists \text{centerline. Line} \cap \exists \text{radius. Distance.}
\]

\[
\text{Line} \subseteq \text{GeometricEntity} \cap \exists \text{point. Point} \cap \exists \text{vector. Distance.}
\]

\[
\text{Point} \subseteq \text{GeometricEntity} \cap \exists \text{xcoord. Distance} \cap \exists \text{ycoord. Distance} \cap \exists \text{zcoord. Distance.}
\]
• Some features of the previously shown brick:
Represent relationships between parts:
- Fits, mates, rotations, translations, gears
Representation: Function

- Emphasis on modeling/reasoning on function
Example: Part Classification

- **Small hierarchy for part categorization:**

  \[\text{PiecesWithStuds} \equiv \geq 1 \text{assemblyFeature}.\text{Stud}.\]
  \[\text{PiecesWithAxleHoles} \equiv \geq 1 \text{assemblyFeature}.\text{AxleHole}.\]
  \[\text{PiecesWithTeeth} \equiv \geq 1 \text{assemblyFeature}.\text{GearTeeth}.\]
  \[\text{PiecesWithAxleConns} \equiv \geq 1 \text{assemblyFeature}.\text{AxleConnector}.\]
  \[\text{PiecesWithTechnicHoles} \equiv \geq 1 \text{assemblyFeature}.\text{TechnicHole}.\]

  \[\text{LEGOPiece} \equiv \text{PiecesWithStuds} \sqcup \text{PiecesWithAxleConns} \sqcup \text{PiecesWithTechnicHoles}.\]
Example: Part Classification (cont.)
Example: Part Classification (cont.)

- Hierarchy with parts classified:

```
LEGOPiece
  ┌──────────┐
  │         │
  │         │
  │         │
  └──────────┘
PiecesWithStuds  PiecesWithAxleConn  PiecesWithTeeth

PiecesWithTechnicHoles

PiecesWithAxleHoles
```

Slide 14 of 21
LEGO Robots

- Popular hobbyist, educational robot platforms
Online Collections of Designs, Information

- Provide information, instructions, designs to students

The Buggy

The Buggy is a very simple robot in terms of software and construction. It has an interesting steering mechanism and one interesting behavior. Its software very simply tries to navigate the robot towards bright light sources. When placed in the hallway outside our lab the robot can successfully navigate around the building simply by using this behavior and the fact that obstacles create shadows which the robot avoids. It's fairly impressive given its simplicity. Unfortunately it has some trouble navigating in the smaller environment of our LEGO lab table because it has a large turning radius.

Pics

The two black boxes atop the motors contain light sensors. Note the very simple construction, with few pieces being used in the model. The robot twists in the middle by advent of a large turntable piece, which allows it to steer by powering the wheels at different levels.

SRF04 Connections and Mounting

The following pictures demonstrate one way to connect an SRF04 sensor and a servo motor to an HBR. This is one way to connect the units used in Robot Building Lab. Another construction developed by Nick Morris is presented here.

This shot gives a general overview of the wiring connections and the servo motor assembly.

Pics

This shot illustrates how to hook up the servo motor using the A5 port (not the standard digital port B header). The white line plugs into TO3 in the Handboard Expansion brick. In order to avoid unnecessarily clobbering the LEGO every time the servo has to be initialized or connected, we have provided each group in SRL with a small shunt which plugs into TO3. The shunt is at the left hang, and the white servo line plugged into TO3. The orientation of the plug is not important. The black connector lugs are the servo power headers, just above the power lights. The servo is connected to ground and thus the wires should be on the right side.

I T C S L : S R F 0 4  C o n n e c t io n s  a n d  M o u n t in g h t t p : / / p la n . m c s . d r e x e l . e d u / p r o j e c t s / l e g o r o b o ts / h a r d w a r e / s e n s o r s / s o ...

1 of 4 12/01/2002 01
• Capture important information about artifact using representation
  – Then interpret as a class so comparisons can be made

\[
\text{CDSCell} \equiv \text{Artifact} \sqcap \\
\exists \text{function.} (\text{Measure} \sqcap \\
\exists \text{output. AnalogElectricalSignal}) \\
\sqcap \exists \text{input. Light.}
\]
One of the most common items used on small hobby or educational robots are light sensors. With even the simplest of sensors, they enable the robot to perform tasks such as navigating towards a light, hiding in dark corners, following other robots, etc. More advanced uses permit following lines and detecting obstacles.

The most common of such sensors fall into two categories: photoresistors and phototransistors. One more particular type of the former are made of Cadmium Sulphide cells. A picture is presented below. These are commonly available from Radio Shack or from any electronic components catalog pretty cheaply.

CDS Cells are photoresistive light sensors. When no light is present their impedance is extremely high, a consequently very low when no light is present. In contrast to phototransistors, these cells generally don’t see to have as large a range of values between the two extremes of light and dark. These cells also have a much slower reaction time in response to changes in light as they have a large memory effect.

These sensors are straightforward to wire. They’re bidirectional, so simply connect one leg to your sensor input pin and the other to ground. Follow this link for a discussion on connecting phototransistors to a HandyBoard. These CDS Cells connect in the Ziflex form factor except you don’t have to worry about which leg goes to which pin.

[Note: This page is a small demonstration of marking up pages about robots and their components. You can click on the “DAML Inside” link to see the markup, which corresponds to the function and flow diagram above. The diagram not (truly) necessary as it currently has nothing in it. I know that the markup is in a weird form—a class oriented view—which I think will be edited quickly. Instead, the markup is intended to show the possibilities of this markup and damn near impossible to edit quickly.]

[Note: This page is an example showing the多么 interesting things to add is the additional information present in this text, such as the notes about its response time and notes on connecting it to other devices.]
Semantic Web

- Limited expressiveness
  - Limited translation
  - Limited reasoning

- Desirable computability
  - Theoretically and pragmatically
Future Work

• Develop annotation tool
  – Incorporate representation into MUG (conceptual design tool)

• Apply to RobotLab course at Drexel
  – Populate collection of designs with student solutions
  – Examine interface needs, system requirements

• Richer ontology
  – Explore more expressive Web-enabled logic languages?
  – Enable more reasoning
    * Automated design
Acknowledgements

- This work was supported in part by National Science Foundation (NSF) CAREER Award CISE/IIS-9733545 and Office of Naval Research (ONR) Grant N00014-01-1-0618. Additional support has been provided by Honeywell FM&T, AT&T Labs, Bentley Systems and Lockheed Martin, Naval Electronics and Surveillance Systems.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.