DESIGN REPOSITORIES ON THE SEMANTIC WEB WITH DESCRIPTION-LOGIC ENABLED SERVICES

Joseph B. Kopena[⋆] and William C. Regli[⋆⋆]

Geometric and Intelligent Computing Laboratory
Department of Computer Science, College of Engineering
Drexel University
`http://gicl.cs.drexel.edu/`

Abstract. All engineering firms maintain archives of previously designed artifacts, often in the form of databases of computer aided design (CAD) data. Design repositories are an evolution of such databases to include more heterogenous information and to provide enhanced capabilities through the application of knowledge representation techniques. This paper introduces on-going work on applying description logic and the Semantic Web to constructing such design repositories.

1 Introduction

Engineers spend large portions of their time searching through vast amounts of corporate legacy data and catalogs searching for existing solutions which can be modified to solve new problems or to be assembled into a new device [1]. This requires utilizing databases or online listings of text, images, and computer aided design (CAD) data. Browsing and navigating such collections are based on manually-constructed categorizations which are error prone, difficult to maintain, and often based on an insufficiently dense hierarchy. Search functionality is limited to inadequate keyword scanning or matching on overly simplistic attributes.

Design repositories [2, 3] are an evolution of traditional design databases. They aim to overcome existing limitations by applying knowledge representation techniques to storing and working with artifacts in the collection. Function, behavior, rationale, and other aspects of the designs are captured and reasoned on to enable search, categorization, and other tasks in support the engineer, similar to case-based reasoning [4]. Figure 1 depicts this process. Design repositories typically also extend design databases by including more heterogenous information, incorporating such items as CAD data, documentation, simulations, animations, and analyses. Many also have the explicit goal of providing support throughout the lifecycle of a design, as opposed to focusing solely on detailed design geometry suitable for manufacturing.

This paper introduces and briefly overviews work on applying description logic and the Semantic Web to constructing design repositories. An ontology of electromechanical devices based on function and flow is used in representing, classifying, and comparing such devices. In particular, the application of least-common-subsumer and most-

---

[⋆] At the National Institute of Standards and Technology during preparation of this document.

[⋆⋆] Also of the Department of Mechanical Engineering; Email: `regli@drexel.edu`

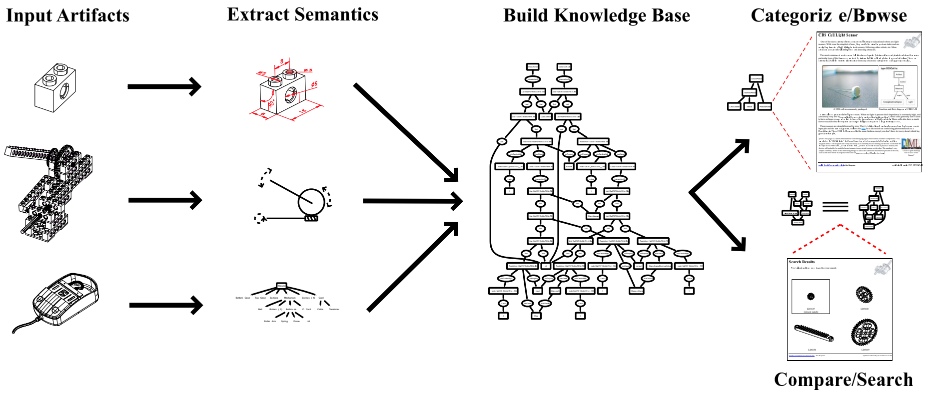


Fig. 1. The design repository process.

specific-class induction to providing design repository inferences not possible with traditional database implementations is introduced. The role of the Semantic Web in creating a new kind of public, distributed, multi-source design repository is also discussed.

The organization of this paper is as follows: Section 2 compares this effort to related work in assembly representation and design repositories. Description logic-based knowledge representation and reasoning for design repositories is described in Section 3. Section 4 discusses design repositories on the Semantic Web. Finally, Section 5 presents some closing conclusions.

2 Related Work

This section analyses work on assembly representation and repository reasoning.

Most familiar to engineers are the representations of devices used in Computer Aided Design (CAD) packages. At the core of these are the solid models of the device's components. Upon these are placed constraints in the form of analytic geometry equations describing the motions present in the mechanism [5–7]. However, such equations provide little basis for reasoning beyond simulation through constraint solving. In addition, current CAD does not typically capture abstract information such as function in any form suitable for automated reasoning or even efficient human use, nor information across multiple domains (e.g. electrical and mechanical).

Also familiar to engineering design students are notations of function as in [8]. Many representations explicitly capture the functions present in an assembly [9–11]. These typically capture function information at variable levels of abstraction and across multiple domains but lack the formal framework to support automated reasoning. Repositories using such representations often rely on simple keyword and attribute matching.

Qualitative physics and logic-based representations [12–16] attempt to define the semantics of assemblies in more abstract manners than the geometric equations employed in CAD in order to provide for richer inferences. However, these systems typically do not address many types of inference associated with design repositories, such as deter-

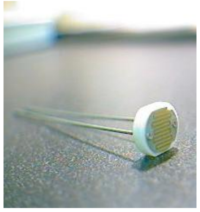


Fig. 2. Typical CDS Cell.

mining similarities between devices. In addition, the expressiveness of the languages used often incurs significant cost in terms of computability and tractability.

3 Representation and Reasoning

Description logic has been chosen as the formal framework for this work because of its favorable complexity results and inference capabilities. In order to promote efficient reasoning, careful attention has been paid to the class expression constructs used. The central ontology is simple enough to be expressed in $\mathcal{ALEN}$[1]. For simplicity, in this paper $\mathcal{EL}$ is used in the examples while in practice $\mathcal{ALEN}$ is also used for these.

For reasons of space[2], the ontology defining the representation is not given in this paper but is instead shown through examples. The core of the ontology defines a simple representation of artifacts based on function and flow [9–11]. Extensions to the core ontology define extensive taxonomies of functions and flows derived from those presented in [9] and [10], which were developed from large surveys of engineers.

Figure 2 depicts a typical artifact, a cadmium sulfide cell. These cells are used as light sensors. They are photoresistors; the presence of light decreases the resistance encountered by electricity flowing across the cell. This information can be represented as in Figure 3(a) by a *function and flow diagram* similar to those presented in [9] but not identical. By interpreting the diagrams as a set of objects and relations, they can be represented in a description logic model as in Figure 3(b).

This representation does not rigorously and unambiguously capture the semantics of mechanisms. Instead, it provides a language expressive enough to describe and distinguish devices while maintaining efficiency and computability. It is neither so formal as to prevent practical computing, nor so informal as to prohibit automated reasoning.

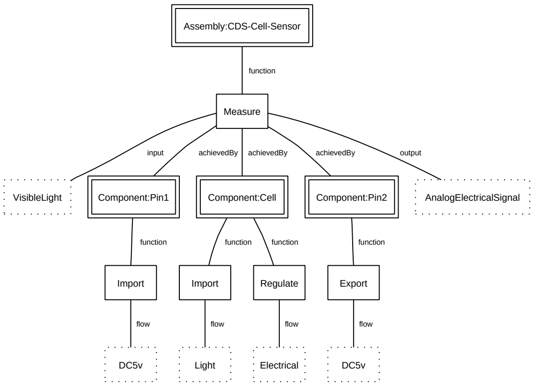
Classification of objects is a standard description logic inference. It can be used to search for devices meeting specific criterion as well as to apply a manually developed categorization scheme. For example, the class of electronic sensors could be defined as those objects which measure some input and return an electrical signal:

$$\begin{gathered}\mathit{Electronic\text{-}Sensor} \equiv \mathit{Artifact} \sqcap \exists \mathit{function}.[\mathit{Measure} \sqcap \exists \mathit{input}.\mathit{Flow} \sqcap \\ \exists \mathit{output}.[\mathit{Electrical} \sqcap \mathit{Signal}]].\end{gathered}$$

---

[1] A good reference for description logic expressivity notations is [17].

[2] (and the lead author breaking an arm in five places shortly before the submission deadline...)



(a) Simplified function and flow diagram.

⟨ ⟨ *type CDS-Cell-Sensor Assembly* ⟩
⟨ *function CDS-Cell-Sensor node7* ⟩
⟨ *function Pin2 node8* ⟩
⟨ *type node0 AnalogElectricalSignal* ⟩
⟨ *flow node1 node2* ⟩
⟨ *flow node3 node4* ⟩
⟨ *function Cell node3* ⟩
⟨ *type node2 DC5v* ⟩
⟨ *type node5 Light* ⟩
⟨ *type node6 VisibleLight* ⟩
⟨ *achievedBy node7 Cell* ⟩
⟨ *type node3 Regulate* ⟩
⟨ *type node7 Measure* ⟩
⟨ *type node1 Import* ⟩
⟨ *input node7 node6* ⟩
⟨ *type Pin1 Component* ⟩
⟨ *type Cell Component* ⟩
⟨ *achievedBy node7 Pin1* ⟩
⟨ *type Pin2 Component* ⟩
⟨ *type node9 Import* ⟩
⟨ *type node4 Electrical* ⟩
⟨ *function Cell node9* ⟩
⟨ *achievedBy node7 Pin2* ⟩
⟨ *flow node9 node5* ⟩
⟨ *flow node8 node10* ⟩
⟨ *type node8 Export* ⟩
⟨ *output node7 node0* ⟩
⟨ *function Pin1 node1* ⟩
⟨ *type node10 DC5v* ⟩
⟩

(b) Figure 3(a) as model.

$$\begin{aligned}
\mathit{CDS\text{-}Cell\text{-}Sensor} \equiv \mathit{Assembly} \sqcap \exists \mathit{function}.[&\mathit{Measure} \sqcap \exists \mathit{input}.\mathit{VisibleLight} \sqcap \\
&\exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Import} \sqcap \exists \mathit{flow}.\mathit{DC5v}]] \sqcap \\
&\exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Import} \sqcap \exists \mathit{flow}.\mathit{Light}] \sqcap \exists \mathit{function}.[\mathit{Regulate} \sqcap \\
&\exists \mathit{flow}.\mathit{Electrical}]] \sqcap \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Export} \sqcap \exists \mathit{flow}.\mathit{DC5v}]] \sqcap \\
&\exists \mathit{output}.\mathit{AnalogElectricalSignal}].
\end{aligned}$$

(c) Function and flow diagram as class description.

Fig. 3. Function and flow modeling of a Cadmium Sulfide (CDS) Cell, a common photoresistor.

The representation in Figure 3 matches this definition, provided that $\mathit{Assembly} \subseteq \mathit{Artifact}$, $\mathit{VisibleLight} \subseteq \mathit{Flow}$, and $\mathit{AnalogElectricalSignal} \subseteq \mathit{Electrical} \sqcap \mathit{Signal}$.

Determination of *subsumption* relations between classes, another standard description logic inference, can be used to organize the knowledge base for more efficient classification. A related but less common inference is induction of the *least common subsumer* of sets of classes [18–20]. The least common subsumer is the class of which each class in the given set is a subclass and for which there exists no subclass of which each class in the given set is a subclass. This is often used in conjunction with the ability to develop the *most specific class* of an object. The most specific class defines the necessary and sufficient conditions for membership in the class consisting of the given object—it is a mapping of all the properties of the given object into a class definition.

Least common subsumer inference can be used in a design repository in several ways. It can facilitate the construction of a categorization scheme in a bottom-up fashion from very specific classes; these tend to be more intuitive for users not trained in knowledge representation to create [21]. It can also be combined with most specific class induction to derive a categorization from given devices, as opposed to developing one *a priori*. The novelty of this capability in this domain is the ability to provide automatically generated, sophisticated categories for browsing/manual search, knowledge discovery of similarities, and a form of searching where the query is placed into the context of the hierarchy and the user can browse through relationships to existing models, rather than being presented solely with items which match exactly.

Figure 4(c) shows the most specific class for the cadmium sulfide cell representation in $\mathcal{EL}$, which consists of conjunctions and existential restrictions[3]. Figures 4(a) through 4(d) depict other devices and their function and flow diagrams. Most specific classes for these are shown in Figure 4(e). By inferring least common subsumers for these classes, it is possible to automatically construct a hierarchy of induced class descriptions. Classes in the hierarchy generated from these devices include at the top level $\exists \mathit{function}.\mathit{Measure}$ and $\exists \mathit{function}.\mathit{Produce}$, provided that $\mathit{Detect} \subset \mathit{Measure}$, partitioning the devices into sensors and effectors. At the other end of the hierarchy, the most specific classes of *CDS-Cell-Sensor* and *Phototransistor* will have been inferred to be subclasses of an induced class identical to the two classes, a result of their identical models and denoting a great deal of similarity between the two devices.

4 The Role of the Semantic Web

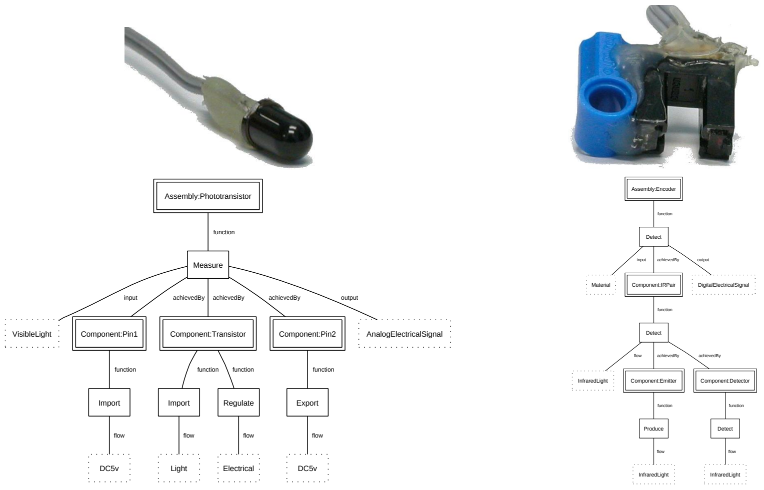
Utilizing current Semantic Web technology, design representations can be embedded inside Web resources. Figure 5(a) gives the source for Figure 3(a), encoded using the Resource Description Framework (RDF) [22], DARPA Agent Markup Language (DAML) [23], and the DAML version of the ontology described in the previous section. In this form it can be cleanly incorporated into webpages such as in Figure 5(b).

One feature of this capability is that the data sources are readily available for consumption by a wide variety of Web users. Existing search engines and directory services may index and retrieve them using standard techniques. Human users may come upon them through links from other webpages, with or without such markup. Importantly, they will be accessible to many different kinds of repositories distributed across the Internet. The information present in the markup can be easily read into a database-based system, which may offer the best performance for classification and applying *a priori* categorizations, as well as into description logic-based repositories or other systems with an enhanced set of services, such as automatically inducing categorizations.

Another feature is that the data sources can be easily published by any organization or individual with an interest in making their designs available for human Web traffic and multiple search engines, databases, and design repositories. This may enable: improved communication and resource utilization between potentially geographically distributed design groups on a company intranet; enhanced search and navigation capabilities of catalogs and listings for consumers looking for products, possibly through third-party portal sites leveraging the semantic markup to achieve increased effectiveness and coverage; design repository support for a wide range of individuals such as hobbyists and students with an interest in maintaining a collection of their own designs or those produced within an entire community of users.

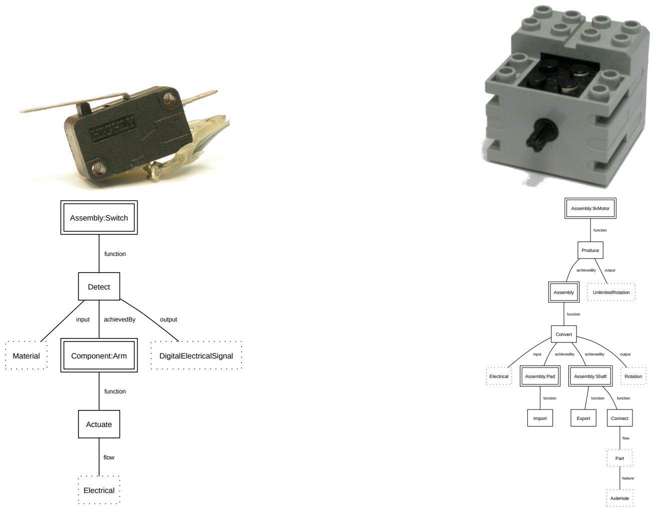
---

[3] In more expressive description logics it might not be.



(a) Phototransistor.

(b) Break-Beam.



(c) Switch.

(d) 9V Motor.

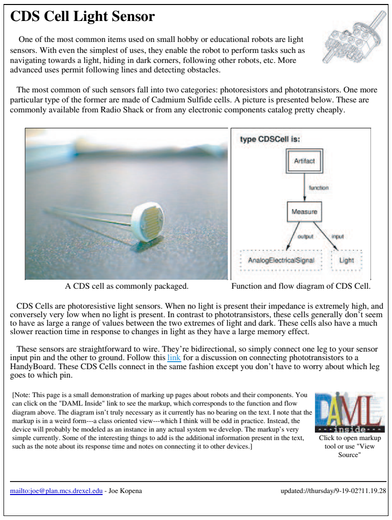
$$
\begin{aligned}
\mathit{Phototransistor} \equiv{}& \mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Measure} \sqcap \exists \mathit{input}.\mathit{VisibleLight} \sqcap \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Import} \sqcap \exists \mathit{flow}.\mathit{DC5v}]] \sqcap \\
& \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Import} \sqcap \exists \mathit{flow}.\mathit{Light}] \sqcap \exists \mathit{function}.[\mathit{Regulate} \sqcap \exists \mathit{flow}.\mathit{Electrical}]] \sqcap \\
& \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Export} \sqcap \exists \mathit{flow}.\mathit{DC5v}] \sqcap \exists \mathit{output}.\mathit{AnalogElectricalSignal}]]. \\
\mathit{Encoder} \equiv{}& \mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Detect} \sqcap \exists \mathit{input}.\mathit{Material} \sqcap \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Detect} \sqcap \exists \mathit{flow}.\mathit{InfraredLight} \sqcap \\
& \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Produce} \sqcap \exists \mathit{flow}.\mathit{InfraredLight}] \sqcap \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Detect} \sqcap \exists \mathit{flow}.\mathit{InfraredLight}]] \sqcap \exists \mathit{output}.\mathit{DigitalElectricalSignal}]. \\
\mathit{Switch} \equiv{}& \mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Detect} \sqcap \exists \mathit{input}.\mathit{Material} \sqcap \exists \mathit{achievedBy}.[\mathit{Component} \sqcap \exists \mathit{function}.[\mathit{Actuate} \sqcap \exists \mathit{flow}.\mathit{Electrical}]] \sqcap \exists \mathit{output}.\mathit{DigitalElectricalSignal}]. \\
\mathit{9vMotor} \equiv{}& \mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Produce} \sqcap \exists \mathit{achievedBy}.[\mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Convert} \sqcap \exists \mathit{input}.\mathit{Electrical} \sqcap \exists \mathit{achievedBy}.[\mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Import}] \\
& \sqcap \exists \mathit{achievedBy}.[\mathit{Assembly} \sqcap \exists \mathit{function}.[\mathit{Export}] \sqcap \exists \mathit{function}.[\mathit{Connect} \sqcap \exists \mathit{flow}.[\mathit{Part} \sqcap \exists \mathit{feature}.\mathit{AxleHole}]] \sqcap \exists \mathit{output}.\mathit{Rotation} \sqcap \exists \mathit{output}.\mathit{UnlimitedRotation}]].
\end{aligned}
$$

(e) Function and flow diagrams as class descriptions.

Fig. 4. Function and flow diagrams for several other devices.

```
<rdf:RDF
  xmlns:rdf  = "&rdf;#" xmlns:rdfs = "&rdfs;#" xmlns:daml = "&daml;#"
  xmlns:eng  = "&eng;#" xmlns:func = "&func;#" xmlns:flow = "&flow;#"
  xmlns =
 "http://edge.mcs.drexel.edu/assemblies/tests/iaai03/cds-cell.daml#">

<daml:Ontology rdf:about="#" >
  <daml:imports rdf:resource="&eng;" />
  <daml:imports rdf:resource="&func;" />
  <daml:imports rdf:resource="&flow;" />
</daml:Ontology>

<eng:Assembly rdf:about="#CDS-Cell-Sensor">
  <eng:function><func:Measure>
      <eng:input><flow:VisibleLight /></eng:input>

      <eng:achievedBy><eng:Component rdf:about="#Pin1">
          <eng:function><func:Import>
            <eng:flow><flow:DC5v /></eng:flow>
          </func:Import></eng:function>
      </eng:Component></eng:achievedBy>

      <eng:achievedBy><eng:Component rdf:about="#Cell">
          <eng:function><func:Import>
            <eng:flow><flow:Light /></eng:flow>
          </func:Import></eng:function>

          <eng:function><func:Regulate>
               <eng:flow><flow:Electrical /></eng:flow>
          </func:Regulate></eng:function>
      </eng:Component></eng:achievedBy>

      <eng:achievedBy><eng:Component rdf:about="#Pin2">
          <eng:function><func:Export>
            <eng:flow><flow:DC5v /></eng:flow>
          </func:Export></eng:function>
      </eng:Component></eng:achievedBy>

      <eng:output><flow:AnalogElectricalSignal /></eng:output>
    </func:Measure></eng:function>
</eng:Assembly>
</rdf:RDF>
```

(a) RDF/DAML source for diagram in Figure 3(a).



CDS Cell Light Sensor

One of the most common items used on small hobby or educational robots are light sensors. With even the simplest of uses, 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 Sulfide cells. A picture is presented below. These are commonly available from Radio Shack or from any electronic components catalog pretty cheaply.

A CDS cell as commonly packaged.

Function and flow diagram of CDS Cell.

CDS Cells are photoresistive light sensors. When no light is present their impedance is extremely high, and conversely very low when no light is present. In contrast to phototransistors, these cells generally don't seem 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 same fashion 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 isn't truly necessary as it currently has no bearing on the text. I note that the markup is in a weird form---a class oriented view---which I think will be odd in practice. Instead, the device will probably be modeled as an instance in any actual system we develop. The markup's very simple currently. Some of the interesting things to add is the additional information present in the text, such as the note about its response time and notes on connecting it to other devices.]

Click to open markup tool or use "View Source"

mailto:joe@plan.mcs.drexel.edu - Joe Kopena

updated://thursday/9-19-02?11.19.28

(b) Webpage with Figure 5(a) embedded in the source.

Fig. 5. Function and flow model incorporated into Web resource for cadmium sulfide cell.

5 Conclusions

This paper has briefly overviewed work on constructing design repositories for the Semantic Web. Applications of description logic inferences to achieve novel repository capabilites have been introduced. The motivations for placing such repositories on the Semantic Web have also been described. A planned future testbed of these ideas is implementation in support of a university course on small mobile robotics [24]. Current work includes the addition of non-standard description logic inferences to DAML-JessKB, a DAML-native description logic reasoner developed for this project [25], to demonstrate the utility of such inferences as applied to design repositories.

Acknowledgements: This work 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 by Honeywell FM&T, AT&T Labs, Bentley Systems and Lockheed Martin, Naval Electronics and Surveillance Systems. All opinions, findings, and conclusions expressed in this material are those of the author(s) and not necessarily those of the supporting organizations.

References

1. Ullman, D.G.: The Mechanical Design Process. McGraw-Hill, Inc. (1997)
2. Szykman, S., Bochenek, C., Racz, J.W., Senfaute, J., Sriram, R.D.: Design repositories: Engineering design's new knowledge base. IEEE Intelligent Systems **15** (2000) 48–55

3. Szykman, S., Sriram, R.D., Regli, W.C.: The role of knowledge in next-generation product development systems. Journal of Comp. and Inf. Science in Engineering **1** (2001) 3–11
4. Fowler, J.E.: Variant design for mechanical artifacts: A state-of-the-art survey. Engineering with Computers **12** (1996) 1–15
5. Lee, K., Andrews, G.: Inference of the positions of components in an assembly: part 2. Computer Aided Design (1985) 20–24
6. Hoffmann, C.M., Joan-Arinyo, R.: Symbolic constraints in constructive geometric constraint solving. In: Journal of Symbolic Computation. (1997) 287–300
7. Anantha, R., Kramer, G.A., Crawford, R.H.: Assembly modelling by geometric constraint satisfaction. Computer-Aided Design **28** (1996) 707–722
8. Pahl, G., Beitz, W.: Eng. Design—A Systematic Approach. 2nd edn. Springer (1996)
9. Szykman, S., Racz, J.W., Sriram, R.D.: The representation of function in computer-based design. In: ASME Design Engineering Technical Conferences, 11th International Conference on Design Theory and Methodology, New York, NY, USA, ASME, ASME Press (1999)
10. Hirtz, J., Stone, R., McAdams, D., S, S., Wood, K.: Evolving a functional basis for engineering design. In: ASME Design Engineering Technical Conferences, 13th conference on Design Theory and Methodology, New York, NY, USA, ASME, ASME Press (2001)
11. Kirschman, C., Fadel, G., Jara-Almonte, C.: Classifying functions for mechanical design. In: ASME Design Engineering Technical Conferences, 8th conference on Design Theory and Methodology, New York, NY, USA, ASME, ASME Press (1996)
12. Faltings, B.: Qualitative kinematics in mechanisms. A.I Journal **44** (1990) 89—119
13. Kuipers, B.: Commonsense reasoning about causality: Deriving behavior from structure. Artificial Intelligence Journal **24** (1984) 169—204
14. Forbus, K.: Qualitative process theory. Artificial Intelligence Journal **24** (1984) 85–168
15. Subramanian, D., Wang, C.: Kinematic synthesis with configuration spaces. In: Proceedings of Qualitative Reasoning 93,. (1993) 228–239
16. Joskowicz, L., Neville, D.: A representation language for mechanical behavior. Artificial Intelligence in Engineering (1996) 109—116
17. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2002)
18. Cohen, W.W., Borgida, A., Hirsh, H.: Computing least common subsumers in description logics. In Rosenbloom, P., Szolovits, P., eds.: Proceedings of the Tenth National Conference on Artificial Intelligence, Menlo Park, California, AAAI Press (1993) 754–761
19. Baader, F., Küsters, R., Molitor, R.: Computing least common subsumers in description logics with existential restrictions. In Dean, T., ed.: Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Morgan Kaufmann (1999) 96–101
20. Küsters, R., Molitor, R.: Computing least common subsumers in $\mathcal{ALEN}$. In Nebel, B., ed.: Seventeenth IJCAI, Morgan Kaufmann (2001) 219–224
21. Brandt, S., Turhan, A.Y.: Using non-standard inferences in description logics - what does it buy me? In: KI Workshop on Applications of Description Logics. (2001)
22. W3C: Resource Description Framework (RDF) model and syntax specification. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/` (1999)
23. DARPA: DAML march 2001 specifications (DAML+OIL). `http://www.daml.org/2001/03/daml+oil-index` (2001)
24. Greenwald, L.G., Kopena, J.B.: On achieving educational and research goals with small, low-cost mobile robot platforms. IEEE Robots and Automation, Special Issue on Robotics in Education (To appear 2003)
25. Kopena, J.B., Regli, W.C.: DAMLJessKB: A tool for reasoning with the semantic web. In: International Semantic Web Conference. (To appear 2003)