A Gentle Introduction to Gendl®, a Common Lisp-based Knowledge Based Engineering Environment

Dave Cooper
Genworks International
Birmingham, MI (USA)
david.cooper@genworks.com

ABSTRACT
In this two-hour tutorial we will introduce the Gendl® project (source available at http://github.com/genworks/gendl), and we will cover the following topics:

1. Installation either through Quicklisp or through a pre-built install package
2. Basic Gendl® declarative authoring syntax
3. Example Non-geometric application
4. Example geometric application (twisty tower)
5. Example geometric application with a mechanism (4-bar link)
6. Adding a web user interface
7. Building a runtime application, including integrated web server, using ASDF3’s new application delivery capabilities
8. Uploading and deploying the runtime application on a server

Categories and Subject Descriptors
I.3.5 [Computing Methodologies]: COMPUTER GRAPHICS—Computational Geometry and Object Modeling; I.2.4 [Computing Methodologies]: ARTIFICIAL INTELLIGENCE—Knowledge Representation Formalisms and Methods; D.3.4 [Software]: PROGRAMMING LANGUAGES—Processors

General Terms

Keywords
KBE, Common Lisp, CAD

1. INTRODUCTION
Gendl® is a Generative Programming and Knowledge Based Engineering framework, implementing concepts which date back to the 1980s. Gendl® allows for high-level declarative, object-oriented problem solving and application development, including but not limited to the ability to generate and manipulate 3D geometry. To solve a problem in Gendl, you formulate it using the define-object operator, which allows you to specify inputs, outputs (computed-slots), and child objects, which then gives the ability to generate a “tree” of objects, useful for decomposing complexity.

Gendl® is embedded in Common Lisp and is a superset of Common Lisp and the Common Lisp Object System.

2. INSTALLATION
First we will cover installing Gendl, using either of two approaches:

1. Using the Quicklisp curated Common Lisp library manager (if you already have a compatible CL development system installed).
2. Using a self-contained downloadable installer package.

3. BASIC GENDL SYNTAX
Next we will cover the fundamental Gendl declarative syntax, consisting of the toplevel define-object operator, mixin (superclass) syntax, and the common slot types, including:

- input-slots (required and optional)
- computed-slots
- functions
- objects (for establishing child objects and hierarchies)

4. EXAMPLE APPLICATIONS
Time permitting, we will create three example Gendl applications:

1. Example Non-geometric application
2. Example geometric application (twisty tower)
3. Example geometric application with a mechanism (4-bar link)

5. WEB USER INTERFACE
Time permitting, we will create an integrated web user interface for one or more of the applications.

1. Example Non-geometric application
2. Example geometric application (twisty tower)
3. Example geometric application with a mechanism (4-bar link)

6. UPLOADING AND DEPLOYING
Time permitting (and assuming sufficient Internet connectivity), we will build a runtime application, possibly making use of ASDF’s new application build capabilities. We will then upload it to a server on the Internet, and invite participants to test it.