An Engineering Design Language

Walter W. Wilson & Yu Lei, The Univ. of Texas at Arlington

What is the best way to represent an engineering design? In earlier times engineering designs were described on paper drawings. While inflexible, they had the advantage that well-preserved drawings could last for centuries and that engineers of the future would be able to read them. Now we store our designs in the proprietary file formats of vendor's CAD (Computer Aided Design) systems. We need a license to access our own data! It is hard to transfer high-level information between different CAD systems. An aircraft program can last for decades, a building for centuries – will our design data be accessible in the distant future? Can we have the openness of paper drawings with all the advantages of a digital representation?

One alternative for saving CAD data is STEP, the STandard for the Exchange of Product model data. But STEP has limitations. Despite its enormous size, it doesn't always fully support all the capabilities of modern CAD systems. User customizations of a CAD system cannot be saved in STEP. Thus information can be lost when writing to STEP. The problem with STEP is that it has no programmability. A data-only format is inherently limited and inflexible. Programmability gives extensibility without having to extend the standard.

This grand challenge proposes that engineering design data be represented in a textual, human-readable language. The language would store definitions of geometry instead of megabytes of triangles and surface control points. Screen images, drawings, and manufacturing information would be generated from the definitions. An interactive system could create and edit the text definitions, but they would also be directly readable and editable. The operations and data types of the engineering design language would be defined by an open-source geometric engine.

Geometry definitions in an engineering design language would be in an open, human-readable format instead of a vendor's secret binary file format. Data would be stored at a higher semantic level that would capture engineering knowledge and design intent. Text definitions would complement the screen image. An image gives fast, intuitive understanding while the text gives precise and complete information. Complete construction history and associativity is inherent in the text definitions and parametric design would be encouraged.

An open-source geometric engine for the engineering design language would give engineers understanding and control of mathematical details, in contrast to the black-box nature of commercial CAD systems. Better algorithms may result and user customizations could be incorporated. The geometric engine would be implemented in a minimal declarative language, which would be the standard, not the engineering design language, which would be free to evolve. For long-term access the geometric engine would be archived along with the design data.

One candidate for the underlying minimal declarative language would be a minimal LISP. But for this grand challenge we propose use of a specification language called "axiomatic language" (http://www.axiomaticlanguage.org/). The small size and elegance of axiomatic language would make it a good long-term standard. Its metalanguage capability would make it well-suited as a host for embedded domain specific languages like the engineering design language. Since there is no built-in floating point, explicitly-defined approximate arithmetic would guarantee that future numerical results would be identical down to the last bit, regardless of future hardware.

The challenge of axiomatic language is that its implementation requires automatically transforming specifications to efficient algorithms, which has been submitted as a separate Grand Challenge ("Automatic Programming Using Axiomatic Language"). The challenge of this engineering design language proposal is the major software development effort needed to define the open-source geometric engine. This would presumably require government funding.