COMPUTER-AIDED CONTROL SYSTEM DESIGN SOFTWARE: STANDARDIZATION *vs* DIVERSIFICATION

François E. Cellier

Dept. of Electr. & Comp. Engr., University of Arizona
TUCSON, AZ 85721, U.S.A.

Keywords: Computer-aided system design; computer software; data handling; digital computer applications; standardization.

Current CACSD tools vary drastically in terms of their application areas, as well as their user interfaces. Is such a diversification justified and desirable, or might a canalization of the various and diversified efforts into one single CACSD software standard be more appropriate? Is there any hope for a CACSD standard? How might such a standard look like?

We believe that with respect to the manner in which control problems are formulated, a standard is both feasible and desirable. The matrix notation of MATLAB-like languages where e.g. the expression:

```
rot = [cos(phi),-sin(phi),sin(phi),cos(phi)]
```

could be used to denote the matrix:

$$rot = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$$

is so natural that we do not see any need for another notation in the future. Although MATLAB's division operators "/" and "\" for right and left matrix «division» are certainly not «standard» operators in the classical mathematical sense, they are very convenient and easy to get used to. In IMPACT, a MATLAB derivate, we suggested additional operators besides "," and ";" for a third dimension, thus IMPACT effectively operates on complex *tensors* in place of complex *matrices*. Multivariable systems can be expressed in terms of polynomial matrices. The "~" operator separates polynomial coefficients, while the "|" operator separates polynomial roots. Thus, the polynomial matrix:

$$P = \begin{bmatrix} (3s^2+10s+3) & (2s-3) \\ s^3 & (-s^2-7s-10) \end{bmatrix}$$

can, in IMPACT, e.g. be coded as:

```
P = [ [3~10~3],  [-3~2]
      [~~~1],    [-10~-7~-1] ].
```

Also with respect to the embedded procedural language, a «standard» can be achieved. CTRL-C, another recent MATLAB «dialect», is very powerful in this respect. It basically extends the PASCAL programming style, operating conveniently on matrix data structures. Very useful, for instance, is the extension of the PASCAL-like "FOR"-statement:

```
FOR I=[1,3,7,28],...
```

This FOR-loop is executed four times with */=1, /=3, /=7,* and */=28,* respectively. IMPACT employs an *ADA-style* instead of a *PASCAL-style*. It actually hardly matters which style is adopted in a forthcoming standard, but a standard must be found in order to allow smooth exchange of the extensive available soft-coded macro libraries between different CACSD software systems.

Also with respect to the user communication interface, *de facto* «pseudo-standards» have already been established. Window interfaces look more and more similar to the Macintosh interface. The *Swiss mouse* is a very convenient, flexible, and fast input device. To our displeasure though, there exist «mice» with one, two, and three buttons. Any standard would be equally acceptable, but a standard must be found. Once the fingers are used to one system, it is very hard to adjust to another.

Another interface, which is rarely even noticed by the casual CACSD software user, is the interface to a *data base* where results of computations, as well as programming modules, notebook files, etc. may be stored. To promote the state-of-the-art of CACSD software further, it is imperative that a data base interface standard be defined as soon as possible. Lacking such a standard, most current CACSD software developers simply rely on the file handling mechanism (directory structure) of the embedding operating environment. This results in a jungle of small and smallest data and program files scattered over different subdirectories which makes it very hard to retrieve data and programs.

With respect to the actual functions offered, we shall probably not see a «standard» quickly. The current diversification into different application areas and design methodologies is most likely to be around for some time, and we actually welcome this, as too early a standard can freeze the lines and hamper the introduction of innovative new concepts.