In the GKS level structure, two independent axes define upwardly compatible sets of functions: level $a$, $b$, or $c$ are used for specifying the amount of input functionality; and 0, 1, or 2 are used for specifying the segment and output capabilities. The same level structuring applies to GKS-3D, where GKS (2D) forms a true subset. (Actually, this dimensionality could be regarded as another level in the structure.)

One solution, which has already been suggested by Steinhart, is simply to add a level 3 giving all the structuring facilities. This solution, however, implies that the functionality of level 2 is a true subset of level 3, a concept that may not always be welcomed.

An alternative solution is to include a third axis with two values designated as flat and structure. As compared with structures, the segments could be seen as a kind of compound primitive. This leads to function sets of level from 2D flat $a_0$ to 3D structured $2c$. From the application point of view, this solution would allow for tailored graphics support systems of much broader applicability than just level 3D structured $0c$, which probably comes closest to the current PHIGS proposal.

Such a solution would also allow a much broader range of applications for this standard. Structure facilities could be used in a 2D application, such as cartography or VLSI design. By the way, VLSI design is also a typical example for the use of structures without primitive attribute inheritance.

**Conclusions**

The issue of compatibility is of paramount importance. Every attempt must be made to find solutions to ensure the success of our current internationally cooperative efforts.

An article this short cannot possibly describe all the changes needed to bring PHIGS into line with GKS. It has, however, tried to sketch some important areas of discrepancy and give some idea as to how problems might be solved. The difficulties are not impossible to overcome. With a continuing spirit of international cooperation within WG2, we can and will successfully resolve them.

**References**