Strategic Planning Assumptions for Fortran 2000

by Craig T. Dedo
January 24, 1997

As we do strategic planning for Fortran 2000, it would be useful to take some time to consider marketing and environmental assumptions that are important to the survivability of Fortran. Often, we have skirted such issues in our deliberations. I believe that we need to face these issues straightforwardly. Following is a list of major assumptions that affect the future of Fortran. Although the tight time constraints for the upcoming joint WG5/X3J3 meeting preclude any floor debate on these issues, I believe that it is useful for those attending the meeting to think about these and related assumptions of how the programming world works during committee and subgroup deliberations.

1. Time Frame.
   a. The relevant period of time for Fortran 2000 will be 2001 - 2012.
   b. Any other time period?

2. Market Presence - Ideal. Ideally, without regard to such factors as political and economic constraints, what kind of market presence should Fortran have?
   a. Blow away all of the competition.
   b. Be second to none as the language of choice for high powered numeric applications, general purpose application development, and teaching.
   c. Be the language of choice for high powered numeric applications and have a moderate presence for general purpose application development and teaching.
   d. Become a niche language for high powered numeric applications, but with little presence elsewhere.
   e. Die a slow, lingering death.
   f. Die a quick, brutal death.

3. Market Presence - Realistic. Considering such factors as economic constraints, internal organizational politics, and the like, what kind of market presence should Fortran have?
   a. Blow away all of the competition.
   b. Be second to none as the language of choice for high powered numeric applications, general purpose application development, and teaching.
   c. Be the language of choice for high powered numeric applications and have a moderate presence for general purpose application development and teaching.
   d. Become a niche language for high powered numeric applications, but with little presence elsewhere.
   e. Die a slow, lingering death.
   f. Die a quick, brutal death.

4. Value as a teaching language.
   a. Popularity as a teaching language is important to the survival of Fortran.
   b. Popularity as a teaching language is of little importance to the survival of Fortran.

5. Value of Object Oriented Programming (OOP)
   a. OOP is an important trend in computing which will be central to computing in the 21st century.
   b. OOP is an important fad which will pass, but it will leave moderate influence on computing in the 21st century.

c. OOP is a fad which will pass and leave little influence on computing in the 21st century.

6. Human Productivity Considerations
   a. Human productivity (e.g., function points per labor hour) will become much more important in the next 10-15 years than it has in the past.
   b. Human productivity is already an important aspect of software management and will continue to be important.
   c. Human productivity, although important in other areas of software engineering, is not an important consideration for Fortran.
   d. Human productivity will be no more important in software management in the next 10-15 years than it has been in the past.

7. Choice of Programming Language Decision
   a. The choice of programming language will be based primarily on rational criteria such as suitability to the application, staff expertise, human productivity, and the like.
   b. The choice of programming language is strongly influenced both by rational and non-rational criteria.
   c. The choice of programming language will be based primarily on non-rational criteria such as fads, prejudices, peer pressure, internal office politics, and the like.

8. Access to System Level Tools and Interfaces
   a. Easy, intuitive access to system level tools and interfaces (such as operating system functions and windowing APIs) is vital to the success of programming languages.
   b. Programming languages can safely ignore access to system level tools and interfaces.

9. Processor Character Sets
   a. Support for standardized character sets will be vital to international and cross-platform computing in the next 10-15 years.
   b. Programming languages can safely ignore user needs for standardized processor character sets.

10. Ease of Application Programmer Interfaces
    a. Syntax constructs which are straightforward, intuitive, and easy to use encourage the popularity of the features that they support and are vital to the success of a programming language.
    b. Syntax constructs which are complex, awkward, confusing, or hard to use have little influence on the use of the features which they implement or the success of a programming language.

11. Size of the Language
    a. A programming language can thrive if it is a large language with a rich feature set.
    b. Large languages with rich feature sets are unwieldy dinosaurs which are doomed to extinction, due to resulting cost and complexity.

[End of 97-115]