

ANS FORTH: Standardising a Moving Language

Dr. Peter J. Knaggs

`pjk@bcs.org.uk`

School of Design, Engineering and Computing,
Bournemouth University, Talbot Campus,
Fern Barrow, Poole, Dorset. UK BH12 5BB

August 26, 1998

Abstract

Developing a standard for a computer language has never been easy, especially for extendible languages. Everybody wants to see their “standard” addition included, without regard to others. The standardisation process is not the place to make recommendations for extensions to the language, but rather to consolidate existing practice.

Thus, I would like to suggest we adopt a three-stage process. In which additions and extensions to the language are accepted. The standardisation committee discusses them and they are either accepted for trial or rejected. Accepted suggestions are placed on an official notice board for developers to adopt and/or experiment with and comment on.

After a suitable period, no less than four years, the committee will take the trial suggestions (and comments) and consider them for inclusion in the standard. This will make the whole process simpler, as most of the argument will have occurred during the review period. Hence the standardisation process becomes what it should be, one of consolidating current practice.

1 Introduction

As anyone involved in the process will know, creating a standard is a very difficulty and time consuming process. The ANS Forth standard required just 23 meetings of the Technical Committee (TC) last-

ing for a total of 88 days. The resulting standard was a concise document with just 124 pages for the definition of the core language and its standard libraries [ANS94], compared to some 256 pages for the definition of the Ada '95 core language with an additional 207 pages for the standard libraries [ISO95a]. The Ada '95 standard required an additional book of approximately 430 pages giving the rationale [ISO95b] as opposed to Forth's 52 pages of rationale and 32 pages of additional informative material. The Forth standard is just 208 pages (both Normative and Informative) compared to Ada's 900 or so pages.

Unlike most languages Forth faces a peculiar problem in that it is one of the few languages that supports interactive development. This leads to an interesting position, in that most advanced Forth programmers will either have their own private language extension's that they would like to see become standard or indeed have developed a personal version of the language. Whilst it is true that other languages have to face this situation, variants in the language—this is after all the reason a standard is being producing, they seldom have as many different variants to take into consideration.

As each developer has their own language extension's that they consider to be standard they recommend their extension to the technical committee, for adoption into the standard. The purpose of a standard is to “*consolidate existing practice*” where this is possible. The adoption of private extensions of an individual developer, no matter how

well developed, can not be considered to be consolidation of existing practice.

2 A Solution

The main problem with these private extension's is that they have been developed over time and normally present a well thought out approach to a particular problem, it's just that they are not in common usage. There is also a question as to what should be done to cater for those areas of the standard that were left unaddressed due to a lack of consensus.

The procedure's provide for the case where the (TC) are unable to establish a standard due to lack of consensus [NCI97]. Under these rules, it is possible for the TC to recommend the creation of a Task Group (TG) to investigate the area concerned, generating a Technical Report (TR) giving a possible standard. Assuming the community accepts the report there will be a consensus when the standard comes around for review. There are a number of areas that fall into this category, these include but are not limited to:

- Internationalisation/Localisation
- Graphics
- Object Oriented Programming
- Event Handling
- Exception Handling
- Programming Languages Interface
- Multitasking
- Portable Libraries

Given that there is this procedure for investigating large areas perhaps the procedure could be adapted for dealing with individual developers private extensions.

3 Three Phase Process

In this way it would be possible for the individual developer to effect the standard by having their

own extension's accepted, although after a period of public review. This is in addition to any TGs the TC may wish to convene to address particular issues. A three stage process is suggested which would allow for the continued development of the standard.

3.1 Stage One

Developers make their suggestion on an ad-hoc basis, giving details of their suggestions, possible methods of implementation, and the rationale. The developer may choose to publish this information at conference, as a paper in "*The Journal of Forth Application and Research*", as an article in "*Forth Dimensions*", or independently, but they must also send it to the TC.

These "candidate proposals" will be posted on a public notice board for all to review¹. Once a year the TC will vote on the candidate proposals, deciding whether to accept them or not. The authors of rejected candidate proposals will receive a communication from the TC giving the reasons for the rejection.

3.2 Stage Two

Accepted proposals are put out to public review/comment, via the notice board. The period of public review can be no less than four years. In response to the public comment the author may wish to withdraw or re-draft the proposal. Re-drafted proposals will be *required* to start their period of public review anew.

3.3 Stage Three

When the review of the standard comes around on its five-year cycle the TC will review those proposals that have survived the minimum review period. The author may be asked to re-draft the proposal in more suitable language. The TC may ask for final comments relating to these proposals, before

¹It is proposed that this "public notice board" is part of a proposed extended ANS Forth web site. (<file://ftp.uu.net/vendor/minerva/uathena.htm>)

considering them for incorporation into the standard.

4 Summary

Forth faces a peculiar problem in that it supports interactive development. A consequence of this is that each developer has their own extensions, which they consider to be standard, they would like their extension accepted into the standard. The standardisation procedures allow for the formation of Task Groups to make recommendation's as to the future direction of a standard, but this should be kept in reserve for the larger area of difference (Multitasking or Object Oriented Program for example).

A three-stage process outwith the official procedure is proposed:

- Developers put forward “candidate proposals”. These proposals are reviewed by the TC, which will either accept or reject them.
- Accepted proposals are put out to public review for a period of at least four years.
- The TC will review the surviving proposals as part of the standard five-year cycle, accepting them into the standard or not depending on the public comments.

Provided this proposal is acceptable to the TC it will provide the community with a method of extending the standard, yet allowing the standard to keep its official role as a description of existing practice.

The five-year review cycle built into the official procedures, along with the proposed procedure will allow for a continuously developing, or rolling, standard.

References

- [ANS94] American National Standards Institute. *ANS for Information Systems—Programming Languages—Forth*, 1994. ANSI X3.215-1994.
- [ISO95a] International Organisation for Standardisation. *Information Technology—Programming Languages—Ada: Reference Manual*, 1995. ISO/IEC 8652:1995(E)—RM95;6.0.
- [ISO95b] International Organisation for Standardisation. *Information Technology—Programming Languages—Ada: Rationale*, 1995. ISO/IEC 8652:1995(E)—R95;6.0.
- [NCI97] *Organization, Rules, and Procedures of NCITS*, July 1997. NCITS/SD-2. <http://www.ncits.org/sd2rev6.pdf>.