Possible Enhancements for Aldordoc

Bálint Joó and Tony Kennedy

July 18, 2001

Abstract

Aldordoc is an application, that extracts documentation from Aldor programs – in particular `.asy` files. It produces XML output which can then be converted to various display formats, one of the current supported being HTML. The appearance of the HTML output however is quite basic, and is not easy to customise. Furthermore it is unclear at this point how mathematical notation is to be embedded into Aldor documentation comments. This document investigates some of these issues

1 Introduction

The Aldor programming language currently supports two kinds of code comments, one intended to facilitate documenting code behaviour in the source programs and one intended to provide user (reference) documentation in the spirit of literate programming. In languages such as C, this second kind is not supported as part of the language, however there are special applications such as Doxygen which allow the combination of code and documentation in some easy way, often by inserting special tags into program comments. Another example for this is Java and Javadoc.

Aldor is particular in the sense that during compilation, the documentation comments are not removed until quite a late stage, perhaps to allow documentation to be extracted from compiled Aldor code, for example from `.asy` (abstract syntax) and potentially also from `.ao` (abstract object) and `.al` (Aldor Library?) files.

Currently there is one known application, Aldordoc which extracts documentation from Aldor programs. It works with `.asy` files and produces XML and HTML output. Although Aldordoc itself is a very useful application program, it is clearly a first attempt at extracting documentation from Aldor programs. It has several limitations, the most important of which we list below:

- It produces fairly dull HTML, which is completely uncustomisable by the code documenter.
- It currently lists every single member of a particular domain or category. Often it is desirable for the documentation user to view a particular feature or part of a category or a domain, when the full detail is distracting.
- It seems (at least my copy) to work solely with `.asy` files. Often to get `.asy` files, programs that are already compiled to `.ao` files or into libraries need to be recompiled. This can create annoyances as existing `.ao` files become out of date etc.

So much for the current implementation of Aldordoc. Let us now turn our attention to a somewhat more subtle question. Often the documentation in Aldor code needs to include mathematical expressions (axioms of category members for example). This is eminently reasonable for a language for symbolic algebra computations. The current documentation style involves only plaintext (ASCII) characters. Currently one can embed XML, and HTML into Aldor documentation, and one would expect that perhaps this way mathematics may be expressed (in the worst case by including images containing the mathematics).

Using markup tags in the documentation also allows one to place images and even references to web pages or external documents into Aldor documentation, although if this is done the source and the documentation necessarily become divorced. On the other hand this may be more convenient then encapsulating encoded images or documents into a program source.

In this document we shall discuss both these issues in turn.

2 Aldordoc and Style Sheets

We believe one possible solution to the problems of Aldordoc, is for the program to be able to access some kind of style sheets.

The user could specify the style sheet file when running Aldordoc. The style sheet could then contain information as to how to render the documentation. One would need the XML produced by aldordoc to contain numerous tags, which could be given styles by the style sheet.

Let us consider some of our earlier mentioned shortcomings and discuss how some kind of style sheet mechanism could solve the problems they pose.

Dullness of Generated HTML: This is simple, style sheets could easily define colours of headings, text, links as well as backgrounds etc.

Contents of categories/domains: Style sheets could be used to hide inherited documentation in the final markup. They could also be used to allow the user to examine various parts of categories. In particular we are thinking of situations, where a category is a `Join` of many others. The style sheet could specify how the sub-categories could be displayed, for example in the full gory detail of the current program, or through a list of links to the documentation of each sub-category/domain. For the really adventurous, using Dynamic HTML, one could even create dynamically expandable/collapsible lists of exports.

Many Documentation Files: When there are many documentation files, and one needs to use the aldordoc–index program to create a table of contents to all the documentation available. Currently one can give aldordoc–index a title string. Style sheets could do much more (background, colours, perhaps even alphabetical ordering, specification of what title to use for each HTML file etc)

One question remains: "Which particular style sheet mechanism to use?" We believe that the first and last of the cases above could easily be implemented using the current standard *Cascading Style Sheet* (CSS) technology. The second one may be trickier. We don't have a deep enough knowledge of cascading style sheets to answer it in full at this time, however we offer up this topic for discussion.

3 Mathematical Expressions in the Documentation

We now turn away from the issue of creating documentation from the source code, and turn to our second topic, namely how to include mathematical expressions into Aldor documents.

This issue is somewhat different from the issue of linking to images or other documents from the documentation. In the latter case, one is often considering a reference to files which may be large compared to the size of the Aldor program source file and which, as often as not, may not even be human readable. It may make very little sense indeed to consider encapsulating such references into the program source and we believe the approach of external links suffices here.

However, in the former case, one is not looking to encapsulate an external file, and ideally one would like the source for the documentation to be reasonably human readable. Mathematical expressions in documentation are by nature sporadic and also varied, ranging from simple vector notation say, to a full mathematical theorem followed by a proof.

Unfortunately, mathematical notation is rather poorly supported by HTML. Recently the MathML standard has arisen which can in principle serve as the method of encapsulation. However although composed of ASCII characters, MathML may be quite difficult to read, due to the numerous and often lengthy attributes its tags contain. Let us take the case of a complex number: one needs to encode effectively two numbers (real and imaginary parts of the complex number). One way of doing this in MathML is to use the source:

```
<cn type="complex-cartesian">2.0<sep/>4.0</cn>
```

which is a bit of a mouthful for writing $2.0 + 4.0i$. Furthermore there are currently two flavours (physics pun?) of MathML, presentation MathML and structural(?) MathML. Which one should we use? Most current browsers only render the presentation subset.

One particular document preparation, which is widely used by the mathematics and physics community is LaTeX. It offers a very rich structure for expressing mathematical notation in a reasonably compact manner. Generally, people who use latex a great deal (such as mathematicians and physicists) can even read simple latex equations without even compiling and rendering them. All in all LaTeX has all the desirable features needed to encapsulate mathematical notation in Aldor documentation.

A way to add LaTeX to Aldor documentation may simply be to define tags such as `<latex>` `</latex>` between which the user could place arbitrary latex source. This could also be converted to HTML using some variant of the `latex2html` program. One could then include even long proofs as part of the documentation. Indeed the whole documentation could be done in LaTeX– which I believe now has support for external links (is this true Tony?) – in which case we would have come full circle and returned to the style of literate programming advocated by tools such as *Web* and *NoWeb*.

4 Conclusions / Topics for Discussion

Aldor is very considerate in providing special comments for the inclusion of documentation into program sources. However no kind of official markup specification currently exists for describing the layout/structure of the documentation.

What we believe is needed is a standard set of markup tags, to describe documentation structure. Such a set of tags could open the way to the usage of style sheets to describe the look and feel of the final rendered documentation. It also has the potential for standardising links to external sources of documentation as well as to the inclusion of extra markup to describe mathematical notation.

We believe LaTeX is a great system to use to mark up mathematical notation in Aldor documentation. Although MathML is also an excellent potential candidate, we prefer LaTeX for its compactness, and also because many mathematicians, physicists and other scientists (who are not necessarily Web buffs) have a more than passing acquaintance with it.

5 Acknowledgements

Bálint Joó would like to thank Delta Air Lines for fuelling him with coffee and rattling him about in their Boeing 767 while writing this document.