CZT: Community Z Tools Initiative

January 2002 for ZB2002

http://web.comlab.ox.ac.uk/oucl/work/andrew.martin/CZT

The Context

One of the biggest barriers to the widespread use of Z seems to be the issue of tool support. Many tools have been constructed, some of a product quality, and most as student projects. Few of them are integrated with each other; fewer still build together to form the kind of integrated environment that developers are beginning to expect. Many good ideas have been developed to prototype stage, and then have been lost as projects have finished and students have moved on. Many commercial projects have embodied huge amounts of research, much of it lost when they have not made money in a suitable time-frame.

Whenever someone wants to experiment with ideas for Z tools, they face the hurdle of constructing parsers, interfaces, pretty-printers, etc. Such things are frequently requested in the Z newsgroups. Because those things are seldom research in their own right, they may receive a minimum of attention, contributing to a lack of usability, stability, interoperability and long-term viability.

The Z Standardization activity has done much to develop a description of the language under discussion. It has, however moved the goal-posts in that Standard Z is a richer language than, say, the language of Spivey's Z Reference Manual. Most existing tools support only the latter, and so give no opportunity to use arbitrary sets of bindings as schemas, say. The Standard says much about what the language is, but naturally says relatively little about how to construct tools for it.

CZT Initiative

CZT aims to address some of these issues by making possible a shared collection of tools or tool components which can be picked up and used by researchers and project students. These will necessarily use standard, widespread technologies: Java; XML; Unicode, but should also aim to make possible the integration of pre-existing tools written in a wide variety of languages. Support for the LaTeX mark-up used by *f*UZZ, `oz.sty`, Z/EVES, etc. is of course essential. Most of the core software will be under some free licence.

The home page for CZT is given at the head of the page. A mailing list has been established, and that page describes how to join and how to write to the list.

Progress

CZT was proposed at the end of September 2001. To date, 36 people have joined the mailing list, and there has been some discussion on how to proceed. Based on that discussion, I have written a requirements document, which is presently under review. The next stage — perhaps the most important — will be to come up with a design which makes possible the component-based development of tools, allowing various projects to contribute parts of the solution.

The requirements have been grouped into a number of 'Levels'. Level 0 entails the design and identification of file formats and programming interfaces. Level 1 will be a collection of pieces of infrastructure, and Level 2 the beginnings of an integrated GUI.

I initially conceived CZT as 'the CZT project', but Roger Jones and others pointed out that several projects may be involved, so realising the nature of the task, *initiative* seems a better term. Several pre-existing projects have offered input, including ZEUS and ZETA.

Moving Forward

It is important that the initiative be owned by the Z Community: buy-in from a large group of people is important to its success. It needs to out-live individual projects, research students, etc., so here are some questions for discussion:

1. Are things moving in the right direction? Is progress at the right speed? Does the requirements document say the right things? What further infrastructure is needed?

2. It looks as though the main source of effort will come from student projects. In order to make this practical, and for those projects eventually to work together, much effort is needed in the design phase from experienced tool-implementors. Some of the details might be worked out (or tested) in projects, but the main interface classes, file formats, etc. probably need a more experienced eye. Is this analysis correct?

3. Is it research? Are there sources of funding to be explored to provide a co-ordinator for the initiative and its core design? I cannot imagine how to cast this as the kind of research likely to be funded, but would happily participate/apply/supervise a research officer, if the right angle can be found.

4. Should be mailing list and web page be supplemented by periodic real meetings? I would be happy to host a meeting in Oxford, or maybe could arrange a meeting as a satellite of FME2002 in Copenhagen in July.

5. Z has always been an organic and developing language. How closely should CZT be tied to the Z Standard? It will want to take account of other dialects, such as Object-Z; how important is it to do so?

Andrew Martin
Oxford University Software Engineering Centre
`Andrew.Martin@comlab.ox.ac.uk`