Report on CZT discussions at ZB2002

Ian Toyn, ian@cs.york.ac.uk

28 January 2002

1 Format of meeting

Mark Utting introduced CZT at the end of the first day, where several delegates expressed their opinions. Discussion continued over dinner that evening, where Mark was joined by Neil Robinson and Ian Toyn. I subsequently wrote this report on the discussions, and was invited by Jonathan Bowen to talk about it for a few minutes in the closing session at the end of the conference.

2 What tools do we envisage?

We hope that CZT[2] will enable the inter-operation of at least the following tools.
Mark-up to Unicode translators (one per mark-up)
Editors (of mark-up and Unicode)
Unicode character database preprocessor (for efficiency of lexer)
Lexer and parser
Typechecker
Browser/WYSIWYG interface
Logical inferences, including schema expansion, animation, etc
Test data generation
Translators (to/from smv, alloy, B, Haskell, Prolog, etc)
XML generator
XML parsers (extracting annotated tree or unannotated tree)
XML stylesheets (prettyprinting for human, projecting Unicode for lexer)

3 Overall approach

The initial suggestion to use Java throughout CZT was perceived to be at odds with the scarcity of available resources. Greatest productivity in the short term is likely to result from improving interoperability of existing toolsets. This means:

- creating potential for interoperability by adapting existing tools to use ISO Standard Z[4];
- devising a usable interchange mark-up—probably an XML[1] one.

Those resulting tools and interchange mark-up should enable useful contributions by small scale student projects. Meanwhile, others can re-implement everything in Java if they wish, but even Java users should be able to reuse some existing Z tools. Research projects seem likely to contribute only indirectly.

4 ISO Standard Z

ISO Standard Z has been developed over a long period by experts from both industry and academia. It captures the consensus of the Z community for what is wanted. It includes innovations such as a more

liberal syntax. Differences between traditional Z[3] and ISO Standard Z are summarised in [5], with the more recent revisions to process mark-up via Unicode being detailed in [11]. Conformance of tools to ISO Standard Z should be encouraged. Tools can support extensions to ISO Standard Z, though uses of such extensions are likely to suffer interoperability restrictions.

5 XML mark-up

An XML DTD has been drafted[8] and is being reviewed. The requirements listed in that paper, and others revealed by discussions at ZB2002, include:

- ability to interchange arbitrary annotations anywhere;
- formulation in terms of commonalities as used in annotated syntaxes;
- injectivity—ability to resurrect recognisable concrete phrases;
- avoidance of apparent variable capture;
- ability to exchange fragments of Z specifications.

The main comment so far has been to use an XML Schema instead of a DTD. Further comparisons with the annotated syntaxes used in other tools are needed to ensure that the XML mark-up will be usable.

The problem of apparent variable capture is illustrated by this example.

$$[X]$$

$$\forall X : \mathbb{P}\, X;\ a : X \bullet X = \{a\}$$

The use of equality is a generic instantiation expression whose instantiation is inferred to be the carrier set of the type of its arguments, i.e. the relational predicate becomes $(X, \{a\}) \in (\_ = \_)[X]$. The new reference to X should refer to the given type, but that is not how a parser would interpret it: the new reference is apparently captured by the universally quantified declaration of X. An annotated syntax, and an interchange mark-up, must avoid that capture.

This specific problem has been handled in the Z standard and within some tools by distinct renamings of global declarations (and references to them).

Apparent variable capture can also result from logical inferences, as in this example.

$$x \in \{x : \mathbb{A} \bullet x * 2\} \quad \Longrightarrow \quad \exists\, x : \mathbb{A} \bullet x * 2 = x$$

If such a logical inference is to be allowed (as it is by CADiZ[6]), then this shows that renaming merely of global declarations is insufficient. Renaming any declaration (and references to it) whose scope encompasses any references to declarations of larger scope is sufficient to avoid all apparent variable captures. Within CADiZ, that renaming is done not by the logical inference rule but by the prettyprinter on presentation of the result to the user. Its annotated syntax remembers the original names, cross references of uses with declarations, and the names used by the prettyprinter.

It might be worth noting that annotations that refer to types always refer to global declarations, and so there need never be any apparent variable capture on uses of types.

For cross referencing uses with declarations, it might be appropriate for the XML mark-up to have numeric attributes representing cross reference annotations on each declaration and each reference. Given those annotations, the renaming discussed above is not necessary. One reason for preferring renaming would be if there was an expectation of being able to parse XML mark-up without paying any attention to annotations.

We should research existing XML applications to see if any would be suitable for marking-up Z. DocBook was suggested by Wolfgang Grieskamp, but has not been investigated yet. We might also

consider whether the XML DTD/Schema can be generalised to accommodate the mark-up of notations other than Z, e.g. B.

The cost of generating and parsing XML is such that it seems unlikely to be used for communication within any single toolset. Toolset-specific APIs are likely to be much more efficient. The role of XML is expected to be for communication between toolsets. Nevertheless, we should allow for interchange of fragments of specifications, as might be desired within toolsets.

6 Existing resources

A repository of reusable Z sections has been established[9]. A yacc grammar for parsing the concrete syntax is available[7]. Release 4.0 of the CADiℤ toolset[6] includes: a translator of LaTeX mark-up to Unicode; a Unicode character database preprocessor; a generator of XML mark-up.

7 Date of next meeting

CZT seems to be making good progress electronically at the moment. The next meeting seems likely to be in Copenhagan at FME, July 20-24, 2002.

References

[1] E.R. Harold and W.S. Means. *XML in a Nutshell*. O'Reilly, 2001.

[2] A.P. Martin. Community Z tools initiative. http://web.comlab.ox.ac.uk/oucl/work/andrew.martin/CZT/, 2002.

[3] J.M. Spivey. *The Z Notation: A Reference Manual, 2nd editon*. Prentice Hall, 1992.

[4] I. Toyn. Information technology—Z formal specification notation—syntax, type system and semantics. ISO/IEC 13568:2002.

[5] I. Toyn. Innovations in the notation of Standard Z. In *ZUM'98: The Z Formal Specification Notation*, LNCS 1493. Springer, September 1998.

[6] I. Toyn. CADiZ web pages. http://www-users.cs.york.ac.uk/~ian/cadiz/, 2002.

[7] I. Toyn. A grammar for parsing Standard Z concrete syntax. In *Z Resources* [10].

[8] I. Toyn. An interchange mark-up for Standard Z: XML DTD. In *Z Resources* [10].

[9] I. Toyn. Toolkit sections: repository. In *Z Resources* [10].

[10] I. Toyn, editor. *Z Notation: Index of Resources*. http://www-users.cs.york.ac.uk/~ian/zstan/, 2002.

[11] I. Toyn and S. Stepney. Characters + mark-up = Z lexis. In *ZB 2002: Formal Specification and Development in Z and B*, LNCS 2272. Springer, January 2002.