Application program interface

/Tutorials

A library of routines is provided to enable a program to run cadiz, to receive a copy of cadiz's data structures representing the typechecked Z specification, and to try applying any Z-related cadiz command to any formulae in the specification.

The cadiz windows are unlikely to be wanted with such programs, but the option to have them exists and has proved to be useful while debugging such programs.

The first use of this API was in a testing tool, where the validity of test data obtained from running an operation was determined automatically. The conjecture that the test data is a member of the schema that specifies the operation was generated, and tactics were applied to decide those conjectures. The forms of the conjectures were limited, allowing the tactics to act as completely automatic decision procedures.

Here is an abbreviated fragment of C code showing how to use the API.

```
#include "api.h"
int
main(int argc, char *argv[])
{
        /* cadiz_init returns when the Z spec has been typechecked. */
        switch (cadiz_init(BY_PIPE, WHOLE_SPEC, "cadiz", "-v", "-l", argv
case CORRECT:
```

```
break;
case WRONG:
                fprintf (stderr, "%s: some errors detected\n", argv[0]);
                exit (1);
/*NOTREACHED*/
case QUITTED:
fprintf (stderr, "%s: traversal preempted by user-requested exit\n", argv
exit (0);
/*NOTREACHED*/
default:
eject ("Unknown return from cadiz_init");
/*NOTREACHED*/
        }
        /* cadiz_init() sets some global variables, including setting */
        /* documents to point to a local copy of the annotated syntax tre
        for (; documents != nil; documents = tail(documents)) {
                /* This is the start of a traversal of the tree. */
                /* cadiz_command(cmd) applies a Z-related command. */
                /* The cmd value encodes both operator and operands. */
                /* cadiz_command(cmd) returns the list of paragraphs that
                /* an interactive cadiz would append to the left window.
        }
        /* cadiz_kill() terminates the cadiz process. */
        cadiz_kill();
        exit (0);
}
```

The form of the data structures can be determined at two levels: first, by inspection of the data type definitions in the `lib/types` directory; second, by browsing a particular data structure using the peat-generated `cellbrowse(infp, outfp, cell)` function.

The reference manual provides details of the application program interface.

IT 24-Apr-2002