An Event-B Specification of

Library

This project tests extracting information out from a machine through parameters prefixed with out_.

1 MACHINE Library **2**

1.1 *books borrowers loans* . . . 2

1.2 addBook(*b*) . . . 2

1.3 addBorrower(*b*) . . . 2

1.4 addLoan(*book borr*) . . . 2

1.5 returnBook(*book*) . . . 3

1.6 isBookOnLoan(*book out_onloan*) . . . 3

1.7 whoBorrowsBook(*book out_borrower*) . . . 3

MACHINE Library 1

VARIABLES 1.1

books	All books that are owned by the library.
borrowers	All borrowers registered at the library.
loans	All books that are loaned out.

INVARIANTS

inv1:	$books \in \mathbb{P}(\mathbb{N})$	We represent books
inv2:	$borrowers \in \mathbb{P}(\mathbb{N})$	and borrowers using integers.
inv3:	$loans \in books \pfun borrowers$	A book is only loaned to one borrower at a time.

EVENT INITIALISATION

THEN

init1:	$books := \varnothing$
init2:	$borrowers := \varnothing$
init3:	$loans := \varnothing$

END

EVENT addBook 1.2

Add a new book to the library, the book must not have been added before.

ANY

b

WHERE

grd1:	$b \in \mathbb{N}$
grd2:	$b \notin books$

THEN

act1:	$books := books \cup \{b\}$

END

EVENT addBorrower 1.3

Add a new borrower to the library, the borrower must not have been added before.

ANY

b

WHERE

grd1:	$b \in \mathbb{N}$
grd2:	$b \notin borrowers$

THEN

act1:	$borrowers := borrowers \cup \{b\}$

END

EVENT addLoan 1.4

Loan a book to a borrower, the book must not be on loan already.

ANY

borr
book

WHERE

grd1:	$borr \in borrowers$	Valid borrower.
grd2:	$book \in books$	Valid book.
grd3:	$book \mapsto borr \notin loans$	Not a necessary test, but used for this example anyway.
grd4:	$book \notin \mathrm{dom}(loans)$	The book is not loaned out already.

THEN

act1:	$loans(book) := borr$	Add a new loan in the storage.

END

EVENT returnBook 1.5

Return a book, the book must be on loan.

ANY

$book$

WHERE

grd1:	$book \in books$	Valid book.
grd2:	$book \in \mathrm{dom}(loans)$	The book is on loan.

THEN

act1:	$loans := \{book\} \lhd loans$	Remove the loan from storage.

END

EVENT isBookOnLoan 1.6

Check if a book is on loan.

ANY

$book$
out_onloan

WHERE

grd1:	$book \in books$
grd2:	$out_onloan = \mathrm{bool}(book \in \mathrm{dom}(loans))$

END

EVENT whoBorrowsBook 1.7

Return who is borrowing a book.

ANY

$book$
$out_borrower$

WHERE

grd1:	$book \in books$	Querying a valid book?
grd2:	$book \in \mathrm{dom}(loans)$	That is on loan?
grd3:	$out_borrower = loans(book)$	Return the result through out.

END

addBook, 2
addBorrower, 2
addLoan, 2

books, 2
borrowers, 2

INITIALISATION, 2
isBookOnLoan, 3

Library, 2
loans, 2

returnBook, 3

whoBorrowsBook, 3