## An Event-B Specification of

## Library

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

		NE Library	
		books borrowers loans	
-	1.2	${\it addBook}(b) \ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	
-	1.3	addBorrower(b)	
		addLoan(book borr)	
		$\operatorname{returnBook}(book)$	
-	1.6	isBookOnLoan(book out_onloan)	
-	1.7	whoBorrowsBook(book out_borrower)	

```
1
MACHINE Library
                                                            10a
                                                                                                           1.1
VARIABLES
 books
              All books that are owned by the library.
 borrowers
              All borrowers registered at the library.
              All books that are loaned out.
 loans
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 \rightarrow borrowers
                                          A book is only loaned to one borrower at a time.
EVENT INITIALISATION
THEN
            books := \varnothing
 init1:
            borrowers := \emptyset
 init2:
 init3:
            \mathit{loans} := \varnothing
END
                                                                                                           1.2
EVENT addBook
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
                                                                                                           1.3
EVENT addBorrower
Add a new borrower to the library, the borrower must not have been added before.
ANY
 b
WHERE
           b \in \mathbb{N}
 grd1:
           b \notin borrowers
 grd2:
THEN
           borrowers := borrowers \cup \{b\}
 act1:
END
EVENT addLoan
                                                                                                           1.4
Loan a book to a borrower, the book must not be on loan already.
ANY
 borr
```

book WHERE

```
Valid borrower.
          borr \in borrowers
 grd1:
 grd2:
          book \in books
                                 Valid book.
          book \mapsto borr \notin loans
                                 Not a necessary test, but used for this example anyway.
 grd3:
          book \notin dom(loans)
                                 The book is not loaned out already.
 grd4:
THEN
 act1:
          loans(book) := borr
                                Add a new loan in the storage.
END
                                                                                                  1.5
EVENT returnBook
Return a book, the book must be on loan.
 book
WHERE
          book \in books
                               Valid book.
 grd1:
          book \in dom(loans)
                               The book is on loan.
 grd2:
THEN
 act1:
          loans := \{book\} \triangleleft loans Remove the loan from storage.
END
EVENT isBookOnLoan
                                                                                                  1.6
Check if a book is on loan.
ANY
 book
 out\_onloan
WHERE
          book \in books
 grd1:
          out\_onloan = bool(book \in dom(loans))
 grd2:
END
                                                                                                  1.7
EVENT whoBorrowsBook
Return who is borrowing a book.
ANY
 book
 out\ borrower
WHERE
 grd1:
          book \in books
                                        Querying a valid book?
 grd2:
          book \in dom(loans)
                                        That is on loan?
          out\_borrower = loans(book)
                                        Return the result through out.
 grd3:
END
```

 $\begin{array}{c} {\rm addBook,\,2} \\ {\rm addBorrower,\,2} \\ {\rm addLoan,\,2} \end{array}$ 

books, 2 borrowers, 2

 $\begin{array}{c} {\rm INITIALISATION,\; 2} \\ {\rm isBookOnLoan,\; 3} \end{array}$ 

Library, 2 loans, 2

 ${\rm returnBook},\,3$ 

who<br/>BorrowsBook,  $3\,$