MODULE *IdemProxy*

Expectations :
- Every request must be hit the server exactly once
- For each try of a request, the response must be the same
- 2 requests must be processable in parallel

EXTENDS $Integers, FiniteSets, Sequences$

CONSTANTS

$\_ReqIDs,$ the request *IDs* sent by the user (an *Id* is the idempotent key)

$\_MaxTries$ how many times a request is retried

ASSUME $\_MaxTries < 10$

VARIABLES

$requests$ the state of all requests and their corresponding tries

$vars \triangleq \langle requests \rangle$

tryKeys is a simple helper providing the set of try keys

$tryKeys \triangleq 1 \,..\, \_MaxTries$

Initial State

$Init \triangleq$

requests is a struct with *_ReqIDs* as keys associated with structs with *tryKeys* as keys associatied with the given try current status

$\land\ requests = [req \in \_ReqIDs \mapsto [st \in tryKeys \mapsto \text{"pending"}]]$

$TypeInvariants \triangleq$

$\land$ TRUE todo

Actions

$HitProxy(r,\ i) \triangleq$

$\land\ requests[r][i] = \text{"pending"}$

$\land\ requests' = [requests \text{ EXCEPT } ![r][i] = \text{"submitted"}]$

$HitServer(r,\ i) \triangleq$

$\land\ requests[r][i] \quad = \text{"submitted"}$

$\land\ Cardinality(\{x \in \text{DOMAIN } requests[r] : requests[r][x] = \text{"processed"}\}) = 0$ add a blocking thread

$\land\ requests' = [requests \text{ EXCEPT } ![r][i] = \text{"processed"}]$

Spec

$Next \triangleq$

```
  ∨ ∃ r ∈ _ReqIDs, i ∈ tryKeys :
     ∨ HitProxy(r, i)
     ∨ HitServer(r, i)

Fairness ≜ ∀ r ∈ _ReqIDs, i ∈ tryKeys :
                      ∧ SF_vars(HitServer(r, i))
                      ∧ SF_vars(HitProxy(r, i))

Spec ≜
  ∧ Init
  ∧ □[Next]_vars
  ∧ Fairness

RequestIsProcessedOnlyOnce ≜
    □(∀ req ∈ DOMAIN requests : Cardinality({x ∈ DOMAIN requests[req] : requests[req][x] = "processed"}) < 2)

THEOREM Spec ⇒ RequestIsProcessedOnlyOnce
```