MODULE *FridjappImpl*

EXTENDS *Integers*, *FiniteSets*, *Sequences*

CONSTANTS $INGREDIENT\_TYPES$, MAX_QTTY, $USERS$, $SERVER$

ASSUME $SERVER \in USERS$

VARIABLES *fridj*, *shoppingList*, *nRecipesMade*, *msgs*

$PT \triangleq$ INSTANCE PT

Sum up all integers in function FUN.

$Sum(fun) \triangleq PT!ReduceSet(\text{LAMBDA } k, acc : acc + fun[k],$
$\text{DOMAIN } fun, 0)$

Refinement mapping: the number of recipes split by users is summed up. The *fridj* of reference is the server's.

$fj \triangleq$ INSTANCE *fridjapp* WITH $nRecipesMade \leftarrow Sum(nRecipesMade),$
$fridj \leftarrow fridj[SERVER]$

What's a *Fridj* and a Shopping List.

$AllFridjes \triangleq [INGREDIENT\_TYPES \to Nat]$
$AllShoppingLists \triangleq AllFridjes$

$FinalUsers \triangleq USERS \setminus \{SERVER\}$

What's a synchronisation message, and it's actions.

$AddAction \triangleq$ "add"
$RmAction \triangleq$ "rm"
$Actions \triangleq \{AddAction, RmAction\}$
$FridjObject \triangleq$ "fridj"
$ShoppingListObject \triangleq$ "shoppingList"
$Objects \triangleq \{FridjObject, ShoppingListObject\}$
$Messages \triangleq [user : FinalUsers,$
$action : Actions,$
$object : Objects,$
$value : AllFridjes]$

$TypeOK \triangleq$

Every user's device is assigned a *fridj* function. One of these users is the server.

$\land fridj \in [USERS \to AllFridjes]$

Each user has a shopping list as well.

$\land shoppingList \in [USERS \to AllShoppingLists]$

We count the number of recipes made by each user.

$\land nRecipesMade \in [USERS \to Nat]$

The sequence of messages sent to *USERS* by *Message.user*

$\land\ msgs \in [USERS \to Seq(Messages)]$

$vars \triangleq \langle fridj,\ shoppingList,\ nRecipesMade \rangle$

$Min(a,\ b) \triangleq \text{IF } a < b \text{ THEN } a \text{ ELSE } b$

Definitions for creating messages.

$NewMessage(user,\ action,\ object,\ value) \triangleq$
$[user \mapsto user,$
$\ action \mapsto action,$
$\ object \mapsto object,$
$\ value \mapsto value]$

$AddToShoppingListMsg(user,\ value) \triangleq$
$NewMessage(user,\ AddAction,\ ShoppingListObject,\ value)$

$RmFromShoppingListMsg(user,\ value) \triangleq$
$NewMessage(user,\ RmAction,\ ShoppingListObject,\ value)$

$AddToFridjMsg(user,\ value) \triangleq$
$NewMessage(user,\ AddAction,\ FridjObject,\ value)$

$RmFromFridjMsg(user,\ value) \triangleq$
$NewMessage(user,\ RmAction,\ FridjObject,\ value)$

Send messages, that is the value of the 'msgs' variable in the second state of a step.

$Send(to,\ new\_msgs) \triangleq$
$msgs' = [msgs \text{ EXCEPT } ![to] = @ \circ new\_msgs]$

$NotifyServer(messages) \triangleq$
$Send(SERVER,\ messages)$

Actions taken by users. The first is when users add items in the shopping list.

$AddToShoppingList(user) \triangleq$
$\exists\, t \in INGREDIENT\_TYPES,\ n \in 1\,..\,MAX\_QTTY :$
$\text{LET } new\_shopping\_list \triangleq [shoppingList \text{ EXCEPT } ![t] = @ + n]$
$\text{IN} \quad \land\ shoppingList' = new\_shopping\_list$
$\land\ NotifyServer(\langle AddToShoppingListMsg(user,\ new\_shopping\_list) \rangle)$
$\land\ \text{UNCHANGED } \langle fridj,\ nRecipesMade \rangle$

Next, users add bought items in their *fridj* instance.

$BuyIngredients(user) \triangleq$
$\exists\, t \in INGREDIENT\_TYPES :$
$\text{LET } bought\_n \triangleq Min(MAX\_QTTY - fridj[t],\ shoppingList[t])$
$new\_shopping\_list \triangleq [shoppingList \text{ EXCEPT } ![t] = @ - bought\_n]$
$new\_fridj \triangleq [fridj \text{ EXCEPT } ![t] = @ + bought\_n]$
$\text{IN} \quad \land\ bought\_n > 0$

$\land$ $shoppingList' = new\_shopping\_list$
$\land$ $fridj' = new\_fridj$
$\land$ $NotifyServer(\langle RmFromShoppingListMsg(user,\ new\_shopping\_list),$
$\qquad AddToFridjMsg(user,\ new\_fridj)\rangle)$
$\land$ UNCHANGED $\langle nRecipesMade\rangle$

Finally, users cook! They remove items from the *fridj*.

$MakeRecipe(user) \triangleq$
$\quad \exists\, r \in fj\,!\,AllRecipes :$
$\qquad$ LET $new\_fridj \triangleq [t \in \text{DOMAIN } fridj \mapsto fridj[t] - r[t]]$
$\qquad \land \forall\, t \in \text{DOMAIN } r : fridj[t] \geq r[t]$
$\qquad \land fridj' = new\_fridj$
$\qquad \land nRecipesMade' = [nRecipesMade \text{ EXCEPT } ![user] = @ + 1]$
$\qquad \land NotifyServer(\langle RmFromFridjMsg(user,\ new\_fridj)\rangle)$
$\qquad \land$ UNCHANGED $shoppingList$

$Next \triangleq \exists\, u \in FinalUsers :$
$\quad \lor AddToShoppingList(u)$
$\quad \lor BuyIngredients(u)$
$\quad \lor MakeRecipe(u)$

$Init \triangleq$
$\quad \land fridj = [t \in INGREDIENT\_TYPES \mapsto 0]$
$\quad \land shoppingList = [t \in INGREDIENT\_TYPES \mapsto 0]$
$\quad \land nRecipesMade = [u \in USERS \mapsto 0]$
$\quad \land msgs = [u \in USERS \mapsto \langle\rangle]$

$Spec \triangleq Init \land \Box[Next]_{vars}$
$FairSpec \triangleq$
$\quad \land Spec$
$\quad \land \forall\, u \in USERS :\ \land \mathrm{WF}_{vars}(BuyIngredients(u))$
$\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(MakeRecipe(u))$
$\qquad\qquad\qquad\qquad\ \land \mathrm{WF}_{vars}(AddToShoppingList(u))$

THEOREM $Spec \Rightarrow fj\,!\,Spec$
THEOREM $FairSpec \Rightarrow fj\,!\,FairSpec$

$TempInv \triangleq \Diamond(\forall\, u \in USERS : nRecipesMade[u] > 0)$

* Modification History
* Last modified *Mon Jul* 29 17:09:38 *CEST* 2024 by *davd*33
* Created *Thu Jul* 25 23:17:45 *CEST* 2024 by *davd*33