MODULE *digital_watch*

Notes: Adaptation of the digital watch model presented by *Harel* (87)

To cite this model, please use:

```
@inproceedings{AbBa18modre,
 author = {Ali Abbassi and Amin Bandali and Nancy A.Day and Jose Serna},
 title = {A Comparison of the Declarative Modelling Languages{B}, {Dash}, and{TLA+}},
 booktitle = {International Workshop on Model - Driven Requirements
 Engineering(MoDRE)@IEEE International Requirements Engineering Conference(RE)},

 publisher = {To appear},
 year = 2018
}
```

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

VARIABLES
light, *status*,
waited_2_min, *waited_2_sec*,
pressed

$STATUS \triangleq \{$ "Time", "Date", "Wait", "Update"
, "Alarm1", "Alarm2", "Chime", "StopWatch"
, "Alarms_Beep" doesn't seem to be used right now $\}$

$KEYS \triangleq \{$"a", "b", "c", "d"$\}$

$vars \triangleq \langle light, status, waited\_2\_min, waited\_2\_sec, pressed \rangle$

$vars\_but\_light \triangleq \langle status, waited\_2\_min, waited\_2\_sec, pressed \rangle$

$vars\_but\_status \triangleq \langle light, waited\_2\_min, waited\_2\_sec, pressed \rangle$

Helper predicate for range of a function
$Range(f) \triangleq \{f[x] : x \in \text{DOMAIN } f\}$

$TypeOK \triangleq$ Typing invariant
$\land \quad light \in \text{BOOLEAN}$ FALSE: Off, TRUE: On
$\land \quad status \in STATUS$
$\land \quad waited\_2\_min \in \text{BOOLEAN} \land waited\_2\_sec \in \text{BOOLEAN}$
$\land \quad pressed \in [KEYS \to \text{BOOLEAN}]$

$Init \triangleq$ Initial state
$\land\, light = \text{FALSE}$ initially light is off
$\land\, status =$ "Time" initially display shows time
$\land\, waited\_2\_min = \text{FALSE} \land waited\_2\_sec = \text{FALSE}$
$\land\, pressed = [k \in KEYS \mapsto \text{FALSE}]$

$< \text{Light} >$

$light\_off\_light\_on \triangleq$

$\land\ pressed[\text{“b”}]$

$\land\ light' = \text{TRUE}$

$\land\ \text{UNCHANGED}\ vars\_but\_light$

$light\_on\_light\_off \triangleq$

$\land\ \neg pressed[\text{“b”}]$

$\land\ light' = \text{FALSE}$

$\land\ \text{UNCHANGED}\ vars\_but\_light$

$< /\text{Light} >$

$< Time >$

$time\_show\_date \triangleq$

$\land\ status = \text{“Time”}$

$\land\ pressed[\text{“d”}]$

$\land\ status' = \text{“Date”}$

$\land\ \text{UNCHANGED}\ vars\_but\_status$

$time\_try\_update \triangleq$

$\land\ status = \text{“Time”}$

$\land\ pressed[\text{“c”}]$

$\land\ status' = \text{“Wait”}$

$\land\ \text{UNCHANGED}\ vars\_but\_status$

$time\_go2alarm1 \triangleq$

$\land\ status = \text{“Time”}$

$\land\ pressed[\text{“a”}]$

$\land\ status' = \text{“Alarm1”}$

$\land\ \text{UNCHANGED}\ vars\_but\_status$

$< /Time >$

$< Date >$

$date\_show\_time \triangleq$

$\land\ status = \text{“Date”}$

$\land\ pressed[\text{“d”}]$

$\land\ status' = \text{“Time”}$

$\land\ \text{UNCHANGED}\ vars\_but\_status$

$date\_return\_to\_time \triangleq$

$\land\ status = \text{“Date”}$

$\land\ waited\_2\_min$

$\land\ status' = \text{“Time”}$

$\land\ \text{UNCHANGED}\ vars\_but\_status$

$< /Date >$

$< Wait >$

$wait\_show\_time \triangleq$

$\land\ status = \text{“Wait”}$

$\land\ \neg pressed[\text{“c”}]$

$\land\ status' = \text{“Time”}$

$\land$ UNCHANGED $vars\_but\_status$

$wait\_show\_update \triangleq$

$\land\ status = \text{“Wait”}$

$\land\ waited\_2\_sec$

$\land\ status' = \text{“Update”}$

$\land$ UNCHANGED $vars\_but\_status$

$< /Wait >$

$< Update >$

$update\_show\_time \triangleq$

$\land\ status = \text{“Update”}$

$\land\ pressed[\text{“b”}]$

$\land\ status' = \text{“Time”}$

$\land$ UNCHANGED $vars\_but\_status$

$< /Update >$

$< Alarm1 >$

$alarm1\_go2alarm2 \triangleq$

$\land\ status = \text{“Alarm1”}$

$\land\ pressed[\text{“a”}]$

$\land\ status' = \text{“Alarm2”}$

$\land$ UNCHANGED $vars\_but\_status$

$< /Alarm1 >$

$< Alarm2 >$

$alarm2\_go2chime \triangleq$

$\land\ status = \text{“Alarm2”}$

$\land\ pressed[\text{“a”}]$

$\land\ status' = \text{“Chime”}$

$\land$ UNCHANGED $vars\_but\_status$

$< /Alarm2 >$

$< Chime >$

$chime\_go2Stopwatch \triangleq$

$\land\ status = \text{“Chime”}$

$\land\ pressed[\text{“a”}]$

$\land\ status' = \text{“StopWatch”}$

$\land$ UNCHANGED $vars\_but\_status$

$< /Chime >$

$< StopWatch >$

$Stopwatch\_go2time \triangleq$
$\quad \land\ status = \text{“StopWatch”}$
$\quad \land\ pressed[\text{“a”}]$
$\quad \land\ status' = \text{“Time”}$
$\quad \land\ \text{UNCHANGED}\ vars\_but\_status$

$< /StopWatch >$

$< Alarms\_Beep >$
$< /Alarms\_Beep >$

$<$ Helpers $>$

Key presses

$PressKey(k) \triangleq$
$\quad \land\ \neg pressed[k] \land pressed' = [pressed\ \text{EXCEPT}\ ![k] = \text{TRUE}]$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_min,\ waited\_2\_sec \rangle$
$ReleaseKey(k) \triangleq$
$\quad \land\ pressed[k] \land pressed' = [pressed\ \text{EXCEPT}\ ![k] = \text{FALSE}]$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_min,\ waited\_2\_sec \rangle$

Waits

$waited\_2\_min\_t \triangleq \neg waited\_2\_min \land waited\_2\_min' = \text{TRUE}$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_sec,\ pressed \rangle$
$waited\_2\_min\_f \triangleq waited\_2\_min \land waited\_2\_min' = \text{FALSE}$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_sec,\ pressed \rangle$

$waited\_2\_sec\_t \triangleq \neg waited\_2\_sec \land waited\_2\_sec' = \text{TRUE}$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_min,\ pressed \rangle$
$waited\_2\_sec\_f \triangleq waited\_2\_sec \land waited\_2\_sec' = \text{FALSE}$
$\quad \land\ \text{UNCHANGED}\ \langle light,\ status,\ waited\_2\_min,\ pressed \rangle$

$<$ /Helpers $>$

$<$ Temporal properties $>$

I believe the original *eventually_time* property from Dash expressed in *CTL* says that “on a *press_a*, it’s possible that in the future the display will display the time”. However, since TLA+’s temporal logic is LTL-based and not *CTL*, we can’t easily express possibility properties. So, instead, we’ll state that “on a *press_a*, in the future the display will display the time”.

$EventuallyTime \triangleq$
$\quad \Box(pressed[\text{“a”}] \Rightarrow \Diamond(status = \text{“Time”}))$

Note: the above property does NOT hold with weak or strong fairness on all the actions on *vars*

< /Temporal properties >

Spec

$$
\begin{aligned}
Next \triangleq\ & \lor light\_off\_light\_on \lor light\_on\_light\_off \\
& \lor time\_show\_date \lor time\_try\_update \lor time\_go2alarm1 \\
& \lor date\_show\_time \lor date\_return\_to\_time \\
& \lor wait\_show\_time \lor wait\_show\_update \\
& \lor update\_show\_time \\
& \lor alarm1\_go2alarm2 \\
& \lor alarm2\_go2chime \\
& \lor chime\_go2Stopwatch \\
& \lor Stopwatch\_go2time \\
& \lor \exists\, k \in KEYS : PressKey(k) \\
& \lor \exists\, k \in KEYS : ReleaseKey(k) \\
& \lor waited\_2\_min\_t \lor waited\_2\_min\_f \\
& \lor waited\_2\_sec\_t \ \lor waited\_2\_sec\_f
\end{aligned}
$$

$Live \triangleq \mathrm{WF}_{vars}(Next)$

$Spec \triangleq Init \land \Box[Next]_{vars} \land Live$

every transition either satisfies the action formula *Next* or leaves the expression *vars* unchanged. In particular, this admits "stuttering transitions" that do not affect *vars* . That is to say, $\Box[Next]\_vars \triangleq \Box(Next \lor (vars' = vars))$

* Modification History
* Last modified *Tue Jul* 17 14:04:48 *EDT* 2018 by *amin*
* Created *Tue* May 29 18:29:07 *EDT* 2018 by *amin*