One-Way Bridge[1]

Manuel Carro
manuel.carro@upm.es

Universidad Politécnica de Madrid &
IMDEA Software Institute

Correctness by Construction
Manuel Carro
UPM / IMDEA

[1] Example and several slides from J. R. Abrial book *Modeling in Event-B: system and software engineering*.

Correctness by Construction
Manuel Carro
UPM / IMDEA

Goals of this chapter

- Example of reactive system development.
- Including modeling the environment.
- Invariants: capture requirements.
  - Invariant preservation will prove that requirements are respected.
- Increasingly accurate models (refinement).
- Refinements "zoom in", see more details.
- Models separately proved correct.
  - Final system: correct by construction.
- Correctness criteria: proof obligations.
- Proofs: helped by theorem provers working on sequent calculus.

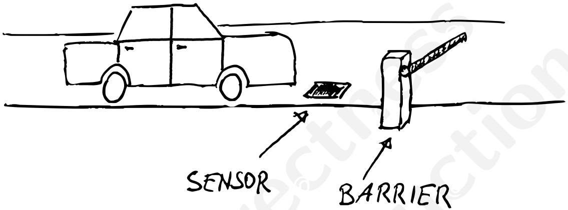
Correctness by Construction
Manuel Carro
UPM / IMDEA

Difference with previous examples

- Previous examples were *transformational*.
  - Input $\Rightarrow$ transformation $\Rightarrow$ output.
- Current example:
  - Interaction with environment.
- Sensors and communication channels:
  - Hardware sensors modeled with events.
  - Channels modeled with variables.

Correctness by Construction — Manuel Carro — UPM / IMDEA

Correctness within an environment

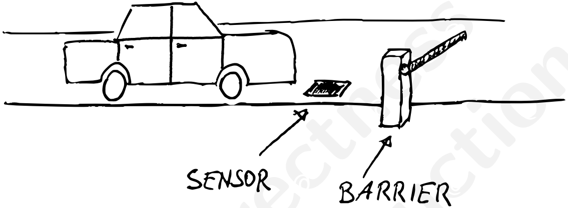


- Control software reads sensor, raises barrier.
  - If conditions allow it.

- Software behavior relies on environment:
  - Cars stop on a closed barrier.
  - Cars drive over sensor.
  - ...
- Correctness proofs: take this behavior into account.

Correctness by Construction © Manuel Carro UPM / IMDEA

Correctness within an environment



- Control software reads sensor, raises barrier.
  - If conditions allow it.

- Software behavior relies on environment:
  - Cars stop on a closed barrier.
  - Cars drive over sensor.
  - ...
- Correctness proofs: take this behavior into account.
  - Model external actions as events.
    - E.g., sensor signal raised by event.
    - Following expected behavior.
  - Software control also events.
  - Everything subject to proofs.

Correctness by Construction
Manuel Carro UPM / IMDEA

Requirements document

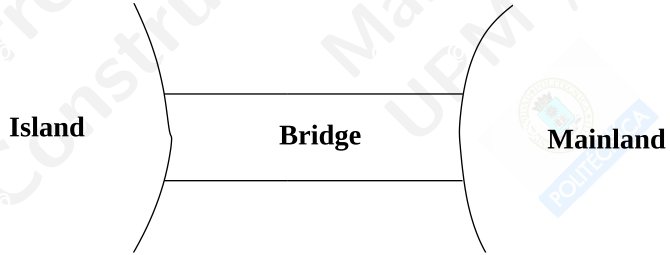
- Sequential systems specified through $\{Pre\}\ P\ \{Post\}$.
- Considerably more difficult in case of (a) large real-world and (b) reactive systems.
- Building it piece-wise, modeling (natural-language) requirements and ensuring they are respected: a way to ensure we have a detailed system specification that is provable correct.

- Two kinds of requirements:
  - Concerned with the equipment (EQP).
  - Concerned with system functionality (FUN).
- Objective: control cars on a narrow bridge.
- Bridge links the mainland to (small) island.

Requirements

The system is controlling cars on a bridge between the mainland and an island	FUN-1

- This can be illustrated as follows



Correctness by Construction Manuel Carro UPM / IMDEA

Requirements

- The controller is equipped with two traffic lights with two colors.

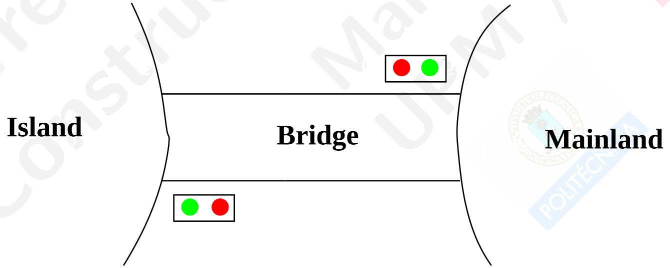
The system has two traffic lights with two colors: green and red	EQP-1

Correctness by Construction — Manuel Carro — UPM / IMDEA

Requirements

- One of the traffic lights is situated on the mainland and the other one on the island. Both are close to the bridge.

- This can be illustrated as follows



Requirements

The traffic lights control the entrance to the bridge at both ends of it	EQP-2

- Drivers are supposed to obey the traffic light by not passing when a traffic light is red.

Cars are not supposed to pass on a red traffic light, only on a green one	EQP-3

Correctness by Construction Manuel Carro UPM / IMDEA

Requirements

- There are also some car sensors situated at both ends of the bridge.

- These sensors are supposed to detect the presence of cars intending to enter or leave the bridge.

- There are four such sensors. Two of them are situated on the bridge and the other two are situated on the mainland and on the island.

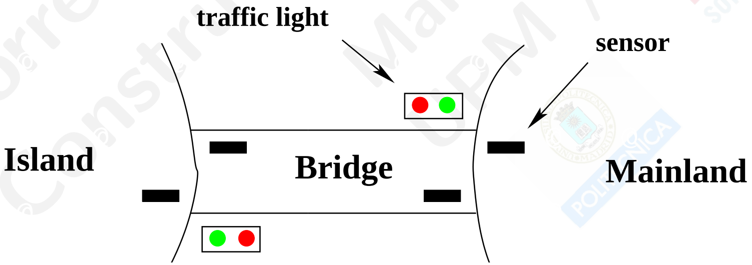
The system is equipped with four car sensors each with two states: on or off	EQP-4

Correctness by Construction Manuel Carro / IMDEA UPM

Requirements

The sensors are used to detect the presence of cars entering or leaving the bridge	EQP-5

- The pieces of equipment can be illustrated as follows:



Requirements

- This system has two main constraints: the number of cars on the bridge and the island is limited and the bridge is one way.

The number of cars on the bridge and the island is limited	FUN-2

The bridge is one way or the other, not both at the same time	FUN-3

Correctness by Construction Manuel Carro UPM / IMDEA

Overview

- Software controller has model of the world.
  - In some sense, it partially simulates it.
  - Knowledge of world through sensors.
  - Incrementally adding requirements, proving they are implemented.

- When finished, an additional software layer (= more events) simulate the “real world”.
  - “Real world” simulation only interacts with controller through sensors, actuators.
  - Proof that controller + simulation follow requirements.
- Real implementation: strip “Real world” layer, derive code from software controller.

Correctness by Construction — Manuel Carro — UPM / IMDEA

Strategy

Initial model $\boxed{\text{Limiting the number of cars (FUN-2).}}$

First refinement Introducing the one-way bridge (FUN-3).

Second refinement Introducing the traffic lights (EQP-1,2,3)

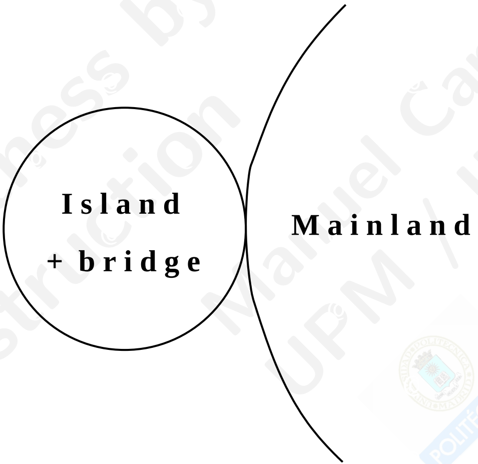
Third refinement Introducing the sensors (EQP-4,5)

Correctness by Construction
Manuel Carro
UPM / IMDEA

Initial model

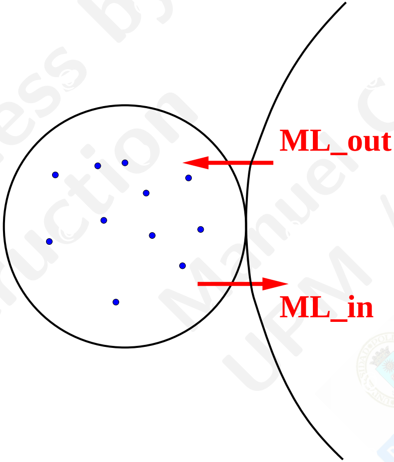
- We ignore the equipment (traffic lights and sensors).
- We do not consider the bridge.
- We focus on the pair island + bridge.
- FUN-2: limit number of cars on island + bridge.

Situation from the sky



Correctness by Construction
Manuel Carro UPM / IMDEA

Situation from the sky



Correctness by Construction
Manuel Carro
UPM / IMDEA

Formalization of state

✓ ***Create project $Cars$, context $c0$, machine $m0$, add constant, axiom, variable, invariants, initialization***

Static part (context):

constant: d

axm0_1: $d \in \mathbb{N}$

d is the maximum number of cars allowed in island + bridge.

Dynamic part (machine):

variable: n

inv0_1: $n \in \mathbb{N}$

inv0_2: $n \leq d$

n number of cars in island + bridge

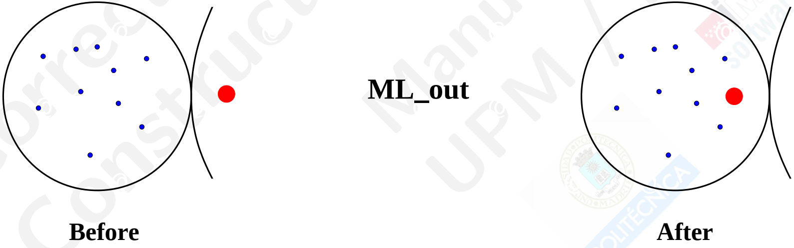
Always smaller than or equal to d (FUN_2)

- Labels axm0_1, inv0_1, chosen systematically.
- Label **axm**, **inv** recalls purpose.
- 0 (as in inv0_1): initial model.
- Later: inv1_1 for invariant 1 of refinement 1, etc.
- Any systematic convention is valid.

Situation from the sky

- This is the first transition (or event) that can be observed

- A car is leaving the mainland and entering the Island-Bridge



Correctness by Construction
Manuel Carro
UPM / IMDEA

Situation from the sky

- We can also observe a second transition (or event)

- A car leaving the Island-Bridge and re-entering the mainland

- The number of cars in the Island-Bridge is decremented

Correctness by Construction
Manuel Carro
UPM / IMDEA

Situation from the sky

✓ ***Create events ML_out, ML_in. Add actions. Guards?***

- Event ML_out increments the number of cars

$$\textbf{ML\_out} \quad n := n + 1$$

- Event ML_in decrements the number of cars

$$\textbf{ML\_in} \quad n := n - 1$$

- An event is denoted by its name and its action (an assignment)

Correctness by Construction Manuel Carro UPM / IMDEA

Events

INITIALISATION
$n := 0$

Event ML_out
  where
    $n < d$
  then
    $n := n + 1$
  end

Event ML_in
  where
    $0 < n$
  then
    $n := n - 1$
  end

ML_out/inv0_1/INV	$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d, n < d \vdash n + 1 \in \mathbb{N}$
ML_out/inv0_2/INV	$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d, n < d \vdash n + 1 \leq d$
ML_in/inv0_1/INV	$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d, 0 < n \vdash n - 1 \in \mathbb{N}$
ML_in/inv0_2/INV	$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d, n < d \vdash n - 1 < d$

Progress

- It is common to require that physical systems progress.
- We want cars to be able to either enter or exit.
- Therefore, (some) event(s) have to always be enabled.
- Depends on guards: *deadlock freedom*.

$$A_{1\ldots l}, I_{1\ldots m} \vdash \bigvee_{i=1}^{n} G_i(v, c)$$

- In our case:

$$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d \vdash n < d \vee 0 < n$$

- ✓ *Add invariant at the end, mark as theorem.*

Correctness by Construction
Manuel Carro
UPM / IMDEA

Progress

- It is common to require that physical systems progress.
- We want cars to be able to either enter or exit.
- Therefore, (some) event(s) have to always be enabled.
- Depends on guards: *deadlock freedom*.

$$A_{1\ldots l}, I_{1\ldots m} \vdash \bigvee_{i=1}^{n} G_i(v, c)$$

- In our case:

$$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d \vdash n < d \vee 0 < n$$

- ✓ *Add invariant at the end, mark as theorem.*
- Cannot be proven!

Correctness by Construction
Manuel Carro
UPM / IMDEA

Progress

- It is common to require that physical systems progress.
- We want cars to be able to either enter or exit.
- Therefore, (some) event(s) have to always be enabled.
- Depends on guards: *deadlock freedom*.

$$A_{1\ldots l}, I_{1\ldots m} \vdash \bigvee_{i=1}^{n} G_i(v, c)$$

- In our case:

$$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d \vdash n < d \lor 0 < n$$

- ✓ *Add invariant at the end, mark as theorem.*
- Cannot be proven!
- Why? Let us find out in which cases events may be in deadlock.
- Solve $\neg(n > 0 \lor n < d)$.

Progress

- It is common to require that physical systems progress.
- We want cars to be able to either enter or exit.
- Therefore, (some) event(s) have to always be enabled.
- Depends on guards: *deadlock freedom*.

$$A_{1\ldots l}, I_{1\ldots m} \vdash \bigvee_{i=1}^{n} G_i(v, c)$$

- In our case:

$$d \in \mathbb{N}, n \in \mathbb{N}, n \leq d \vdash n < d \vee 0 < n$$

- ✓ *Add invariant at the end, mark as theorem.*
- Cannot be proven!
- Why? Let us find out in which cases events may be in deadlock.
- Solve $\neg(n > 0 \vee n < d)$.
- If $d = 0$, no car can enter! Missing axiom: $0 < d$. Add it.
- Note that we are calculating the model.

Strategy

Initial model Limiting the number of cars (FUN-2).

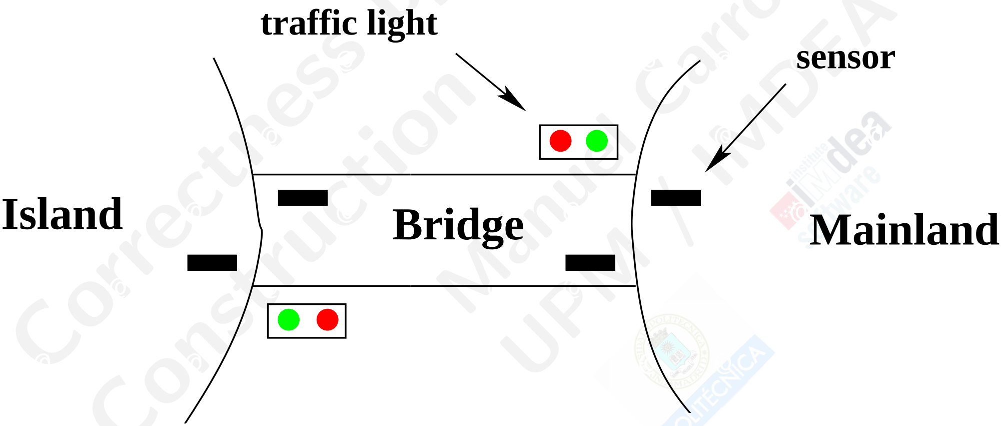
First refinement Introducing the one-way bridge (FUN-3).

Second refinement Introducing the traffic lights (EQP-1,2,3)

Third refinement Introducing the sensors (EQP-4,5)

Correctness by Construction
Manuel Carro
UPM / IMDEA

Physical system (reminder)

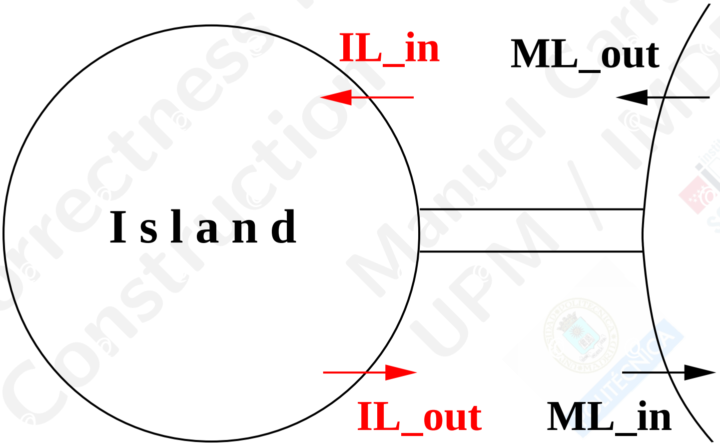


One-way bridge

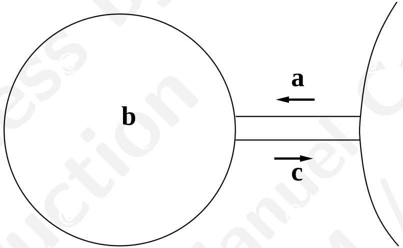
- We introduce the bridge.
- We refine the state and the events.
- We also add two new events: IL_in and IL_out.
- We are focusing on FUN-3: one-way bridge.

Correctness by Construction Manuel Carro UPM / IMDEA

One-way bridge



One-way bridge



- a denotes the number of cars on bridge going to island
- b denotes the number of cars on island
- c denotes the number of cars on bridge going to mainland
- a, b, and c are the concrete variables

Correctness by Construction Manuel Carro UPM / IMDEA

Refining state: invariants

Cars on bridge going to island	inv1_1	$a \in \mathbb{N}$
Cars on island	inv1_2	$b \in \mathbb{N}$
Cars on bridge to mainland	inv1_3	$c \in \mathbb{N}$
Linking new variables to previous model	inv1_4	??
Formalization of one-way bridge (FUN-3)	inv1_5	??

Correctness by Construction
Manuel Carro
UPM / IMDEA

Refining state: invariants

Cars on bridge going to island	inv1_1	$a \in \mathbb{N}$
Cars on island	inv1_2	$b \in \mathbb{N}$
Cars on bridge to mainland	inv1_3	$c \in \mathbb{N}$
Linking new variables to previous model	inv1_4	$a + b + c = n$
Formalization of one-way bridge (FUN-3)	inv1_5	??

inv1_4 glues the abstract state n with the concrete state a, b, c

Correctness by Construction Manuel Carro UPM / IMDEA

Refining state: invariants

Cars on bridge going to island	inv1_1	$a \in \mathbb{N}$
Cars on island	inv1_2	$b \in \mathbb{N}$
Cars on bridge to mainland	inv1_3	$c \in \mathbb{N}$
Linking new variables to previous model	inv1_4	$a + b + c = n$
Formalization of one-way bridge (FUN-3)	inv1_5	$a = 0 \vee c = 0$

Correctness by Construction
Manuel Carro
UPM / IMDEA

Refining state: invariants

Cars on bridge going to island	inv1_1	$a \in \mathbb{N}$
Cars on island	inv1_2	$b \in \mathbb{N}$
Cars on bridge to mainland	inv1_3	$c \in \mathbb{N}$
Linking new variables to previous model	inv1_4	$a + b + c = n$
Formalization of one-way bridge (FUN-3)	inv1_5	$a = 0 \vee c = 0$

A new class of invariant

Note that we are not finding an invariant to prove the correctness (= postcondition) of a loop. We are establishing an invariant to capture a requirement and we want the model to preserve the invariant, therefore implementing correctly that requirement.

Correctness by construction — Manuel Carro — UPM / IMDEA