Lecture 17: Introducing Formal Methods (with an example)

By Jean-Raymond Abrial
Definition of Formal Methods

- Not given yet
- Many very different definitions
- Give your own at the end of this lecture!
Why using formal methods (FM)?

- When there is **nothing better to do**.
- When the **risk is too high**.
- When people have already **suffered enough**.
- When people question their **development process**.
- Decision of using FM is **always strategic**.
Which formal method?

- This is a difficult question.
- Today many formal methods vendors.
- FM has become a meaningless buzz word.
- “Formal” alone does not mean anything.
Questions to be asked to FM vendors

- Is there a theory behind your FM?
- What kind of language is your FM using?
- Does there exist any RFINEMENT mechanism in your FM?
- Do you PROVE anything when using your FM?
- Have you got an efficient automatic prover?
Claimed difficulties in using FM

- You have to be a mathematician.
- Formalism is hard to master.
- Not visual enough (no boxes, arrows, etc.).
- People will not be able to do formal proofs.
Genuine difficulties (my own view)

- You have to **think a lot** before final coding.
- Incorporation in **development process**.
- **Model building** is an elaborate activity.
- **Prover technology** has to improve.
- Making proofs a **design criterium**.
- Poor quality of **requirement documents**.
Application areas

- Train systems
- Car systems
- Avionics and Space
- Power station control
- Telecom
- Defense
- Complex databases
- Large business network
- SmartCard applications
- Machine tools
- ...
QUESTION: What is common to
- an electronic circuit
- a file transfer protocol
- an airline booking system
- a PC operating system
- a nuclear plant control system
- a SmartCard electronic purse
- a launch vehicle flight controller

ANSWER: They are all complex.
Complex systems (2/2)

- They are made of many parts.
- They interact with a possibly hostile environment.
- They involve several executing agents.
- They require a high degree of correctness.
- Their construction spreads over several years.
- Their specifications are subjected to many changes.
- Their construction process requires a talented team.
Discrete systems

- These systems operate in an **discrete fashion**.
- Their dynamical behavior can be **abstracted by**:
  - A succession of **steady states**
  - Intermixed with **sudden jumps**.
- The possibility of state changes is **enormous**.
- The change frequency is **unthinkable**.
- Such systems are called **transition systems**.
Reasoning about (discrete) systems

- Two broad categories:
  - Test reasoning (98%)
  - Blue Print reasoning (10%)
Test reasoning

- Based on laboratory execution.
- Obvious incompleteness.
- The oracle is usually missing.
- Often implies postponing serious thinking.
- Re-adapting and re-shaping after testing.
- Reveals an immature technology.
“Blue Print” reasoning

- Based on a model: the “blue print”.
- Describing the system with the required precision.
- Completeness can be approached.
- Serious thinking made on the model, not on the final system.
- This is validated by proofs.
- Reveals a mature technology.
Incorporation within the development process

- Carefully **rewriting the requirement document**.

- Develop models by **successive refinement**.

- **Prove** each refinement step.

- **Use efficient tools** for:
  - Analyzing formal texts.
  - Generating proof statements.
  - Proving (as much as possible automatically).
Example: a mechanical press

- Presenting the *rewritten requirement document*.
- Partial development of models by *successive refinement*.
Mechanical press schema

MOTOR

ROD

SLIDE

TOOL

PART
Basic equipment

- A vertical slide with a tool at its lower extremity.
- An electrical rotating motor.
- A connecting rod transforming rotary movement to vertical movement of slide.
- A clutch engaging or disengaging the motor on the rod.
- When the clutch is disengaged, the slide stops “immediately”.
Initial situation
Starting the motor
The motor works
The motor works
The motor works
Adding a tool
The motor works
The motor works
Putting a part
The motor works
The motor works
The motor works
The motor works
The motor works
The motor works
The motor works
The motor works
The motor works
Engaging the clutch
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
Disengaging the clutch
The motor works
The motor works
The motor works
Removing the part
The motor works
The motor works
Adding a new part
Engaging the clutch
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
The press works
Disengaging the clutch
The motor works
The motor works
The motor works
Removing the part
The motor works
The motor works
Removing the tool
Stopping the motor
Final situation
Basic commands

- Command 1: Start motor.
- Command 2: Engage clutch.
- Command 3: Disengage clutch.
- Command 4: Stop motor.
Basic user actions

- Action 1: **Change the tool** at the lower extremity of the slide.
- Action 2: **Put a part** to be treated under the slide.
- Action 3: **Remove the part**.
First schematic view

COMMANDS → EQUIPMENT
A typical user session

1: Start the motor (command 1),
2: Change the tool (action 1),
3: Put a part (action 2),
4: Engage the clutch (command 2): the press now works,
5: Disengage the clutch (command 3): the press does not work,
6: Remove the part (action 3),
7: Repeat zero or more times actions 3 to 6,
8: Repeat zero or more times actions 2 to 7,
9: Stop the motor (command 4).
Danger: Necessity of a controller

- Action 2 (change the tool),
- Action 3 (put a part),
- Action 6 (remove the part) are all DANGEROUS.
Second schematic view
More elaborate commands for protecting the user.

- Controlling the way the clutch is engaged or disengaged.
- Protection by means of the bi-manual device.
- Protection by means of a front door.
- The pedal.
The bi-manual device
The bi-manual device: assumptions

- A single user.

- A single user only has two hands.

- The user has both hands either
  - on the bi-manual device or (exclusively)
  - within the press.

- Distance between the device and the press is long enough (more below).
The bi-manual device: behavior

- When both hands are put *simultaneously* on the device
  - The clutch is *engaged*.

- As soon as the user removes *at least one hand* from the device
  - The clutch is *disengaged*.

- Before putting one’s hands on the device
  - Hands must be *both removed* from the device.

- *Simultaneously* means that the *delay* between both hands is *bounded*: delay D5 (more on delays below).
The bi-manual device: consequence

- Maintaining the **clutch engaged**
- and having at the same time **one’s hands in the press**
- is impossible.
The front door
The front door: Assumptions

- User can have hands within the press only when door is open.

- Distance between door and inside of the press is long enough (more below).
The front door: behavior

- When front *door is closed*, the user can engage the clutch (with the bi-manual device).

- He can then *freely remove* both hands from the device (clutch is *not disengaged*).

- As soon as he *opens* the front door, the *clutch is disengaged*.

- As soon as he *closes* the front door, the *clutch is engaged again*.

- Pressing a *special button* B6 stops this procedure.
The front door: consequence

- Having the **clutch engaged**
- and at the same time **one’s hands in the press**
- is **impossible**.
The distance problem

- **Distance** between the device and the press is long enough.

- **Distance** between door and inside of the press is long enough.

- These distances must be carefully calculated
  - so that the press is **effectively stopped**
  - before the user can put hands within the press.

- Consequence: carefully checking the **stopping time** of the press after disengaging the clutch (more below).
The pedal: assumptions and behavior

- The user is moving the motor manually (no danger thus).

- The clutch is engaged by pressing the pedal (with the foot).
Buttons and commands so far at the disposal of the user

- **B1**: void
- **B2**: void
- **B3**: void
- **B4**: start motor
- **B5**: stop motor
- **B6**: continuous cycle stop (when using front door)
- **B7**: void
- **BM**: bi-manual device
- **FD**: front door
- **PL**: pedal
The concept of modes of operation

- Using the bi-manual device.
- Using the front door.
- Using the pedal.
- Also normal and maintenance modes.
Summary of modes (more below)

- **M1**: Maintenance mode without motor and pedal
- **M2**: Maintenance mode with motor and bi-manual device
- **M3**: Normal mode with motor and bi-manual device
- **M4**: Normal mode with motor and front door
- **M5**: Stop mode
Changing modes (1/2)

- A rotation button B1 is used for changing mode.

- When using B1, the clutch must be automatically disengaged.

- Five wires (on/off) are installed between B1 and the controller.

- Only one wire should be “on” at a time: emergency otherwise (more on emergency below).
Changing modes (2/2)

- A small delay D1 should be awaited after turning button (for electrical stabilization).

- To enter the new mode, user must push an “arming” button B2.

- B2 tests for some special conditions depending on the mode (more below).
Buttons and commands so far at the disposal of the user

- **B1**: mode selection (5 positions)
- **B2**: arming
- **B3**: void
- **B4**: start motor
- **B5**: stop motor
- **B6**: continuous cycle stop
- **B7**: void
- **BM**: bi-manual device
- **FD**: front door
- **PL**: pedal
Summary of delays so far

- **D1**: when changing mode
- **D2**: void
- **D3**: void
- **D4**: void
- **D5**: when using the bi-manual device
Upper and lower positions of the vertical slide

- In M2, clutch automatically disengaged at upper point.
- In M3, clutch automatically disengaged at upper point.
- In M3, clutch disengaged when removing hands while going down.
- In M4, clutch disengaged at upper point after pressing button B6.
- Upper and lower positions determined by cams (next slide).
Upper and lower cams

upper cam “on”

340° 350° 15°

lower cam “on”

170°
More on motor and clutch

- Controller sends commands (start/stop) to motor and clutch.

- After a change is received, they must send an acknowledgment.

- Acknowledgment must be received before certain delays D2 and D3 (emergency otherwise).
Summary of delays so far

- **D1**: when changing mode
- **D2**: when starting or stopping the motor
- **D3**: when engaging or disengaging the clutch
- **D4**: void
- **D5**: when using the bi-manual device
Braking

- In mode M3 or M4, clutch automatically disengaged at upper point.
- If acknowledgment from clutch received after 15 degrees (upper cam)
  - an emergency is raised.
Emergency stop

- Can be raised manually (button B7).

- Can also be raised by specific conditions depending on the mode.

- Lit an emergency lamp.

- Emergency state: no normal command can be used.

- Press arming button B2 to resume normal mode (turn off lamp).
Buttons and commands so far at the disposal of the user

- **B1**: mode selection (5 positions)
- **B2**: arming
- **B3**: void
- **B4**: start motor
- **B5**: stop motor
- **B6**: continuous cycle stop
- **B7**: emergency
- **BM**: bi-manual device
- **FD**: front door
- **PL**: pedal
- **SD**: side door
Environment actuators

- **MR**: motor
- **CL**: clutch
- **LP**: lamp
Wires (1/2)

- **Bi-manual**: 2 input wires per hand (when different: emergency)

- **Front door**: 2 input wires (when different: emergency)

- **Pedal**: 2 input wires (when different: emergency)

- **Clutch**: 2 output wires, 2 input wires (when different: emergency)

- **Motor**: 1 output wire, 1 input wire

- **Lamp**: 1 output wire
Wires (2/2)

- **Upper cam**: 1 input wire
- **Lower cam**: 2 input wires (when different: emergency)
- **Button B1**: 5 input wires (when inconsistent: emergency)
- **Other buttons**: 1 input wire per button
- **Side door for maintenance**: 1 input wire
Controller input wires (26 wires)
Controller output wires (4 wires)
Summary of emergency stop

- Emergency button,
- Brake,
- Cam (bad redundancy),
- Front door (bad redundancy),
- Motor (elapsed delay),
- Clutch (bad redundancy and elapsed delay),
- Modes (inconsistency),
- Foot (bad redundancy),
- Left hand (bad redundancy),
- Right hand (bad redundancy)
Mode analysis: M1

- Initial condition: Motor should not work (done by controller; delay D4)
- Emergencies: motor, clutch, pedal

- Mode selection button: Yes
- Arming button: Yes
- Motor starting button: No
- Motor stopping button: No
- Stopping continuous cycle button: No
- Emergency button: Yes
- Bi-manual device: No
- Pedal: Yes
M1: Clutch disengagement

- When removing foot from pedal.
Mode analysis: M2

- Initial condition: Motor should work (press B4)
- Emergencies: motor, clutch, bi-manual device

- Mode selection button: Yes
- Arming button: Yes
- Motor starting button: Yes
- Motor stopping button: Yes
- Stopping continuous cycle button: No
- Emergency button: Yes
- Bi-manual device: Yes
- Pedal: No
M2: Clutch disengagement

- When removing hands from bi-manual device.
- At upper point.
Mode analysis: M3

- Initial condition: Motor should work (press B4), side door closed
- Emergencies: motor, clutch, bi-manual device, brake, cam

- Mode selection button: Yes
- Arming button: Yes
- Motor starting button: Yes
- Motor stopping button: Yes
- Stopping continuous cycle button: No
- Emergency button: Yes
- Bi-manual device: Yes
- Pedal: No
M3: Clutch disengagement

- When removing hands from bi-manual device
  - if press is going down
  - and after it has stopped at upper point.

- When opening side door.

- At upper point.
Mode analysis: M4

- Initial condition: Motor should work (press B4), side door closed, front door closed
- Emergencies: motor, clutch, bi-manual device, brake, front door, cam

- Mode selection button: Yes
- Arming button: Yes
- Motor starting button: Yes
- Motor stopping button: Yes
- Stopping continuous cycle button: Yes
- Emergency button: Yes
- Bi-manual device: Yes
- Pedal: No
M4: Clutch disengagement

- When opening front door.
- When opening side door.
- At upper point after pressing button B6.
Mode analysis: M5

- Initial condition: Motor should not work (done by controller)
- Emergencies: motor

- Mode selection button: Yes
- Arming button: No
- Motor starting button: No
- Motor stopping button: No
- Stopping continuous cycle button: No
- Emergency button: No
- Bi-manual device: No
- Pedal: No
Summary of delays

- **D1**: when changing mode
- **D2**: when starting or stopping the motor
- **D3**: when engaging or disengaging the clutch
- **D4**: before entering mode M1
- **D5**: when using the bi-manual device
Characterizing the model

- It is a closed model of:
  - the environment (equipment and commands),
  - the controller.

- This model is developed by means of successive refinements.

- When it will be complete, it could be used to:
  - perform a simulation (environment and controller).
  - program a micro-computer (controller).
The first three models

- These first models are devoted to the environment only.

- They refine each other.

- 1\textsuperscript{st} model: Introducing the free movements of the press.

- 2\textsuperscript{nd} model: Introducing the behavior and safety laws.

- 3\textsuperscript{rd} model: Introducing the motor and the clutch.
The five next models: treating equipment

- 4th model: Simplified **clutch** commands.
- 5th model: Simplified model of **movements**.
- 6th model: The **front door**.
- 7th model: The **side door**.
- 8th model: Starting and stopping **motor**.
The next two models: refining treatments

- 9th model: Refining movement (the cams).
- 10th model: Refining the clutch command (bi-manual device).
The last models

- 11th model: Changing modes and emergencies.
- 12th model: Delays and wire redundancies.
- 13th model: Refining the clocks.
- 14th model: Refining the mode changing.
Model structure: discrete systems

- A model is made of
  - a number of variables
  - a number of transitions on these variables (called events).

- Variables are typed.

- An event is made of
  - a guard (necessary enabling conditions)
  - an action (variable modifications).

- A model has no control mechanism besides the events.
Structure of final model

- Environment and controller events.
- Environment and controller variables.
- Sensor and actuator variables (correspond to the wires).
The environment events.

The environment variables modified by environment events.

The sensor variables modified by environment events.

The actuator variables read by environment events.

The controller variables not seen by environment events.
Decomposing the final model: controller

- The controller events.
- The controller variables modified by controller events.
- The sensor variables read by controller events.
- The actuator variables modified by controller events.
- The environment variables not seen by controller events.
The controller does not exist: thus no sensors, no actuators.

The equipment just “knows” the various modes.

These models describe what an external observer can “see”.

They also describe the invariant laws of the various modes.

These models are gradually refined.
Such variables are defined *without constraints* to begin with:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRESS</td>
<td>{stopped, working}</td>
</tr>
<tr>
<td>HANDS</td>
<td>{free, busy}</td>
</tr>
<tr>
<td>FRONT_DOOR</td>
<td>{open, closed}</td>
</tr>
<tr>
<td>SIDE_DOOR</td>
<td>{open, closed}</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>{up, down}</td>
</tr>
<tr>
<td>STOP_UPPER_POINT</td>
<td>{yes, no}</td>
</tr>
</tbody>
</table>
The press is **stopped**: one observes that it *can be started*.

\[
\text{start\_press} \equiv \\
\text{when} \\
\quad \text{PRESS} = \text{stopped} \\
\text{then} \\
\quad \text{PRESS} := \text{working} \\
\text{end}
\]
The press works: one observes that it can be stopped.

\[
\text{stop\_press} \rightleftharpoons \\
\text{when} \\
\text{PRESS} = \text{working} \\
\text{then} \\
\text{PRESS} := \text{stopped} \\
\text{end}
\]
1st model: freeing the hands (case 1)

- Press is **working** and hands are **busy**: one can observe that **hands are freed and press still works**.

\[
\text{free\_hands} \iff \\
\text{when} \\
\text{PRESS} = \text{working} \land \\
\text{HANDS} = \text{busy} \\
\text{then} \\
\text{HANDS} := \text{free} \\
\text{end}
\]
Press is **working** and hands are **busy**: one can observe that **hands are freed** and **press is stopped**.

\[
\text{stop}_\text{press}_\text{free}_\text{hands} \equiv \\
\text{when} \\
PRESS = \text{working} \land \\
HAN\text{DS} = \text{busy} \\
\text{then} \\
PRESS, HAN\text{DS} := \text{stopped, free} \\
\text{end}
\]
1st model: other events

- busy_hands
- press_up
- close_front_door
- open_front_door
- close_side_door
- open_side_door
- stop_press_down
- press_down
- stop_press_open_front_door
- stop_press_open_side_door
More on model structure: invariant and refinement

This slide is the most important one.

- The variables of a model can be constrained by some invariant laws

- Proving that the invariant laws are maintained by the events.

- A model can be refined by
  - Transforming the existing events
  - Adding new events

- Proving that the refinement is correct.
2\textsuperscript{nd} model: modes and rules

\[ mode \in \{M1, M2, M3, M4, M5\} \]

- The rules define the constraints to be followed when the press works.

- In mode M2, \textit{hands must be busy}:

\[ \begin{align*}
    mode &= M2 \land \\
    PRESS &= \text{working} \\
    \Rightarrow \hspace{1cm} HANDS &= \text{busy}
\end{align*} \]
2nd model: rules (cont’d)

- In mode M3, hands must be busy
  - when the press goes down
  - and after the stop at the upper point.
- In mode M3, the side door must be closed.

\[
\begin{align*}
\text{mode} &= M3 \land \\
\text{PRESS} &= \text{working} \land \\
\text{DIRECTION} &= \text{down} \land \\
\text{STOP_UPPER_POINT} &= \text{yes} \\
\Rightarrow \quad \text{HANDS} &= \text{busy}
\end{align*}
\]

\[
\begin{align*}
\text{mode} &= M3 \land \\
\text{PRESS} &= \text{working} \\
\Rightarrow \quad \text{SIDE_DOOR} &= \text{closed}
\end{align*}
\]
In mode M4, the **front door** must be closed.

\[
\text{mode} = M4 \land \\
\text{PRESS} = \text{working} \\
\implies \\
\text{FRONT\_DOOR} = \text{closed}
\]

In mode M4, the **side door** must be closed.

\[
\text{mode} = M4 \land \\
\text{PRESS} = \text{working} \\
\implies \\
\text{SIDE\_DOOR} = \text{closed}
\]
In mode M5, the press is always stopped.

\[
\text{mode} = M5 \implies \text{PRESS} = \text{stopped}
\]

When the press goes up, the stop at upper point is not done.

\[
\text{DIRECTION} = \text{up} \implies \text{STOP_UPPER_POINT} = \text{no}
\]
Observe the guard strengthening.

start_press ≜

when

PRESS = stopped ∧
mode = M2 ⇒ HANDS = busy ∧
mode = M3 ∧ DIRECTION = down ∧

STOP_UPPER_POINT = yes ⇒ HANDS = busy ∧
mode = M3 ⇒ SIDE DOOR = closed ∧
mode = M4 ⇒ SIDE DOOR = closed ∧
mode = M4 ⇒ FRONT DOOR = closed ∧
mode ≠ M5

then

PRESS := working

end
Hands can be freed without stop
- in all modes except M2
- in mode M3 only if the press goes up or
  if stop at upper point has not happened yet.

\[
\text{free\_hands} \ni \text{when} \begin{align*}
\text{PRESS} &= \text{working} \land \\
\text{HANDS} &= \text{busy} \land \\
\text{mode} &\neq M2 \land \\
\text{mode} &= M3 \Rightarrow \text{DIRECTION} = \text{up} \lor \\
\text{STOP\_UPPER\_POINT} &= \text{no}
\end{align*} \text{then} \\
\text{HANDS} &:= \text{free} \end{align*}
\]
Hands have to be freed with a stop
- in modes M2 or M3
- in mode M3 if the press goes down and if stop at upper point already occurs.

\[
\text{stop\_press\_free\_hands} \triangleq \\
\text{when} \quad \\
\quad \text{PRESS} = \text{working} \land \\
\quad \text{HANDS} = \text{busy} \land \\
\quad \text{mode} \in \{M2, M3\} \land \\
\quad \text{mode} = M3 \implies \text{DIRECTION} = \text{down} \land \\
\quad \text{mode} = M3 \implies \text{STOP}\_\text{UPPER}\_\text{POINT} = \text{yes} \\
\text{then} \quad \\
\quad \text{PRESS}, \text{HANDS} := \text{stopped}, \text{free} \\
\text{end}
\]
Abstract variable \textit{PRESS} will disappear.

One is going to link \textit{PRESS} with \textit{MOTOR} and \textit{CLUTCH}.
3rd model: the linking invariant

- In modes M1 and M5, the motor is stopped.
- In mode M5, the clutch is disengaged.
- When the clutch is disengaged, the press is stopped.

\[
\begin{align*}
\text{mode} = M1 & \implies \text{MOTOR} = \text{stopped} \\
\text{mode} = M5 & \implies \text{MOTOR} = \text{stopped} \\
\text{mode} = M5 & \implies \text{CLUTCH} = \text{disengaged} \\
\text{CLUTCH} = \text{disengaged} & \implies \text{PRESS} = \text{stopped}
\end{align*}
\]
- In mode M1, the press works if the clutch is engaged.

- In other modes (except M5), the press works if motor works and clutch is engaged.

\[
\begin{align*}
\text{mode} &= M1 \land \text{CLUTCH} = \text{engaged} \\
\Rightarrow \\
\text{PRESS} &= \text{working}
\end{align*}
\]

\[
\begin{align*}
\text{mode} &\neq M1 \land \text{MOTOR} = \text{working} \land \text{CLUTCH} = \text{engaged} \\
\Rightarrow \\
\text{PRESS} &= \text{working}
\end{align*}
\]
start_press \equiv

\textbf{when}

\begin{align*}
&CLUTCH = \text{disengaged} \land \\
&mode \neq M1 \Rightarrow MOTOR = \text{working} \land \\
&mode = M2 \Rightarrow HANDS = \text{busy} \land \\
&mode = M3 \land DIRECTION = \text{down} \land \\
&\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{STOP\_UPPER\_POINT} = \text{yes} \Rightarrow HANDS = \text{busy} \land \\
&mode = M3 \Rightarrow SIDE\_DOOR = \text{closed} \land \\
&mode = M4 \Rightarrow SIDE\_DOOR = \text{closed} \land \\
&mode = M4 \Rightarrow FRONT\_DOOR = \text{closed} \land \\
&mode \neq M5
\end{align*}

\textbf{then}

\begin{align*}
&CLUTCH := \text{engaged}
\end{align*}

\textbf{end}
3rd model: starting motor

- Before starting motor **clutch must be disengaged.**

```
start_motor ≜
  when
    MOTOR = stopped ∧
    CLUTCH = disengaged ∧
    mode ≠ M1 ∧
    mode ≠ M5
  then
    MOTOR := working
  end
```
The five next models: treating equipment

- 4th model: Simplified **clutch** commands.
- 5th model: Simplified model of **movement**.
- 6th model: The **front door**.
- 7th model: The **side door**.
- 8th model: Starting and stopping **motor**.
The next two models: refining treatments

- 9\textsuperscript{th} model: Refining \textit{movement} (the cams).
- 10\textsuperscript{th} model: Refining the \textit{clutch command}. 
The last models

- 11th model: Changing modes and emergencies.
- 12th model: Delays and wire redundancies.
- 13th model: Refining the clocks.
- 14th model: Refining the mode changing.
Summary: 20 sensors

- Clutch sensor (3rd refinement),
- 2nd clutch sensor (11th refinement),
- Motor sensor (7th refinement),
- Left hand sensor (9th refinement),
- 2nd left hand sensor (11th refinement),
- Right hand sensor (9th refinement),
- 2nd right hand sensor (11th refinement),
- Foot sensor (3rd refinement),
- 2nd foot sensor (11th refinement),
Summary: 20 sensors (cont’d)

- Front door sensor (5th refinement),
- 2nd front door sensor (11th refinement),
- Side door sensor (6th refinement),
- Upper cam sensor (8th refinement),
- Lower cam sensor (8th refinement),
- 2nd lower cam sensor (11th refinement),
- M1 sensor (13th refinement),
- M2 sensor (13th refinement),
- M3 sensor (13th refinement),
- M4 sensor (13th refinement),
- M5 sensor (13th refinement).
Summary: 5 clocks

- Bi-manual clock (9th refinement),
- Motor clock (11th refinement),
- Clutch clock (11th refinement),
- Mode clock (11th refinement),
- M1 clock (11th refinement).
Summary: 9 emergency stops

- Button (10\textsuperscript{th} refinement),
- Brake (11\textsuperscript{th} refinement),
- Cam (11\textsuperscript{th} refinement),
- Front door (11\textsuperscript{th} refinement),
- Motor (11\textsuperscript{th} refinement),
- Clutch (11\textsuperscript{th} refinement),
- Modes (13\textsuperscript{th} refinement),
- Foot (11\textsuperscript{th} refinement),
- Left hand (11\textsuperscript{th} refinement),
- Right hand (11\textsuperscript{th} refinement).
Summary: variables of the last refinement

- 9 environment variables,
- 26 sensor variables,
- 4 actuator variables,
- 12 clock variables,
- 32 controller variables.
Summary: events of the last refinement

- 68 environment events,
- 89 controller events,
- 329 lines for constants, variables and initialization,
- 745 lines for environment events,
- 1536 lines for controller events.
- 5500 lines of assembly code for the controller.
Summary: proofs (total, interactive)

- 1st refinement: 56, 6
- 2nd refinement: 15, 2
- 3rd refinement: 174, 4
- 4th refinement: 32, 0
- 5th refinement: 12, 0
- 6th refinement: 12, 0
- 7th refinement: 47, 2
- 8th refinement: 31, 7
- 9th refinement: 49, 0
- 10th refinement: 56, 1
- 11th refinement: 255, 19
- 12th refinement: 154, 19
- 13th refinement: 32, 0
- TOTAL: 925, 60
End of lecture 17