Introducing Formal Methods (with an example)

J-R. Abrial

September 2004
Formal Methods: a Great Confusion

- **What** are they used for?

- **When** are they to be used?

- Is **UML** a formal method?

- Are they needed when doing **OO programming**?

- What is their **definition**?
Why Using Formal Methods?

- 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**.
- 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?

- Do there exist any refinement mechanism?

- 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.
- 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**.
About Formal Proofs in Industry (Some Figures)

- Rules of Thumb:

\[ n \text{ lines of final code implies } \frac{n}{3} \text{ proofs} \]

- 90% of proofs discharged automatically
- 10% of proofs discharged interactively
- 400 interactive proofs per man-month

- 60,000 lines of final code \( \sim \) 20,000 proofs \( \sim \) 2,000 int. proofs
- 2,000 interactive proofs \( \sim \) \( \frac{2000}{400} = 5 \) man-months
- Less expensive than heavy testing
What About Other Engineering Disciplines

- Some mature engineering disciplines:
  - Avionics,
  - Space,
  - Civil engineering,
  - Mechanical engineering,
  - Train systems,
  - Ship building.

- Are there any equivalent approaches to Formal Methods?

- Yes, BLUE PRINTS
What is a Blue Print?

- A certain representation of the future system

- It is not a mock-up (although mock-ups can be very useful too)

- The basis is lacking (you cannot “drive” the blue print of a car)

- Allows to reason about the future system during its design

- Is it important? (according to professionals) YES
Reasoning about the Future System?

- Defining and calculating its behavior (what it does)

- Incorporating constraints (what it must not do)

- Defining architecture

- Based on some underlying theories
  - strength of materials,
  - fluid mechanics,
  - gravitation,
  - etc.
Techniques of “Blue Printing”

- Using pre-defined conventions (often computerized these days)

- Conventions should help facilitate reasoning

- Adding details on more accurate versions

- Postponing choices by having some open options

- Decomposing one blue print into several

- Reusing “old” blue prints (with slight changes)
What About BEFORE the Blue Print

- Define main objectives of future system
- Define requirements
- Study feasibility
What About AFTER the Blue Print

- Construct the system

- Perform functional tests

- Study how constraints are obeyed

- Organize maintenance
Reasonings 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
Definitions of Formal Methods (subjective)

- Formal methods are techniques for building and studying blue prints

ADAPTED TO OUR DISCIPLINE

Our discipline is: design of hardware and software SYSTEMS

- Such blue prints are now called models

- Reminder:
  - Models allow to reason about a FUTURE system
  - The basis is lacking (hence you cannot “execute” a model)
Example: a Mechanical Press

- Presenting the *rewritten* requirement document

- Partial Development of models by *successive refinement*
Mechanical Press Schema

- MOTOR
- ROD
- SLIDE
- TOOL
- PART
- start
- stop
Basic Equipment

- A Vertical Slide with a tool at its lower extremity

- An electrical Rotating Motor

- A Connecting Rod transforming rot. mvt. to vert. mvt. 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
Adding a Tool
Putting a Part
Engaging the Clutch
The Press Works
The Press Works
The Press Works
Disengaging the Clutch
Removing the Part
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
The Press works
Disengaging the Clutch
Removing the Part
Removing the Tool
Stopping 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
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

CONTROLLER

COMMANDS

CONTROLLER

EQUIPMENT
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: 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 ones hands on the device
  - they must be *both removed* from it

- *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 **ones 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 removes both hands from the device (clutch is not disengaged)

- As soon has he opens the front door, the clutch is disengaged

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

- Pressing a special button B6 stops this procedure
The Front Door: Consequence

- Having the **clutch engaged**

- and at the same time **ones 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
- A rotating 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)
- 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"

lower cam "on"

340° 350° 15° 170°
- Controller sends commands (start/stop) to Motor and Clutch

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

- Ack. 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
- In mode M3 or M4, clutch automatically disengaged at upper point

- If Ack. from clutch received after 15 degrees (upper cam)

- an emergency is raised
- 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
- **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
- **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)

- MR
- LP
- CL
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

- Init. Cond.: 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

- Init. Cond.: **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

- Init. Cond.: 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

- Init. Cond.: Motor should work (press B4), side door closed, front door closed
- Emerg.: 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

- Init. Cond.: **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
- 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

- 1st model: Introducing the free movements of the press

- 2nd model: Introducing the behavior and safety laws

- 3rd 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)
Decomposing the **Final Model: Environment**

- 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
1st Model: the Environment Variables

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>
1st Model: Starting the Press

The press is stopped: one observes that it can be started

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

\[
\text{stop\_press} \equiv \\
\text{when} \\
PRESS = \text{working} \\
\text{then} \\
PRESS := \text{stopped} \\
\text{end}
\]
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

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

\[
\begin{align*}
\text{stop} \text{ press} \_ \text{free} \_ \text{hands} & \equiv \\
\text{when} & \\
\text{PRESS} & = \text{working} \\
\text{HANDS} & = \text{busy} \\
\text{then} & \\
\text{PRESS}, \text{HANDS} & : = \text{stopped}, \text{free} \\
\text{end} & 
\end{align*}
\]
<table>
<thead>
<tr>
<th>busy_hands</th>
<th>open_side_door</th>
</tr>
</thead>
<tbody>
<tr>
<td>press_up</td>
<td>stop_press_down</td>
</tr>
<tr>
<td>close_front_door</td>
<td>press_down</td>
</tr>
<tr>
<td>open_front_door</td>
<td>stop_press_open_front_door</td>
</tr>
<tr>
<td>close_side_door</td>
<td>stop_press_open_side_door</td>
</tr>
</tbody>
</table>
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
2nd Model: Modes and Rules

\[
\text{mode} \in \{M_1, M_2, M_3, M_4, M_5\}
\]

- The rules define the constraints to be followed \textit{when the press works}.

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

\[
\begin{align*}
\text{mode} &= M_2 \\
\text{PRESS} &= \text{working} \\
\implies \\
\text{HANDS} &= \text{busy}
\end{align*}
\]
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

\[
\begin{align*}
\text{mode} & = M3 \\
\text{PRESS} & = \text{working} \\
\text{DIRECTION} & = \text{down} \\
\text{STOP\_UPPER\_POINT} & = \text{yes} \\
\implies & \\
\text{HANDS} & = \text{busy}
\end{align*}
\]
- In mode M4, the front door must be closed

\[
\begin{align*}
\text{mode} &= M4 \\
\text{PRESS} &= \text{working} \\
\implies \quad \text{FRONT\_DOOR} &= \text{closed}
\end{align*}
\]

- In mode M4, the side door must be closed

\[
\begin{align*}
\text{mode} &= M4 \\
\text{PRESS} &= \text{working} \\
\implies \quad \text{SIDE\_DOOR} &= \text{closed}
\end{align*}
\]
- 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

\[ DIRECTION = \text{up} \implies \text{STOP UPPER POINT} = \text{no} \]
- Observe the guard strengthening

\[
\text{start\_press} \equiv \\
\text{when} \\
PRESS = \text{stopped} \\
mode = M2 \implies \text{HANDS} = \text{busy} \\
mode = M3 \text{DIRECTION} = \text{down} \\
\text{STOP\_UPPER\_POINT} = \text{yes} \implies \text{HANDS} = \text{busy} \\
mode = M4 \implies \text{FRONT\_DOOR} = \text{closed} \\
mode = M3 \implies \text{SIDE\_DOOR} = \text{closed} \\
mode = M4 \implies \text{SIDE\_DOOR} = \text{closed} \\
mode \neq M5
\text{then} \\
PRESS := \text{working} \\
\text{end}
\]
2nd model: Freeing Hands (1st case) (Refined Version)

- 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 yet happened

\[
\text{free\_hands} \equiv \\
\text{when} \\
PRESS = \text{working} \\
HANDS = \text{busy} \\
mode \neq M2 \\
mode = M3 \implies DIRECTION = \text{up} \lor STOP\_UPPER\_POINT = \text{no} \\
\text{then} \\
HANDS := \text{free} \\
\text{end}
\]
2nd model: Freeing Hands (2nd case) (Refined Version)

- 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

\[
\text{stop\_press\_free\_hands} \iff
\begin{align*}
&\text{when} \\
&PRESS = \text{working} \\
&HANDS = \text{busy} \\
&mode \in \{M2, M3\} \\
&mode = M3 \implies DIRECTION = \text{down} \\
&mode = M3 \implies STOP\_UPPER\_POINT = \text{yes} \\
&\text{then} \\
&PRESS, HANDS := \text{stopped, free} \\
\end{align*}
\]
3rd model: Introducing motor and clutch

\[
\begin{align*}
&MOTOR \in \{ \text{stopped, working} \} \\
&CLUTCH \in \{ \text{disengaged, engaged} \}
\end{align*}
\]

Abstract variable \( PRESS \) will disappear

One is going to link \( PRESS \) with \( MOTOR \) and \( CLUTCH \)
3rd model: the Linking Invariant

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

\[
\begin{align*}
\text{mode} &= M1 \iff \text{MOTOR} = \text{stopped} \\
\text{mode} &= M5 \iff \text{MOTOR} = \text{stopped} \\
\text{mode} &= M5 \iff \text{CLUTCH} = \text{disengaged} \\
\text{CLUTCH} = \text{disengaged} &\iff \text{PRESS} = \text{stopped}
\end{align*}
\]
3rd model: the Linking Invariant (cont’d)

- In mode M1, the press works if the clutch is engaged:

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

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

\[
\begin{align*}
\text{mode} &\neq M1 \\
\text{MOTOR} &= \text{working} \\
\text{CLUTCH} &= \text{engaged} \\
\Rightarrow & \\
\text{PRESS} &= \text{working}
\end{align*}
\]
3rd model: Starting the Press (Refined Version)

\[
\text{start	extunderscore press} \iff \\
\quad \text{when} \\
\quad \quad \text{CLUTCH} = \text{disengaged} \\
\quad \quad \text{mode} \neq M1 \implies \text{MOTOR} = \text{working} \\
\quad \quad \text{mode} = M2 \implies \text{HANDS} = \text{busy} \\
\quad \quad \text{mode} = M3 \text{DIRECTION} = \text{down} \\
\quad \quad \quad \text{STOP	extunderscore UPPER	extunderscore POINT} = \text{yes} \implies \text{HANDS} = \text{busy} \\
\quad \quad \text{mode} = M4 \implies \text{FRONT	extunderscore DOOR} = \text{closed} \\
\quad \quad \text{mode} = M3 \implies \text{SIDE	extunderscore DOOR} = \text{closed} \\
\quad \quad \text{mode} = M4 \implies \text{SIDE	extunderscore DOOR} = \text{closed} \\
\quad \quad \text{mode} \neq M5 \\
\text{then} \\
\quad \quad \text{CLUTCH} := \text{engaged} \\
\text{end}
\]
Before starting motor **clutch must be disengaged**

```
start_motor  \equiv
  when
    MOTOR = stopped
    CLUTCH = disengaged
    mode \neq M1
    mode \neq 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

- 9th model: Refining movement (the cams)

- 10th model: Refining the 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),
• 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 (10th refinement),
- brake (11th refinement),
- cam (11th refinement),
- front door (11th refinement),
- motor (11th refinement),
- clutch (11th refinement),
- modes (13th refinement),
- foot (11th refinement),
- left hand (11th refinement),
- right hand (11th 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