

Mistake Proof Code

Emmanuel CHENU

emmanuel.chenu@fr.thalesgroup.com

<http://emmanuelchenu.blogspot.com>

In **IMPLEMENTING LEAN SOFTWARE DEVELOPMENT**, the authors, Mary and Tom Poppendieck give an example of mistake-proofing (pages 6 and 7):

« A properly mistake-proofed system will not need inspection. My video cable is an example of mistake-proofing. I can't plug a monitor cable into a computer or video projector upside down because the cable and the plug are keyed. »

This example really puzzled me: How can I translate this into my code?

Of course there are the automated self-checking tests, but there is also design-by-contract with assertions. Just have a look at this short example (in Java). The code illustrates the example of a video projector and a monitor cable.

```
public class VideoProjector {
  /*****
  * Connect a monitor cable.
  * PRE monitorCable is not null.
  * POST ... some post-condition ...
  */
  public void plug(MonitorCable monitorCable) {
    assert monitorCable != null : "PreCond: monitorCable != null";
    // some code ...
  }
  ...
}
```

If you call operation plug with a null monitorCable argument, then the assertion halts the application and you get the following message:

```
run:
Exception in thread "main" java.lang.AssertionError: PreCond: monitorCable !=null
at zeroinspection.VideoProjector.plug(VideoProjector.java:14)
```

Therefore, the code of classes VideoProjector and MonitorCable are keyed to be a mistake-proof collaboration.

