Communicating Process Architectures 2016
K. Chalmers, J.B. Pedersen et al. (Eds.)
Open Channel Publishing Ltd., 2016

Messing Around with Timeouts. In Contracts?

Fringe Presentation

Øyvind TEIG [1]

Autronica Fire and Security AS [2]*, Trondheim, Norway*

Abstract. Many (embedded) systems are often designed with timeouts at places where they are not needed or even wrong. When there is a timeout, it may break the very idea of how a contract should be: without timeout. For example, some response from an internal communication driver (that handles an external connection) may be awaited for with a timeout when it might be better just to wait for a proper response from the driver telling that the connection is indeed broken (i.e. that the driver performs the timeout). Timeout has a dimension of layer associated with it. For the above example, the timeout was properly handled by the driver to detect the broken connection, not by the client. Timeouts are, of course, appropriate for periodic processes like blinking an LED or pinging a line, where the connotation of a timer is used. However, having timeouts *between* internal communicating processes quickly makes matters difficult. We define timeout not to be part of a contract *per se* (even if this is rather contrary to judicial understanding of the term, where an expiration date often is necessary). In *design by contract*, failed critical assertions may be handled locally and the whole system may restart – best detected during testing, before final release. Formally verifying a system or using deadlock-free patterns to ensure a correct design would be better.

Requiring (over an external link that has a timeout of 5 seconds) that a heating element must increase the temperature after the push of a button (within 4 seconds) and a response must be shown on the display (by 3 seconds) should trigger a (hopefully) interesting and useful discussion of the specifications. We shall also consider the process network shown in Figure 1, where timeouts are indicated by three-sided arrows and labelled "t1", "t2" etc. The timer "t2" may not be needed and the system much simplified without it.[4]

Keywords. Timeout, layered design, design by contract

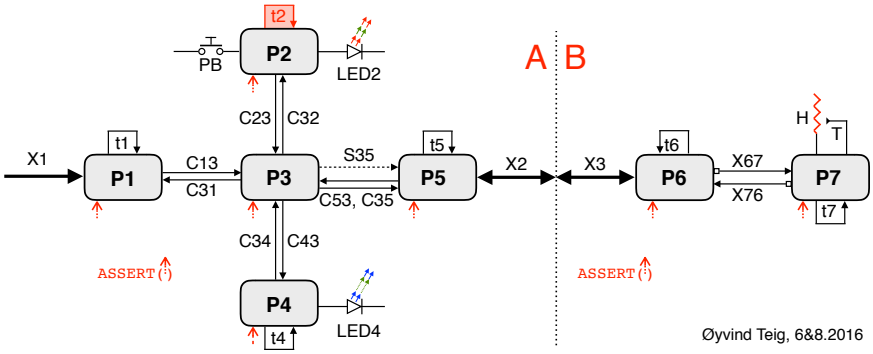

Figure 1. Process/data-flow diagram of a possibly unnecessary timer.

[1] (Private) *Øvre Møllenberg 11, 7014 Trondheim, Norway*. E-mail: `oyvind.teig@teigfam.net` .

[2] Autronica is a part of UTC Building & Industrial Systems, a unit of United Technologies Corporation.

[4] This abstract is based on the blog note *""Timing out design by contract with a stopwatch"* by Øyvind Teig. See `http://www.teigfam.net/oyvind/home/technology/128-timing-out-design-by-contract/` .