

CITS4401 Software Requirements and Design

Semester 1, 2020

Workshop week 11 – Non-OO design

For each of the following systems, discuss whether one of the alternative approaches to design that we have looked at might be appropriate.

1. A website for a graphic design business. The business owner wants to make it easy for customers to upload existing artwork (logos, designs, etc) and specifications, and download finished artwork. Customers should be able to log onto the system to upload or download files.
2. Embedded software for an insulin pump. The system monitors a diabetic patient's blood sugar levels and their rate of change. If necessary, it injects insulin (to mimic the normal function of the pancreas). If incorrect amounts are injected, a patient may go into a diabetic coma and die. Medical products like this are heavily governed by legislation and regulations.
3. A compiler for a new language. It takes as input *source code* files, and produces as output *machine code*. The structure of a program being compiled is quite complex. Some of the language features that need to be represented are:
 - source files
 - modules (source files can contain multiple modules)
 - imports and exports (what a module uses from other modules, and what it makes available)
 - functions, parameters, local variables, statements, blocks of statements, and types

In general, a program forms a complex hierarchy, with *source files* at the top, and statements and variable declarations down the bottom. The elements of a hierarchy don't have to "act" concurrently or independently – indeed, they often have fairly minimal behaviour to be implemented.

4. A dropbox-like service: customers install an application to access the service, and specify that specific folders need to be synced with a remote version.