

IDENTIFYING GENERIC FUNCTIONALITY^{*}

Alex Voss

Marzieh Asgari-Targhi

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0[†]

Economies of scale can be realised where effort invested can be reused. While, arguably, the mechanics of software reuse are relatively well understood, identifying just what is and what is not generic and how generic components can be adapted to specific purposes, remains a challenging task.

"the amount of work that you have to put in to getting something that works for [a service] for example I mean they have a very different access model to HPC and you know if you're going to have a sort of common look and feel to both to them I think you may struggle and you'd struggle to get something that was workable and met the needs of the different communities and they're very significant practical issues." (service provider)

"what I fear is that the e-science software so far has been too focused on building general infrastructure and maybe they better off solving, fewer, smaller [...] problems but at least solving them properly [...] rather than try to write something so general that everybody should be able to use it but in the end no one will use it." (researcher)

^{*}Version 1.2: Jun 7, 2009 5:06 pm +0000

[†]<http://creativecommons.org/licenses/by/3.0/>