

HW4 (team): Architecture

In this assignment, you will design the architecture for your system.

Submit a PDF containing the following parts:

- Give a dataflow diagram showing a high-level architecture of your system (approx 1 page)
- Give a dataflow diagram showing an alternate high-level architecture of your system; this alternate architecture should have a different architectural style than your first architecture (approx 1 page)
- Identify the key quality attributes (from the list below) for your system and assess how well each of the two architectures supports each quality attribute (approx 1 page)
 - Quality attributes: Reliability, Efficiency, Integrity, Usability, Maintainability, Testability, Flexibility, Portability, Reusability, Interoperability
- Identify two failure modes; for each mode, draw a fault tree and explain which of your two architectures is probably more prone to failure (and why) (approx 2 pages)
- Based on the quality and failure assessment, select one of your two architectures for further decomposition. Identify two important elements in your selected architecture, and for each element, give a lower-level dataflow diagram (approx 2 pages)
- Validate your selected architecture: walk through the use cases (from HW3) and succinctly describe how the architecture supports each use case (approx 1 page)
- Verify your selected architecture: succinctly answer each of the following questions (from Section 5.9 p. 278) (approx 1 pages):
 - Is the architecture modular, well structured, and easy to understand?
 - Can we improve the structure and understandability of the architecture?
 - Is the architecture portable to other platforms?
 - Are aspects of the architecture reusable?
 - Does the architecture support ease of testing?
 - Does the architecture maximize performance, where appropriate?
 - Does the architecture incorporate appropriate techniques for handling faults and preventing failures?
 - Can the architecture accommodate all of the expected design changes and extensions that have been documented?
- Explain the implications: how would you revise your selected architecture based on the results of the validation and verification? And how would this impact the quality attributes and potential for failure? (approx 1 page)
- Briefly summarize the contribution of each of your team members.

Some comments

Your work will be graded based on whether you appear to have produced an architecture that is valid (that meets the requirements) and verified (that follows good design principles). The page estimates above total 10 pages, but you may turn in up to 15 pages at your discretion.

You can divide this work however you like among your team, but here is a suggested approach that would complete the assignment very efficiently...

- Day 1, two team members each create a dataflow diagram showing a high-level architecture. They email these two "candidate" architectures to the entire team.
- Day 2, two team members meet to identify quality attributes. They then evaluate the candidate architectures and write up the results, which they email to the entire team.
- Day 2 (simultaneously), two other team members meet to identify failure modes. They evaluate the candidate architectures and write up the results, which they email to the entire team.
- Day 3, two team members meet to choose an architecture for decomposition. They each decompose one architectural element. They email results to the entire team.
- Day 4, two team members meet to validate how well the architecture supports the use cases. They write up and email results.
- Day 4 (simultaneously), two other team members meet to verify the architecture's conformance to design principles. They write up and email results.
- Day 5, one person combines the results into a single document and adds the final page explaining how the architecture should be modified.

Copyright (c) Christopher Scaffidi 2009 All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Oregon State University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Modified by Scott D. Fleming <Scott.Fleming@memphis.edu> 2011.