

CS3301 – Component Technologies

Assignment: Coding – Design and implementation of a simple pub-sub system

Deadline: 13 Apr

Credits: 40% of coursework and 16% of overall module mark for CS3301

You are expected to have read and understood all the information in this specification and any accompanying documents at least a week before the deadline. You must contact the lecturer regarding any queries well in advance of the deadline.

Aim / Learning objectives

To gain hands-on experience of developing a simple pub-sub system.

- Understand how to structure messages and message queues
- Understand basic interfaces of operating around events in a pub-sub system
- Understand QoS and appreciate the challenges of guaranteeing strong QoS

Requirement

You will need to build a publish-and-subscribe message-passing system that is as robust and scalable as you can make it, and demonstrate that this is the case. Extend the system to address different issues that cause problems, including but not limited to, temporary interruptions of connection; crashing queues; crashing consumers; increasing populations of producers and consumers; long delays in network traffic; dropped messages; out-of-order messages; duplicated messages; and so forth.

The key to the practical is to **demonstrate** that your solution works. For this you'll need to be able to "inject" the sorts of faults you want to guard against, and measure the ways in which your system deals with them. So for example you might want to be able to set things up so that you can kill a message queue and check that all messages still get through.

You can use any base middleware you like: Java RMI, CORBA, SOAP, XML RPC, something with a binding into your favorite language. As long as you don't choose one that is basically a pub-sub system itself, it's fine.

The main, compulsory, part of the practical should cover the following functionalities:

- R1. Subscribing and publishing events;
- R2. Lookup, discovery, and access of event channels;
- R3. Dynamic message queue management.

The following issues need to be considered:

- R4. Temporary interruptions of connections;
- R5. Crashing queues;
- R6. Crashing customers;
- R7. Increasing number of queries;
- R8. Long delays in network traffic;
- R9. Dropped messages;
- R10. Out of order messages;
- R11. Duplicated messages;
- And so on.

Submission

Submit the work via MMS to the P2 slot, as a single .zip file containing

- All your source files. This should be well presented and commented and in a form that can be compiled by the markers. If you are developing a mobile app, I will need you to demonstrate the app to the markers for assessment.
- A report of **no more than 1000 words** is structured as follows:
 - o **Overview:** Give a short overview of the practical.
 - o **Design and implementation:** Describe the design of your program. Justify the decisions you made.
 - o **Testing:** Describe how you tested your program. Your report should include the output from a number of test runs to demonstrate that your program satisfies the specification.
 - o **Questions:** include a short paragraph to explain why you think that the tests you submitted as output are sufficient to indicate that your program satisfies the specification.
 - o **Evaluation:** Evaluate the success of your program against what you were asked to do.
 - o **Conclusion:** Conclude by summarising what you achieved, what you found difficult, and what you would like to do given more time.
 - o **Instruction:** Describe how the marker should test your code, including installing external libraries/tools, configuration, and testing environment.

Assessment Criteria

Marking will follow the guidelines given in the school student handbook (see link in next section).

Some specific descriptors for this assignment are given below:

Mark range	Descriptor
1-6	Little or no attempt has been made on the assessment.
7-10	Reasonable attempt at R1, R2, and R3. A basic implementation of publisher, subscriber, message, and queue, and a basic implementation of their interfaces as defined in the lecture slides, including publish, subscribe, lookup/discover, etc.
11 – 13	A good attempt at R1, R2, and R3, with robust fault management on two of R4, R5, or R6+ good attempts at designing dynamically evolving queues. The report is well-written and clearly describes design decisions, evaluation process and results.
14 - 16	The above with robust fault management on two of R7, R8, or R9. A detailed description on evaluation and results: good justification of experiment setup and methodology, detailed description of design and evaluation process, and critical analysis of the results. The report is of a high academic standard.
17 - 20	The above with robust fault management on either R10 or R11. The report is of an exceptional academic standard.

Policies and Guidelines

Marking

See the standard mark descriptors in the School Student Handbook:

http://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptors

Lateness penalty

The standard penalty for late submission applies (Scheme B: 1 mark per 8 hour period, or part thereof):

<http://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#lateness-penalties>

Good academic practice

The University policy on Good Academic Practice applies:

<https://www.st-andrews.ac.uk/students/rules/academicpractice/>