

## Winter 2016 Project

Software Architecture

COMP 529

**Purpose:** The student will develop the skills needed to construct a software architecture using reverse engineering techniques. To practice working in a team environment effectively. To learn how to communicate software architectures in commonly accepted written and verbal formats. Exposing the classroom to multiple technologies and architectures through class presentations.

**Overall project guidelines:** The student will use the ACM document writing style for submissions 1 and 2 of this project (please review assignment 1). The student will use the minimization of energy principal when communicating their ideas for all three submissions (please review assignment 2). The student will demonstrate critical judgement, both in identifying positive and negative software elements. All critical statements must be justified (please review assignment 3).

**Basic grading guidelines:** The evaluation of all three submissions will be based on variants of the following general scheme:

A\* (100), A+ (90), A(85), A- (80), B+ (75), B (70), B- (65), C+ (60), C (55), D (50), F (0).

- A level: Well-articulated insights showing evidence of an in-depth understanding of your case study and/or that synthesize knowledge from multiple sources.
- B level : Useful and correct observations relying on a basic understanding of the material or closely related facts.
- C level: Some new insights but problems of correctness, or overly shallow arguments.
- D level : Relevant facts trivially extracted from the sources with only minimal additional reasoning.
- F level: Generalities and other platitudes too vague to be useful (e.g., “Loose coupling is desirable.”).

**Academic Integrity:** This project will require you to work with many documents produced by other people, and it is understood that part of your task will be to integrate the intellectual contributions of others into your own work. In this situation, you must be especially careful to cite and reference all sources. If a piece of intellectual property is not referenced, the assumption is that you are its author or creator. For examples, all figures taken from external documents must be referenced. Paraphrasing should be used with extreme caution, if at all. The McGill Fair Play website provides many resources that clarify the academic integrity issues at stake.

Submitting a deliverable (submission 1, 2 and 3) on behalf of a group carries the implication that all group members have made non-trivial contribution to the deliverable. If that is not the case and one member has made no or only symbolic contributions to the project, you should notify the instructor within 24hrs of submission. I reserve the right to adjust grades of individual group members if presented with clear evidence that contributions to the project were highly unequal.

**Submissions and due dates:** This project has 3 submissions. The first two submissions are written based on the ACM style as seen from assignment 1. The last submission is a classroom presentation. Basic submission information:

- Submission 1, due: Nov 15, 2015, ACM-based architectural description. A decomposition and the overall architecture without opinions of your case study (as selected from assignment 3).
- Submission 2, due: Dec 6, 2016, ACM-based augmentation of submission 1 containing the critical analysis of your case-study architecture. It will contain educated opinions with justification of the pros and cons in the architecture. You may propose alternative architectures in this submission after presenting your critical analysis and justifications.
- Submission 3, due: last 2 or 3 classes in Nov 22 – Dec 6 2016 by appointment, in class 10-minute team presentation. The presentation must exhibit energy minimization and demonstrates a thorough covering of your case-study architecture with critical analysis (or highlights thereof).

Submission 1 [20% of grade]

**What to submit:** Submit to My Courses three documents. The first document is a PDF paper formatted as per the ACM guidelines. The second document is a milestones table detailing what you did (historical) for the preparation of the paper, as a PDF. The last submission is an appendix, as described bellow. Only the team leader will submit these documents. Make sure that the documents have all the team member names so that everyone is graded.

**Requirements:**

- Do a full, energy sensitive, **ACM paper**-based architectural description of your target case-study as selected from assignment 3. The maximum length of your paper, including references, is 3 pages. Your paper should include the following:
  - **Introduction.** Describe the application domain and main functionality of the system, and anything else relevant to the case study. The introduction to most of the articles in “The Architecture of Open-Source Applications” are good examples.
  - **Technology.** Describe the main underlying technology. This will include the programming language(s) and any major framework(s), design pattern(s), Style(s).
  - **Preliminary Overview.** Give a brief overview of what you managed to learn about the architecture of the system by installing it, studying the libraries, and computing the required statistics. Summarize the statistics in a table or chart.
  - **Preliminary Architectural Description.** Provide an initial description of the components of the project and how they operate. Provide a single static UML diagram that details the parts, connectors, and attributes. Provide a single dynamic UML diagram that describes the primary data or execution flow of the program.
  - Do not provide a **conclusion**, this is only your first submission.
  - **References.** Include a reference section where you list all pertinent reference material. Your list of reference must include, minimum, the most relevant document about your system. Make sure to cite, using a number, throughout your document when you are using material from your reference in the text.
  - Above the 3 pages of the paper, provide an appendix with the following:
    - **Metrics.** Provide a table with the following metrics: number of lines of code, number of lines of comments, documentation mass, programming language, distribution information, library to core ratio in the distribution, function count / module count / library count / object count, speed measurements, slowest part of code, fastest part of code, directory structure, and compile time.
    - **Scripts.** Briefly describe the strategy you employed to compute the metrics. This can be as simple as “Three Python scripts using regular expressions

doing A,B,C”.

- **Scripts 2.** Submit your actual scripts here.
- **Contributions.** Briefly indicate what each group member has contributed to the milestones.
- Create a **milestones** table recording the steps your team performed when preparing the paper.
- Suggested steps you should perform:
  - Download, install and run your open source case-study application
  - Download or acquire all written documentation of your case-study application, read, find out what already is said about the project... then, verify if those sources are incorrect!
  - Determine what parts of the download are external libraries and not part of the system and what files belong to the system.
  - Do basic computational analysis: Write scripts that will compute the following data about your system: number of non-blank lines of source code (excluding external libraries), number of libraries, size (in MB) of each library and total size of required libraries, number of primary units of decomposition (packages, namespaces, directories). You must compute the statistics yourself: do not simply reproduce the data provided by aggregator websites. Do not consider libraries that are part of the standard distribution for a technology platform (e.g., Java JRE, .NET class libraries, etc.).

#### Submission 2 [20% of grade]

**What to submit:** Submit to My Courses the fully completed ACM paper as a PDF file (submission 1 + submission 2). The paper follows the ACM guidelines. Only the team leader will submit these documents. Make sure that the documents have all the team member names so that everyone is graded.

#### **Requirements:**

- Do a full, energy sensitive, **ACM paper**-based critical architectural description and analysis of your target case-study as selected from assignment 3. The maximum length of your paper, including references, is 8 pages. It will contain the 3 pages from your submission 1 edited and improved, plus the additional pages of your critical analysis. Your paper should include the following:

##### ----- INTRODUCTION

- **Abstract.** The abstract's role is to provide the means by which the reader will decide whether they should invest the energy to read your paper. Is it worth their time? A concise advertisement based summary of what the paper is about with a hint of what it might prove. 'Advertisement' light and professional summary heavy.
- **Introduction.** An improved version of your submission 1 introduction. What is the open source program about? Define or describe the problem space. Why is the open source project important to the community? Its overall adoption in the world at large. Is there any interesting history or current events (keep this short)? Define or describe the solution space. Give a synopsis of how you will present your findings in this paper (even if this might sound a little like the Abstract). Like the Abstract is a tool for the reader to see if they want to read your paper, the Introduction and Conclusion is the next tier down. I read your Abstract to see if I should read the paper. If I have not outright discarded your paper due to the abstract but I am still not fully convinced of your paper I will then read the Introduction and Conclusion to see if the paper is worth my time. This means you need to write the Introduction and Conclusion as a unit, even though they appear at vastly different locations in the

document. The Introduction and Conclusion must be understandable to a reader who has not read your paper!

- **Technology.** An improved version of your submission 1 technology identification. You will have likely learned new things on the way, made corrections based on comments from the TA/Prof evaluating your submission 1. There are two basic things to talk about: the technology used to compose the software, “Software Architecture” (libraries, components, run-time environment, major software subdivisions), and the identification (naming) of major Styles, Architectures and Design Patterns present in the application “Architectural Description”. This is still part of the introduction to the paper. The reader would use this part to get their first technical summary. Readers may stop reading your paper after this point, so provide a good technical summary. There is no analysis.
- ----- DISCOVERY PROCESS (the groundwork for your proofs)
  - **Installation Overview.** An improved version of your Preliminary Overview. Include additional analytical techniques you employed above what was required in submission 1. Summarize these new results in a table or chart or just text. (If writing code was required then also include that as an appendix, not included in the 6 page limit) Provide a UML deployment diagram.
  - **Discovery Process Summary.** Summarize the process you followed to discover the architecture.
  - **Critique of existing context.** Give a short critical discussion concerning the validity of existing documentation that you found purporting accurate descriptions of your target case-study. Highlight the best and worst sources and why you have identified them as such.
- ----- THE ARCHITECTURE (Software & Style)
  - **Architectural Description.** This is an expansion to your submission 1 description. In this version you may use as many diagrams as you like to support your full and complete architectural description. Make sure to identify all the important styles, frameworks, design patterns, attributes, etc. Follow the terminology and standards seen in class. For additional guidance, feel free to browse “Documenting Software Architecture”. Here you must go into detail covering the “Software Architecture” and the “Architectural Description”. This expands what you outlined in the Technology section, above. Here you will educate your reader about the program before you do the analysis.
- ----- JUSTIFIED ANALYSIS
  - **Critical Analysis.** Give a critical analysis of your target case-study. Identify the weaknesses and strengths of the architecture. Justify all your claims. Use assignment 3 as a model of what you would need to do. Identify at least 3 non-trivial discoveries.
  - **Alternatives.** This section is written as a summary or an overview. Identify an alternative way of implementing your target case-study. Do one of the following: (a) find a better architectural solution that overcomes the weaknesses present in the current design, with justification. Or, (b) if you think the design is already optimal, find the next best design and state why there is no other design between your choice and the optimal, and justify.
- ----- CONCLUDING (Note your intro & conclusion should stand alone for lazy readers)
  - **Conclusion.** Summarize what you have learned. Is this a good application? Is it a good installation? How should it be improved, or what should the next version of the application contain? Rate the developers and the application. How would you have done it differently? What good things would you like to highlight?
  - **References.**
  - **Appendix (optional)**

Things to keep in mind with this submission:

- ✓ Completeness. All the requirements are met. All the required data is present.
- ✓ Depth. The observations are based on in-depth and original investigation of the implementation artifacts.
- ✓ Rigour. Observations are supported by evidence. Facts, observations, figures, etc. are precise and based on well-defined terms.
- ✓ Relation to class material. The description and views are framed in terms of the concepts and terms seen in class. These are used correctly.
- ✓ Clarity. The description of the project and the synthesis of the existing knowledge are clear and insightful.
- ✓ Quality. The writing is free of typos and grammatical mistakes. The style is crisp and consistent.
- ✓ Be organized: keep notes on your discovery process, tool usage, and the links between your observations and the documentation of the architecture.

### Grading Rubric

Email me for your graded files (if you want them)

#### ----- Paper composition

- ( / 5) Introduction
- ( / 10) Technology
- ( / 10) Installation Overview
- ( / 5) Discovery Process Summary
- ( / 10) Architectural Description
- ( / 5) Critique of existing context
- ( / 10) Critical Analysis
- ( / 5) Alternatives
- ( / 5) Conclusion
- ( / 5) References

#### ----- Writing quality

- ( / 4) Completeness
- ( / 4) Depth
- ( / 4) Rigour
- ( / 4) Relation to lectures
- ( / 4) Clarity
- ( / 4) Quality
- ( / 2) Organization
- ( / 4) Energy

#### ----- Special deductions

- ( ) AMC Standards (letter grade loss, -5)
- ( ) 8 Page limit (letter grade loss, -5)
- ( ) Use of UML (-5)

#### ----- Final grade

- ( / 100)

### Submission 3 [5% of grade]

**What to submit:** Submit to My Courses the Power Point presentation (or whatever) in its original file format, like .ppt plus a PDF version of that same presentation. Only the team leader will submit these documents. Make sure that the documents have all the team member names so that everyone is graded.

### Requirements:

- Make an appointment with the professor to schedule your class presentation date and time.
- Make sure to email the presentation to the professor the day before (or earlier) both in native and PDF format. You will be giving your presentation from the professor's laptop. Failure to do this will result in no presentation.
- Your presentation must be 13 minutes long. The class will be permitted 2 (maybe 3) questions following your presentation. 10 minutes for presentation. 1 setup. 2 questions.
- The presentation must summarize the contents of your submission 2.
  - Keep text to a minimum (eliminating text from a presentation is a mistake)
  - Use UML diagrams and other pictures to a maximum

- Use of tables and bullets should be short and easy to read
- Writing on the board to help with the explanation will deduct points from your grade. This includes when answering questions from the audience.
- Each team member must speak during the presentation.
- The goal of your presentation is to give the students in your class a good understanding of the architecture, run-time environment, your critical analysis, and your discovery process.
  - Your grade comes primarily from this item, plus your presentation techniques using what you learned from the course.