

Assignment 5 (2011): Development iteration 3 (the last one)

Due date: Monday, April 11, by 12 noon, to your supervisor

Deliverables: submit the Final Report according to guidelines given in [Required Format for Assignments and Final Project Report](#). Do not submit Assignment 5 report separately

1. Describe project plans

Describe updated plan for the whole project and for this development iteration. Follow formats from Assignment 3.

2. SPA implementation

- 1) Implement all the query types described in the Project Handbook and summarized in the Appendix A. Two outstanding issues are **pattern** clause and relationships **Affect** and **Affects***.

Remark: Your solution must be scalable, therefore pre-computing **Affects** and **Affects*** for all program statements and storing the it is not acceptable. Design efficient algorithms to compute **Affects** and **Affects*** on demand, during query evaluation.

- 2) Extend Query Pre-processor to validate queries and build query tree.
- 3) Extend Basic Query Evaluator (BQE) to evaluate all the remaining queries.
- 4) Implement optimizations. Integrate with BQE to form Query Evaluator.

Hints for query optimization:

The two main factors that have to do with effective query evaluation is the size of the intermediate result and the evaluation time. Optimizations should reduce the size of the intermediate result and/or the evaluation time. Changing the order in which you evaluate relationships in a query, as well as extra information about the source program may help you evaluate queries in more effective way. Assess various optimizations analytically before you implement them.

3. Update (if necessary) documentation of abstract PKB API

Include complete, updated documentation of abstract PKB API in the assignment report.

4. UML diagrams

Draw UML diagrams that you found useful. For each diagram that you draw, explain how you used it (e.g., in project planning, communication, test planning or in other situations), and comment on the value a diagram added to your project.

Hint: Read through relevant parts of Handbook Section 10. Refine UML sequence diagrams given in examples into more detailed sequence diagrams showing communication between SPA functional components and specific data abstractions in PKB.

5. Documentation of important detailed design decisions

Follow guidelines in Handbook Section 10.2 to properly analyze, justify and document detailed design decisions. Pay attention to clarity of the description (check hints in Section 10.2).

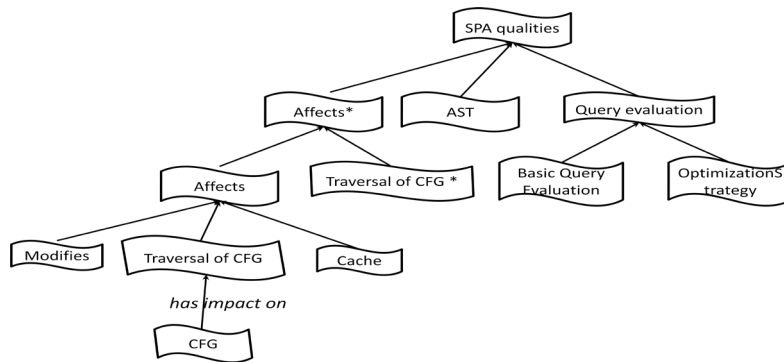
Address detailed design decisions related to data representations for design abstractions in PKB, any other solutions to speed up access to information stored in PKB, algorithms to fetch data from PKB during query evaluation, and any other issues that you consider important, except query evaluation (which is addressed in Section 7.2).

5.1. Documentation of design decisions

While you should document all the design decisions that you consider important, it is obligatory that you document your choice of design solution for relationship **Affects** and **Affects***. Include documentation of all design solutions from previous assignment.

Documentation of design decisions should consist of:

- 1) Diagram depicting inter-dependencies among design decisions. Expand example given below:



- 2) For each design problem:
 - a) State design goals relevant to this
 - b) Consider alternative design solutions
 - c) Evaluate design solutions in view of design goals. Use Big O notation to describe complexity of algorithms
 - d) Justify your choice of design solution
 - e) Document the above process

5.2. Documentation of design patterns

Design patterns provide standardized solutions to design problems. You studied some typical design problems for which published design patterns exist in CS2103. By applying a design pattern, you usually win more flexibility, but an overall program solution may be more complex to work with. Always evaluate carefully the trade-offs involved in terms of expected benefits and the cost of applying a design pattern. Apply a design pattern only if the benefits outweigh the cost.

If you applied design patterns, document them in this section:

- 1) Explain the design problem and pattern you applied to solve it
- 2) Document expected benefits and costs of applying a design pattern
- 3) Document the actual benefits and costs of a design pattern that you experienced in the project after applying it.

6. Coding standards and experiences

- 1) State coding standards adopted by your team.
- 2) Comment on how you used abstract PKB API to guide design of relevant C++ classes and how you keep abstract and concrete PKB API in sync with other.
 - a) Do you use same naming conventions in abstract PKB API and your C++ program?
 - b) Do you use typedef to map type names used in abstract PKB API to C++ types?
- 3) Comment on any experiences - problems and benefits - that you observed working from abstract PKB API towards C++ program.

7. Query processing

7.1. Validating program queries

Describe query validation rules, only in case there is some difference as compared to what you described in your previous assignment. An example of query validation rule is: "checking if all relationships have correct number and types of arguments, as defined in PQL definition in Handbook". DO NOT provide procedural description (pseudocode) of how Query Pre-processor checks the rules.

If you use table-driven approach to query validation – show the structure of your tables.

7.2. Design and implementation of query evaluator

- 1) Describe data representation for program queries
- 2) Describe your strategy for Basic Query Evaluation (BQE)
- 3) Describe optimizations

- 4) Document detailed design decisions regarding BQE and optimizations, in the same style as you documented design decisions in the above point 5.2.

Hint: Follow guidelines in Handbook Section 10.2 to properly analyze, justify and document detailed design decisions regarding BQE and optimizations (if any). Pay attention to clarity of the description (check hints in Section 10.2). Do not repeat what you already discussed in Section 5, but refer to relevant points.

8. Testing: Group-PKB and Group-PQL

Describe your testing experience (not exceeding TWO pages).
Do not include sample test cases.

9. Project evaluation

Free format. Here are examples of issues that you might address (do not feel obliged to address all of them and you can discuss issues not mentioned here):

1. How would you improve your SPA if more time was available?
2. What would you done differently if you were to start project again?
3. Comment on the experience gained in this project in respect to:
 - a) working in the team,
 - b) incremental development,
 - c) complexity of the SPA problem and program solution,
 - d) what did work well?
 - e) what did not work well?
 - f) what did you learn in this project course?
4. Comment on the tools used for the project
 - a) Were the recommended tools useful?
 - b) What other tools did you use (if any), and in what ways were they useful?
 - c) What were the problems you faced when using each tool?
 - d) In which areas would you like to have had more tool support?
5. What management lessons have you learned?
6. What advice would you give to the students who will take this course in the future?
7. Suggest how we could improve this project course.
8. Discuss any other experiences.

--- The End ---