Detailed Design

Greg Butler

Computer Science and Software Engineering Concordia University, Montreal, Canada

Email: gregb@cs.concordia.ca

Detailed Design

Detailed Design Definition [ISO/IEC 24765]

1. the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to be implemented;

2. the result of the process in (1).

OO Software Detailed Design — How To

Sketch CRC (Class-Responsibility-Collaboration) for each class UML diagram for system objects and their collaboration Determine interfaces (ie operations) of each class Specify contract for each operation Select algorithms and data structures for each class Describe algorithm for each major operation using UML note

Example: CRC Cards for Model-View-Control Architecture

| Display Model | Controller Model | |
|--|---|--|
| | Controller | |
| | Interpret User Input View Send changes to Model Model | |
| Model | | |
| | | |
| Maintain model info Notify Views of changes | | |

Example: Collaborations in UML

Sequence Diagram for Observer Pattern



Example: Collaborations in UML

Activity Diagram for Order Processing



Example: Collaborations in UML

Communication Diagram for Model-View-Control



Example: Class Interfaces in UML

UML Class



Example: Pre-conditions and Post-conditions

```
issueBookToStudent(Student s, Book b)
pre:

    s is in students, b is in books

2. b is not issued to anyone
3
  there are no reservations for b or
   if there are reservations for b, e is
    the first one
post:
1. old(students) and students are identical
old(books) and books are identical
3. s has been issued the book b
there is no reservation of s for b
```

Example: Java Assertions



Example: Larman System Operation Contract



Example: Constraints using OCL in UML Object Constraint Language for Tic-Tac-Toe



Example: UML Note for Observer Pattern



Detailed Design Example — Count Substrings

```
//Construct collection C of
// triples <pattern, cnt, position>
for each pattern p in alphabet<sup>k</sup> do
    C[ p ] := < p, 0, empty_list >;
end for
```

```
for i := 0 to n-k do
    ss = S.substring(i, i+k-1));
    C[ ss ].count++;
    C[ ss ].list.append(i);
end for
```

```
Count operations
& data movement
```

|alphabet|^k iterations

```
n - k iterations
data movement?
indexing cost?
indexing cost?
```

Example — Count Substrings — Hand Execution Small Example

Alphabet = { a, c, g, t } String S = ataaaa size n = 6

Substrings of size k = 1Counts Substrings Positions a : 5 a: 5: 0.2.3.4.5 а t:1 t:1:1 t Total = nSubstrings of size k = 2Counts Substrings Positions aa : 3 aa: 3: 2,3,4 аа at : 1 at: 1:0 at ta : 1 ta:1:1 ta Total = n - 1

Compute substrings, counts, and positions!

Detailed Design Example — Count Substrings — Quality

Quality Concerns

Correctness

system must compute the right answers!

Efficiency = Resource Usage Computation time, memory, disk, elapsed time Formulas in terms of *n*, *k*, size of alphabet

Scaleability

size *n* of string *S*, size *k* of substring *P* string *S* may be of size $k = 10^{10}$ or more size *k* of substring *P* is often 17 to 37 potential number of different substrings *P* is $|alphabet|^k$ $4^{37} = 2^{78} = 10^{23}$ approximately Detailed Design Issues for Example — Count Substrings

Data representations character in the string the string the substring the collection of statistics

Algorithms

for enumerating each substring for updating statistics of a substring in the collection indexing and searching the collection of statistics

Interfaces (API)

String: how to iterate over string Collection: how to update the statistics for each substring

packed representation of characters in the string, or not?

pass-by-value versus pass-by-reference

Detailed Design — Use Data Structures in Libraries String

Containers or Collections or Map or Dictionary Vector and Array: indexed by scalar type, often fixed length

Set: no order, no duplicates List: order, duplicates Bag (or Multiset): no order, duplicates

Map: relates "key" and "value" pairs

Do not use library blindly!

Memory management Fixed size vs dynamic size Packed data representations Initialise expected size Set hash function High-Level Algorithm Descriptions: Set Comprehensions

Type Declarations: Design Example

A ::= { a, c, g, t }; //alphabet String ::= seq A; //Set of all finite sequences over A StatisticsContainer ::= map String $\rightarrow \mathbb{N} \times seq\mathbb{N}$;

Variable Declarations: Design Example

Input: S : String; k : \mathbb{N}^+ Output: C : StatisticsContainer; Local variables: p : String; //the substring

 $n : \mathbb{N}^+$; // size of S

High-Level Algorithm Descriptions: Set Comprehensions

Program: Design Example

//define the set of patterns P of size k let P = { p | p eq S[i..i+k-1] for some i, $0 \le i \le n-k$ };

//for a given pattern p, define the set of its starting positions i //take the size of the set to determine the count of pattern p let cnt(p) = | { i | $0 \le i \le n-k$ and p eq S[i..i+k-1] } |;

//for a given pattern p, define the list of its starting positions i let pos(p) = List{ i | $0 \le i \le n-k$ and p eq S[i..i+k-1] };

//the result is a map from pattern to a pair < count, list > return C := Map{ $p \mapsto (cnt(p), pos(p)) | p \in P$ };

Data Structures — Be Careful!

Character storage as one byte

For alphabet = { a, c, g, t } you only need 2 bits per character

Comparison operators on strings

How do you compare strings to index a collection of strings? Character by character? Do you copy the character from string?

Data movement in and out of string

Relationship between string and character Relationship between string and substring Pass-by-value versus Pass-by-reference

Memory management

Need extra memory allocated? may allocate new block of memory and move your data!

IO from disk to memory

IO is typically 30% of total computing time