# 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



## 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:

1. 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:
4. old(students) and students are identical
5. old(books) and books are identical
6. shas been issued the book b
7. there is no reservation of $s$ for $b$

## Example: Java Assertions

```
(J) BankAccount.java {3
    24e /**
    25 Deposits money into the bank account.
    26 @param amount the amount to deposit
    27 Precondition: amount >= 0
    28 Postcondition: getBalance() >= 0
    assertions
    29 */
    300 public void deposit(double amount)
    31
    32
            double newBalance = balance + amount;
            balance = newBalance;
34
```


## Example: Larman System Operation Contract

## Basic Contract Format for Operations

- Name
- Cross Reference
- Responsibilities
- Pre-Conditions
- Post-Conditions

enterltem( itemD, quantity)
Use case Process sale
Record Sale of an item and add it to the sale. Display item description and price.
Product itemID is known to the system.
SalesLineltem corresponding to product itemID was created.
SalesLinetem was
associated with the sale.
Instance.quantity became quantity.
Instance was associated with a
ProductDescription based on
itemid match.


## 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^k 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 ExampleAlphabet $=\{\mathrm{a}, \mathrm{c}, \mathrm{g}, \mathrm{t}\}$ String $S=$ ataaaa size $n=6$
Substrings of size $k=1$
Substrings

$$
\begin{aligned}
& \text { Counts } \\
& \text { a : } 5 \\
& t: 1 \\
& \text { Total }=n
\end{aligned}
$$

Positions
a: $5: 0,2,3,4,5$
$\mathrm{t}: 1$ : 1

Positions
aa: $3: 2,3,4$
at : $1: 0$
ta : 1 : 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 $\mid$ alphabet $\left.\right|^{k}$
$4^{37}=2^{78}=10^{23}$ 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 $\rightarrow \mathbb{N} \times s e q \mathbb{N}$;

Variable Declarations: Design Example Input: S : String; k: $\mathbb{N}^{+}$
Output: C : StatisticsContainer;
Local variables:
p: String; //the substring
$\mathrm{n}: \mathbb{N}^{+} ; / /$size of S

## High-Level Algorithm Descriptions: Set Comprehensions

## Program: Design Example

//define the set of patterns P of size k let $P=\{p \mid p$ eq $S[i . . i+k-1]$ for some $i, 0 \leq i \leq n-k\}$;
//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 $\operatorname{cnt}(\mathrm{p})=\mid\{\mathrm{i} \mid 0 \leq \mathrm{i} \leq \mathrm{n}-\mathrm{k}$ and p eq $\mathrm{S}[\mathrm{i} . \mathrm{i}+\mathrm{k}-1]\} \mid$;
//for a given pattern p , define the list of its starting positions i let $\operatorname{pos}(\mathrm{p})=\operatorname{List}\{\mathrm{i} \mid 0 \leq \mathrm{i} \leq \mathrm{n}-\mathrm{k}$ and p eq $\mathrm{S}[\mathrm{i} . \mathrm{i}+\mathrm{k}-1]\}$;
//the result is a map from pattern to a pair < count, list > return $C:=\operatorname{Map}\{p \mapsto(\operatorname{cnt}(p), \operatorname{pos}(p)) \mid \mathrm{p} \in \mathrm{P}\}$;

## Data Structures - Be Careful!

Character storage as one byte
For alphabet $=\{\mathrm{a}, \mathrm{c}, \mathrm{g}, \mathrm{t}\}$ 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!
10 from disk to memory
IO is typically $30 \%$ of total computing time

