



---

National Institute of Applied Sciences and Technology

CARTHAGE UNIVERSITY

## Graduation Project

Specialty : **Software Engineering**

---

---

**A flexible data analysis library for hybrid pixel detectors**

---

Presented by

**Bechir Braham**

INSAT Supervisor : **Dr. Guesmi Ghada**

Company Supervisor : **Dr. Froejdh Erik**

Presented on : **30/09/2024**

### **JURY**

M. President FLEN (President)

Ms. Reviewer FLENA (Reviewer)

Academic Year : 2023/2024



---

# Acknowledgements

Thank you all!

---

# Table of Contents

|  |           |
|--|-----------|
| <b>List of Figures</b>                                       | <b>iv</b> |
| <b>List of Tables</b>  | <b>v</b>  |
| <b>Abstract</b>  | <b>vi</b> |
| <b>General Introduction</b>                                  | <b>1</b>  |
| <b>I Project Context and Scope</b>                           | <b>3</b>  |
| 1 Presentation of The Host Company . . . . .                 | 4         |
| 2 Detector's Group Presentation and Work . . . . .           | 4         |
| 3 Problem Statement . . . . .                                | 4         |
| 4 Project Goals . . . . .                                    | 4         |
| 5 Work Methodology . . . . .                                 | 4         |
| 6 High Level Planning . . . . .                              | 4         |
| <b>II Requirement Specification and Overall Architecture</b> | <b>6</b>  |
| 1 Requirement Specification . . . . .                        | 7         |
| 1.1 Actors Identification . . . . .                          | 7         |
| 1.2 Functional Specification . . . . .                       | 7         |
| 1.3 Non-Functional Specification . . . . .                   | 7         |
| 2 Overall Architecture and Guidelines . . . . .              | 7         |
| 2.1 Onion Architecture . . . . .                             | 7         |
| 2.2 SOLID Principles . . . . .                               | 7         |
| 2.3 C++ Specific Design . . . . .                            | 7         |
| 2.3.1 Templates Metaprogramming . . . . .                    | 7         |
| 2.3.2 C++ Idioms . . . . .                                   | 7         |
| 2.3.3 C++ Limitations . . . . .                              | 7         |
| 2.4 Project Architectural Design . . . . .                   | 7         |
| 2.4.1 Project Modules . . . . .                              | 7         |
| 2.4.2 Project Architecture . . . . .                         | 7         |
| 2.4.3 Class Diagram . . . . .                                | 7         |

|   |           |
|---|-----------|
| <b>III Implementation of a Flexible Data Analysis Library for Hybrid X-ray Particle</b> |           |
| <b>Detectors</b>  | <b>9</b>  |
| 1    Project Setup . . . . .  | 10        |
| 1.1    Project Structure . . . . .  | 10        |
| 1.2    Build System . . . . .   | 10        |
| 1.3    Testing and Continuous Integration . . . . .                                     | 10        |
| 2    Core Module Implementation . . . . .   | 10        |
| 3    File IO Module Implementation . . . . .  | 10        |
| 4    Network IO Module Implementation . . . . .   | 10        |
| 5    Processing Module Implementation . . . . .   | 10        |
| 6    Python Bindings Implementation . . . . .   | 10        |
| <b>IV Library Applications and Evaluation</b>   | <b>12</b> |
| 1    Examples of Library Applications . . . . .   | 12        |
| 2    Parallelization and Multi-threading . . . . .                                      | 12        |
| 3    Performance Evaluation . . . . .   | 12        |
| <b>Conclusion and Perspectives</b>  | <b>13</b> |
| <b>Appendix : Miscellaneous remarks</b>   | <b>15</b> |

---

## List of Figures

---

# List of Tables

---

# Abstract

This is the english abstract of your project. It must be longer and presented in more details than the abstract you write on the back of your report.



---

# General Introduction

Pour écrire un bon rapport [Gre93] de projet en informatique, il existe certaines règles à respecter. Certes, chacun écrit son rapport avec sa propre plume et sa propre signature, mais certaines règles restent universelles [Gre93].

**La Table de matière** est la première chose qu'un rapporteur va lire. Il faut qu'elle soit :

- Assez détaillée <sup>1</sup>. En général, 3 niveaux de numéros suffisent;
- Votre rapport doit être réparti en chapitres équilibrés, à part l'introduction et la conclusion, naturellement plus courts que les autres;
- Vos titres doivent être suffisamment personnalisés pour donner une idée sur votre travail. Éviter le : Conception , mais privilégier : Conception de l'application de gestion des ... Même s'ils vous paraissent longs, c'est mieux que d'avoir un sommaire impersonnel.

**Une introduction** doit être rédigée sous forme de paragraphes bien ficelés. Elle est normalement constituée de 4 grandes parties :

1. Le contexte de votre application : le domaine en général, par exemple le domaine du web, de BI, des logiciels de gestion ?
2. La problématique : quels sont les besoins qui, dans ce contexte là, nécessitent la réalisation de votre projet?
3. La contribution : expliquer assez brièvement en quoi consiste votre application, sans entrer dans les détails de réalisation. Ne pas oublier qu'une introduction est censée introduire le travail, pas le résumer;
4. La composition du rapport : les différents chapitres et leur composition. Il n'est pas nécessaire de numéroter ces parties, mais les mettre plutôt sous forme de paragraphes successifs bien liés.

---

<sup>1</sup>Sans l'être trop

---

---

Part I

# Chapter 1

---

---

---

# Chapter I

---

## Project Context and Scope

### Summary

|   |  |   |
|---|--|---|
| 1 | Presentation of The Host Company . . . . .       | 4 |
| 2 | Detector's Group Presentation and Work . . . . . | 4 |
| 3 | Problem Statement . . . . .                      | 4 |
| 4 | Project Goals . . . . .                          | 4 |
| 5 | Work Methodology . . . . .                       | 4 |
| 6 | High Level Planning . . . . .                    | 4 |

### Introduction

Une étude théorique [[Knu](#)] peut contenir l'une et/ou l'autre de ces deux parties :

- 1 Presentation of The Host Company**
- 2 Detector's Group Presentation and Work**
- 3 Problem Statement**
- 4 Project Goals**
- 5 Work Methodology**
- 6 High Level Planning**

## **Conclusion**

La conclusion est en général sans numérotation, et n'apparaît pas dans la table des matières.

---

---

Part II

## Chapter 2

---

---

---

# Chapter II

---

## Requirement Specification and Overall Architecture

### Summary

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Requirement Specification . . . . .</b>           | <b>7</b> |
| 1.1      | Actors Identification . . . . .                      | 7        |
| 1.2      | Functional Specification . . . . .                   | 7        |
| 1.3      | Non-Functional Specification . . . . .               | 7        |
| <b>2</b> | <b>Overall Architecture and Guidelines . . . . .</b> | <b>7</b> |
| 2.1      | Onion Architecture . . . . .                         | 7        |
| 2.2      | SOLID Principles . . . . .                           | 7        |
| 2.3      | C++ Specific Design . . . . .                        | 7        |
| 2.4      | Project Architectural Design . . . . .               | 7        |

# Introduction

## 1 Requirement Specification

### 1.1 Actors Identification

### 1.2 Functional Specification

### 1.3 Non-Functional Specification

## 2 Overall Architecture and Guidelines

### 2.1 Onion Architecture

### 2.2 SOLID Principles

### 2.3 C++ Specific Design

#### 2.3.1 Templates Metaprogramming

#### 2.3.2 C++ Idioms

#### 2.3.3 C++ Limitations

### 2.4 Project Architectural Design

#### 2.4.1 Project Modules

#### 2.4.2 Project Architecture

#### 2.4.3 Class Diagram

# Conclusion

Faire ici une petite récapitulation du chapitre, ainsi qu'une introduction du chapitre suivant.

---

---

Part III

## Chapter 3

---



---

---

# Chapter III

---

## Implementation of a Flexible Data Analysis Library for Hybrid X-ray Particle Detectors

### Summary

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Project Setup . . . . .</b>                    | <b>10</b> |
| 1.1      | Project Structure . . . . .                       | 10        |
| 1.2      | Build System . . . . .                            | 10        |
| 1.3      | Testing and Continuous Integration . . . . .      | 10        |
| <b>2</b> | <b>Core Module Implementation . . . . .</b>       | <b>10</b> |
| <b>3</b> | <b>File IO Module Implementation . . . . .</b>    | <b>10</b> |
| <b>4</b> | <b>Network IO Module Implementation . . . . .</b> | <b>10</b> |
| <b>5</b> | <b>Processing Module Implementation . . . . .</b> | <b>10</b> |
| <b>6</b> | <b>Python Bindings Implementation . . . . .</b>   | <b>10</b> |

## Introduction

### 1 Project Setup

#### 1.1 Project Structure

#### 1.2 Build System

#### 1.3 Testing and Continuous Integration

### 2 Core Module Implementation

### 3 File IO Module Implementation

### 4 Network IO Module Implementation

### 5 Processing Module Implementation

### 6 Python Bindings Implementation

## Conclusion

---

---

Part IV

## Chapter 4

---

---

---

# Chapter IV

---

## Library Applications and Evaluation

### Summary

|   |   |    |
|---|---|----|
| 1 | Examples of Library Applications . . . . .    | 12 |
| 2 | Parallelization and Multi-threading . . . . . | 12 |
| 3 | Performance Evaluation . . . . .              | 12 |

### Introduction

- 1 Examples of Library Applications
- 2 Parallelization and Multi-threading
- 3 Performance Evaluation

### Conclusion

---

## Conclusion and Perspectives

---

# Bibliography

- [Gre93] George D. Greenwade. “The Comprehensive Tex Archive Network (CTAN)”. In: *TUGBoat* 14.3 (1993), pp. 342–351.
- [Knu] Donald Knuth. *Knuth: Computers and Typesetting*. URL: <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.

---

## Appendix : Miscellaneous remarks

- Un rapport doit toujours être bien numéroté;
- De préférence, ne pas utiliser plus que deux couleurs, ni un caractère fantaisiste;
- Essayer de toujours garder votre rapport sobre et professionnel;
- Ne jamais utiliser de je ni de on, mais toujours le nous (même si tu as tout fait tout seul);
- Si on n'a pas de paragraphe 1.2, ne pas mettre de 1.1;
- TOUJOURS, TOUJOURS faire relire votre rapport à quelqu'un d'autre (de préférence qui n'est pas du domaine) pour vous corriger les fautes d'orthographe et de français;
- Toujours valoriser votre travail : votre contribution doit être bien claire et mise en évidence;
- Dans chaque chapitre, on doit trouver une introduction et une conclusion;
- Ayez toujours un fil conducteur dans votre rapport. Il faut que le lecteur suive un raisonnement bien clair, et trouve la relation entre les différentes parties;
- Il faut toujours que les abréviations soient définies au moins la première fois où elles sont utilisées. Si vous en avez beaucoup, utilisez un glossaire.
- Vous avez tendance, en décrivant l'environnement matériel, à parler de votre ordinateur, sur lequel vous avez développé : ceci est inutile. Dans cette partie, on ne cite que le matériel qui a une influence sur votre application. Que vous l'ayez développé sur Windows Vista ou sur Ubuntu n'a aucune importance;
- Ne jamais mettre de titres en fin de page;
- Essayer toujours d'utiliser des termes français, et éviter l'anglicisme. Si certains termes sont plus connus en anglais, donner leur équivalent en français la première fois que vous les utilisez, puis utilisez le mot anglais, mais en italique;
- Éviter les phrases trop longues : clair et concis, c'est la règle générale !

## APPENDIX : MISCELLANEOUS REMARKS

---

Rappelez vous que votre rapport est le visage de votre travail : un mauvais rapport peut éclipser de l'excellent travail. Alors prêtez-y l'attention nécessaire.





## APPENDIX : MISCELLANEOUS REMARKS

---