

# Long Time Process Data Visualization

U조 김민수 조용환

2016. 10. 28

## Contents

|           |  |          |
|-----------|--|----------|
| <b>1</b>  | <b>Abstract &amp; Motivation</b>               | <b>2</b> |
| <b>2</b>  | <b>Introduction</b>                            | <b>2</b> |
| 2.1       | 과제목표 . . . . .                                 | 2        |
| 2.2       | 보고서의 구성 . . . . .                              | 2        |
| <b>3</b>  | <b>Background Study</b>                        | <b>2</b> |
| 3.1       | Overview . . . . .                             | 2        |
| 3.2       | Windows Presentation Foundation . . . . .      | 2        |
| 3.3       | Data Reduction Algorithms . . . . .            | 3        |
| 3.4       | Database Model . . . . .                       | 3        |
| <b>4</b>  | <b>Goal &amp; Requirements</b>                 | <b>3</b> |
| 4.1       | 프로젝트 개발 환경 및 서비스 환경 . . . . .                  | 3        |
| <b>5</b>  | <b>Project Architecture</b>                    | <b>3</b> |
| <b>6</b>  | <b>Implementation Sepc</b>                     | <b>4</b> |
| <b>7</b>  | <b>Current Status</b>                          | <b>4</b> |
| <b>8</b>  | <b>Future Work</b>                             | <b>4</b> |
| <b>9</b>  | <b>Division &amp; Assignment of Work</b>       | <b>4</b> |
| <b>10</b> | <b>Schedule</b>                                | <b>4</b> |
| <b>11</b> | <b>Demo Plan</b>                               | <b>4</b> |
| <b>12</b> | <b>[Appendix] Detailed Implementation Spec</b> | <b>4</b> |

# 1 Abstract & Motivation

## 2 Introduction

### 2.1 과제목표

### 2.2 보고서의 구성

## 3 Background Study

### 3.1 Overview

본 project는 서버와 클라이언트 두 개의 application으로 나뉘어있고, 이것은 모두 Windows환경에서 실행된다. 따라서 두 application 모두 .NET Framework를 통하여 작성되었으며, 이 중에서도 Windows Presentation Foundation(WPF) 기반으로 작성되었다. 따라서 WPF framework의 구조와 이에 따른 프로그래밍 패턴에 대한 이해가 필요하다. 또한 project에서 가장 중심이 되는 visualization 성능을 향상시키기 위해서 사용되는 algorithm들을 조사하고 사용될만한 것들을 추려보았다.

### 3.2 Windows Presentation Foundtation

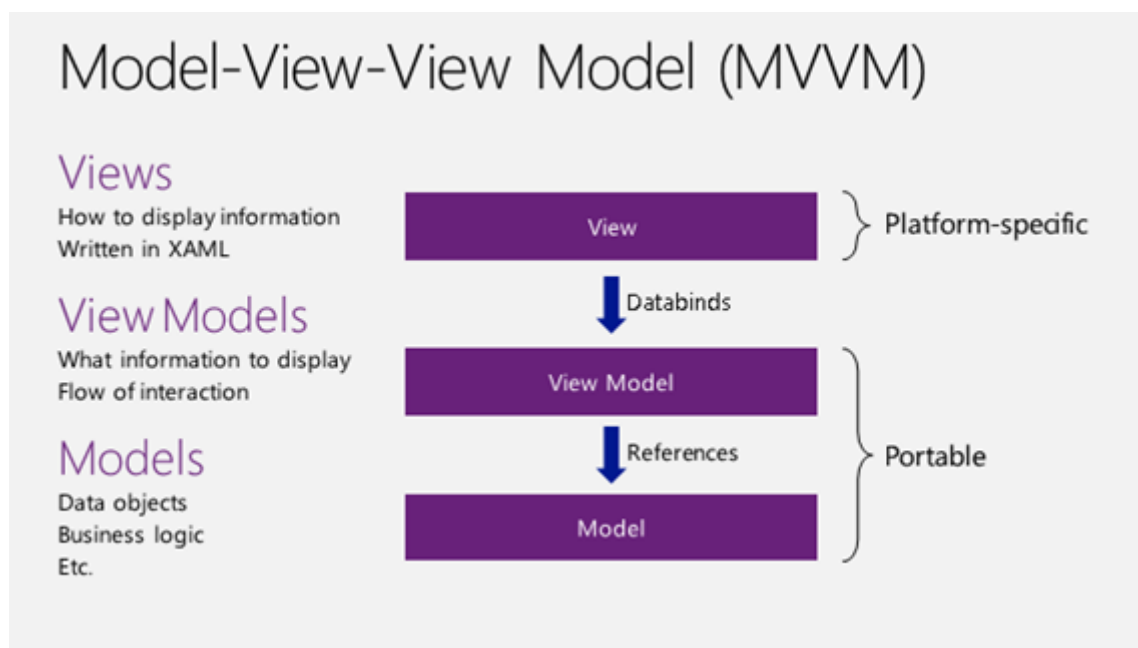


Figure 1: MVVM Pattern

WPF는 .NET framework에서 사용되는 최신의 GUI framework이다. 기존의 WinForms framework와 달리 design(XAML)과 내부 logic(C#)을 구분하여 독립적으로 작성하기 용이하고, 유지관리를 편리하게 할 수 있다. WPF를 사용하는 데에는 MVVM(Model-View-ViewModel) 패턴을 이해하는 것이 필수적이다. MVVM이란 내부 데이터와 로직을 결정하는 Model과, 유저 인터페이스를 디자인하고 이것의 상호관계를 서술하는 View, 그리고 이 사이에서 데이터와 인터페이스를 매핑하는 방법을

서술하는 ViewModel단으로 분리하여 작성하는 패턴이다. ViewModel은 View에 대해 독립적으로 작성되며, 인터페이스와 데이터 간 바인딩을 통해서 표현방식을 지정해줄 수 있다.

### 3.3 Data Reduction Algorithms

### 3.4 Database Model

## 4 Goal & Requirements

### 4.1 프로젝트 개발 환경 및 서비스 환경

## 5 Project Architecture

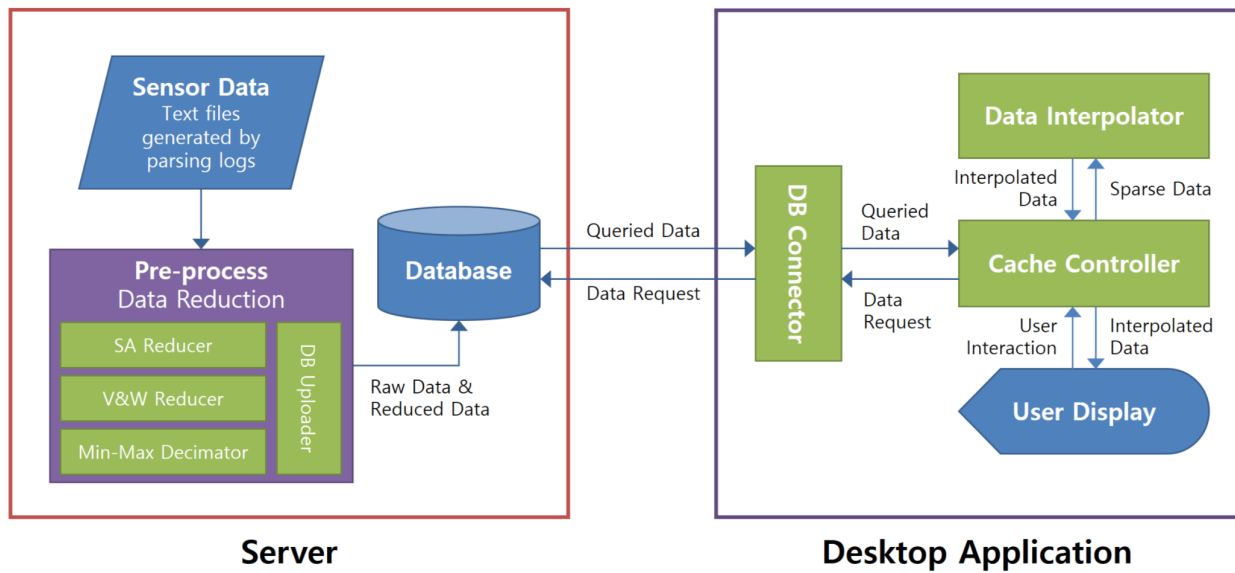


Figure 2: Project architecture

- 
- 6    **Implementation Sepc**
  - 7    **Current Status**
  - 8    **Future Work**
  - 9    **Division & Assignment of Work**
  - 10   **Schedule**
  - 11   **Demo Plan**
  - 12   **[Appendix] Detailed Implementation Spec**