|  |
| --- |
| ADL FRAMEWORK |
| 구조설계서 |
|  |
|  |
| **2021-03-15** |
| **홍덕기(duki.hong@samsung.com)** |

이 문서는 **ADL FRAMEWORK** 개발을 위한 구조설계서이다.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 2021-03-15 | <<홍덕기>> | 초기 문서 생성 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[1. 시스템 개요 3](#_Toc479150191)

[2. 요구사항 4](#_Toc479150192)

[2.1. 기능적 요구사항 4](#_Toc479150193)

[2.2. 비기능적 요구사항 4](#_Toc479150194)

[2.3. 품질 속성 5](#_Toc479150195)

[3. 시스템 구조 6](#_Toc479150196)

[4. 컴포넌트 사양 7](#_Toc479150197)

[부록 8](#_Toc479150198)

[A. 도메인 모델 9](#_Toc479150199)

[B. 품질 시나리오 10](#_Toc479150200)

[C. 품질 시나리오 분석 11](#_Toc479150201)

[D. 후보 구조 12](#_Toc479150202)

[E. 후보 구조 평가 13](#_Toc479150203)

[F. 최종 구조 설계 14](#_Toc479150204)

[G. 최종 구조 평가(ATAM) 15](#_Toc479150205)

# 시스템 개요

## 비즈니스 환경

현대 거의 모든 산업 분야에서 IT가 차지하는 비중이 나날이 높아지고 있다. 이중 핵심이라 할 수 있는 S/W에 관심이 높아지고 있으며, 이런 S/W의 Quality에 큰 영향을 끼치는 S/W Architecture의 중요성이 대두되고 있다.

하지만 S/W Architecture 설계를 위한 Common Tool이 존재하지 않고, 몇 안되는 S/W(그림1. Acme Studio) 들도 그 사용법이 까다로워서 Architect들은 PPT나 종이에 직접 그리는 것을 보다 선호하는 상황이다.

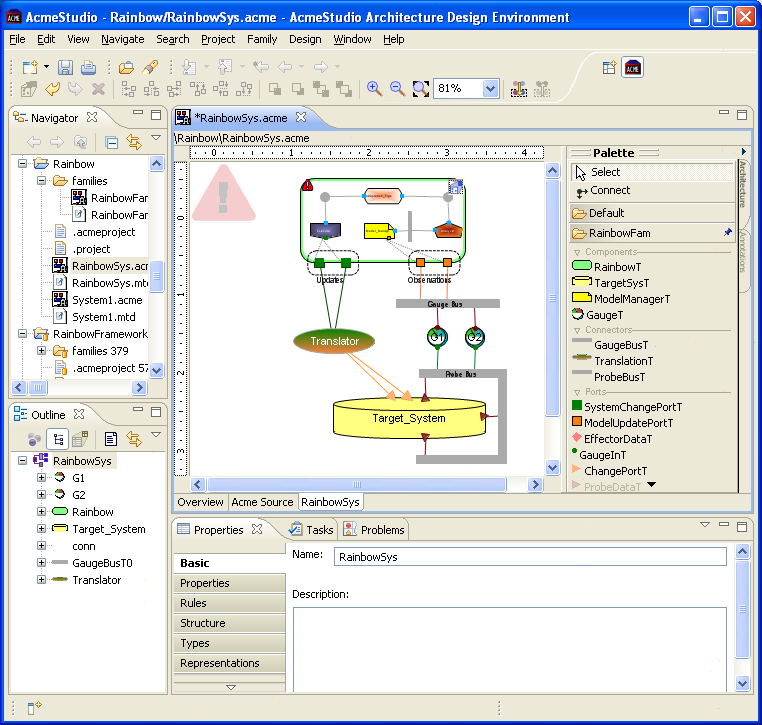


그림 1. ADL 도구 예제(AcmeStudio)

이에 따라, System에 대한 기능적, 비기능적 품질 요구사항을 정의하고 이에 대한 다양한 Solution중 최적 설계 결과를 낼 수 있는 Architecture를 결정했을 때 이를 보다 손쉽게 실행과 개발 관점에서 도식화해서 나타낼 수 있는 도구를 개발하고자 한다.

## 시스템의 정의

ADL(Architecture description Language)은 소프트웨어 구조를 명세하기 위한 언어 또는 개념적인 모델이다. 본 시스템은 ADL을 사용해서 구조를 설계하고 검토하고 분석할 수 있으며 정의는 아래와 같다.



아직까지 소프트웨어 구조를 명세하기 위한 표준 모델링 언어는 존재하지 않는다. 이 과제 에서는 소프트웨어 구조를 명세하기 위한 ADL을 정의하고, 정의된 ADL을 사용해서 소프트웨어 구조를 설계하기 위한 설계 도구를 개발한다.

# 요구사항

## 기능적 요구사항

ADL Framework의 Usecase는 다음과 같다.



ADL Framework은 새로운 Project를 생성하고(UC\_01) 해당 Project에 시스템의 구조를 명세하기 위한 Component & Connector View(UC\_02), 각 Component와 Connector를 어떻게 컴퓨팅 환경에 배치/할당하는지를 명세하기 위한 Deployment View(UC\_03), 개발해야 할 모듈의 관계를 명세하기 위한 Module View(UC\_04) 및 누가 개발/담당할 지를 할당하는 Work Assignment View(UC\_05)를 추가할 수 있다. 또한 생성된 View를 바탕으로 Skeleton Code를 생성(UC\_06) 및 성능 분석(UC\_07)을 할 수 있다. Project는 저장 및 불러올 수 있다.

|  |  |
| --- | --- |
| **UC\_01** | **Project를 생성한다.** |
| 설명 | 각종 View를 포함할 수 있는 새로운 Project를 생성한다. |
| 행위자 | 사용자 |
| 선행조건 | ADL Framework 구동상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 Project 메뉴를 연다. 2. 사용자가 Project 생성 메뉴를 선택한다. 3. System은 Project Name을 설정할 수 있는 화면을 표시한다. 4. 사용자는 Project Name을 입력한다. 5. System은 Project Explorer Window에 새로 생성한 Project를 표시한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_02** | **Project를 저장한다.** |
| 설명 | 각종 View를 포함하는 생성된 Project를 저장한다. |
| 행위자 | 사용자 |
| 선행조건 | Project가 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 Project 메뉴를 연다. 2. 사용자가 Project 저장 메뉴를 선택한다. 3. System은 Project를 Project Name으로 저장한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_03** | **Project를 불러온다.** |
| 설명 | 저장되어 있는 Project를 불러온다. |
| 행위자 | 사용자 |
| 선행조건 | 저장된 Project가 존재하는 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 Project 메뉴를 연다. 2. 사용자가 Project 불러오기 메뉴를 선택한다. 3. System은 Project Load한다.    1. 저장되어 있는 View들을 Load한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_04** | **View를 생성한다.** |
| 설명 | 모든 View의 기본이 되는 View를 생성한다. |
| 행위자 | 사용자 |
| 선행조건 | Project 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 View 메뉴를 연다. 2. 사용자가 Component & Connector View 또는 Deployment View 또는 Module View 또는 Work Assignmeng View 메뉴를 선택한다. 3. System은 View를 Project에 추가한다.    1. System은 그릴 수 있는 Canvas를 생성한다. |
| 추가 동작 | 선택된 View Menu에 맞게 palate menu를 구성한다. |

|  |  |
| --- | --- |
| **UC\_05** | **Component & Connector View를 그린다.** |
| 설명 | System의 구조를 명세하기 위한 Component & Connector View를 그린다. |
| 행위자 | 사용자 |
| 선행조건 | Project 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 View 메뉴를 연다. 2. 사용자가 Component & Connector View 메뉴를 선택한다. 3. System은 View를 Project에 추가한다.    1. System은 그릴 수 있는 Canvas를 생성한다.    2. System은 Palate에 Component와 Connector를 표시한다. 4. 사용자가 Component를 Canvas에 배치한다. 5. System은 배치된 Component를 표시한다. 6. 사용자가 Component간의 관계를 Connector로 연결한다. 7. System은 Component간의 Connector를 표시한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_06** | **Deployment View를 그린다.** |
| 설명 | Component와 Connector를 어떻게 컴퓨팅 환경에 배치/할당하는지를 명세하기 위한 Deployment View를 그린다. |
| 행위자 | 사용자 |
| 선행조건 | Project 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 View 메뉴를 연다. 2. 사용자가 Deployment View 메뉴를 선택한다. 3. System은 View를 Project에 추가한다.    1. System은 그릴 수 있는 Canvas를 생성한다.    2. System은 Palate에 Component와 Connector를 표시한다. 4. 사용자가 Component를 Canvas에 배치한다. 5. System은 배치된 Component를 표시한다. 6. 사용자가 Component간의 관계를 Connector로 연결한다. 7. System은 Component간의 Connector를 표시한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_07** | **Module View를 그린다.** |
| 설명 | 개발해야 할 모듈의 관계를 명세하기 위한 Module View를 그린다. |
| 행위자 | 사용자 |
| 선행조건 | Project 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 View 메뉴를 연다. 2. 사용자가 Module View 메뉴를 선택한다. 3. System은 View를 Project에 추가한다.    1. System은 그릴 수 있는 Canvas를 생성한다.    2. System은 Palate에 Module를 표시한다. 4. 사용자가 Module를 Canvas에 배치한다. 5. System은 배치된 Module를 표시한다. 6. 사용자가 Module간 관계를 연결한다. 7. System은 Module간 관계를 표시한다. |
| 추가 동작 |  |

|  |  |
| --- | --- |
| **UC\_08** | **Work Assignment View를 그린다.** |
| 설명 | 누가 개발/담당할 지를 할당하는 Work Assignment View를 그린다. |
| 행위자 | 사용자 |
| 선행조건 | Project 생성 / 불러온 상태 |
| 후행조건 |  |
| 기본 동작 | 1. 사용자가 View 메뉴를 연다. 2. 사용자가 Work Assignment View 메뉴를 선택한다. 3. System은 View를 Project에 추가한다.    1. System은 그릴 수 있는 Canvas를 생성한다.    2. System은 Palate에 Module를 표시한다. 4. 사용자가 Module를 Canvas에 배치한다. 5. System은 배치된 Module를 표시한다. 6. 사용자가 Module간 관계를 연결한다. 7. System은 Module간 관계를 표시한다. |
| 추가 동작 |  |

## 비기능적 요구사항

// 활동5. 품질 속성 선정

// 점검5-2. 품질 요구사항의 명세가 적절한가?

// 점검5-3. 품질 요구사항의 측정이 가능한가?

// 점검5-4. 비기능적 요구사항의 제약이 명확한가?

|  |  |  |
| --- | --- | --- |
| **NFR\_01** |  |  |
| 설명 |  | |
| 환경 |  | |
| 자극 |  | |
| 반응 |  | |
| 측정 |  | |
| **제약** |  | |

## 품질 속성

// 활동5. 품질 속성 선정

// 점검5-2. 품질 요구사항의 명세가 적절한가?

// 점검5-3. 품질 요구사항의 측정이 가능한가?

|  |  |  |
| --- | --- | --- |
| **QA\_01** |  |  |
| 설명 |  | |
| 환경 |  | |
| 자극 |  | |
| 반응 |  | |
| 측정 |  | |

# 시스템 구조

// 활동8. 구조 명세

// 점검8-1. 프로세스 등의 Allocation이 적절한가?

// 점검8-2. 컴포넌트 측면에서 Grouping이 적절한가?

// 점검8-3. 시스템의 동작 특성에 대한 설명이 적절한가?

# 모듈 사양

// 활동9. 모듈 명세

// 점검9-1. 컴포넌트 명세가 충분한가? (개발 가능)

// 점검9-2. 모듈 측면에서 Grouping이 적절한가?

// 점검9-3. Work Assignment가 적절한가?

// 점검9-4. 개발 측면에서 구조적 특징에 대한 설명이 적절한가?

부록

[A. 도메인 모델 9](#_Toc479150206)

[B. 품질 시나리오 10](#_Toc479150207)

[C. 품질 시나리오 분석 11](#_Toc479150208)

[D. 후보 구조 12](#_Toc479150209)

[E. 후보 구조 평가 13](#_Toc479150210)

[F. 최종 구조 설계 14](#_Toc479150211)

[G. 최종 구조 평가(ATAM) 15](#_Toc479150212)

1. 도메인 모델

// 활동3. 도메인 모델 정립하기

// 점검3-1. 도메인 모델이 충분히 세분화 되었는가?

// 점검3-2. 도메인 모델에 구조가 반영되지 않았는가?

1. 품질 시나리오

// 활동4. 품질 시나리오 생성

// 점검4-1. 구조에 영향을 미치는 품질에 대한 검토가 충분한가?

// 점검4-2. 성능에 대한 검토가 충분한가?

// 점검4-3. 변경용이성/확장성에 대한 검토가 충분한가?

1. 품질 시나리오 분석

// 활동5. 품질 속성 선정

// 점검5-1. 품질 시나리오 분석이 적절한가? (근거)

1. 후보 구조

// 활동6. 후보 구조 설계

// 점검6-1. 품질에 대한 분석과 후보 구조가 적절한가?

// 점검6-2. 성능에 대한 분석과 후보 구조가 적절한가?

// 점검6-3. 변경용이성/확장성에 대한 분석과 후보 구조가 적절한가?

1. 후보 구조 평가

// 활동7. 최종 구조 설계

// 점검7-1. 충돌되는 후보 구조의 비교/분석이 적절한가? (근거)

// 점검7-2. 선정된 후보 구조의 단점/RISK 분석이 명확한가?

1. 최종 구조 설계

// 활동7. 최종 구조 설계

// 점검7-3. 최종 구조로의 통합 과정에 대한 설명이 적절한가?

// 점검7-4. 최종 구조의 단점/RISK 관리가 적절한가? (개선)

1. 최종 구조 평가(ATAM)

// 활동10. 최종 구조 평가

// 점검10-1. 구조에 영향을 미치는 품질에 대한 검토가 충분한가?

// 점검10-2. 설계 결정사항의 식별이 충분한가?

// 점검10-3. 설계 결정사항의 분석이 적절한가? (근거)

// 점검10-4. 최종 구조의 평가가 적절한가? (위험)