IEEE Recommended Practice for Software Design Descriptions

학번 : 200614164
이름 : 김효석

INDEX

1. Scope

2. Consideration for Producing an SDD

3. Design description information content

4. Design description organization

1. Scope

소프트웨어 디자인들을 설명하기 위한 권장 방식이며, 필수적인 정보와 소프트웨어 디자인 문서(SDD)를 위한 권장 구성을 나타낸다.

디지털 컴퓨터에서 수행되는 상업적, 과학적, 군사적 소프트웨어에 적용되며, 소프트웨어의 크기, 복잡성 또는 위험한 상황에 의해 제한되지 않는다.

디자인을 위한 특별한 방법, 구성 관리 또는 품질보증에 제한되지 않는다. 문서, 자동화 데이터 베이스, 디자인 문서 언어 또는 문서의 다른 방법으로 적용된다.

2. Consideration for Producing an SDD

SDD를 생산하기 전에 고려되어지는 정보를 제공한다. SDD가 어떻게 소프트웨어 생명 주기에 일치되는지, SDD가 어디에 적합한지, 왜 사용하는지에 대해 기술 되어있다.

Software life cycle
SDD within the life cycle
Purpose of an SDD

2. Consideration for Producing an SDD (cont)

Software life cycle

소프트웨어 시스템의 생명 주기는 일반적으로 소프트웨어 제품이 구성될 때 시작하며, 제품이 더 이상 사용되지 않을 때 끝나는 시간의 주기로써 정의 된다. 생명 주기 접근방법은 효과적인 엔지니어링 관리 도구이며, SDD의 준비 및 사용을 설명하는 배경을 위한 모델을 제공한다. 전형적인 주기는 SDD를 위한 배경을 정의하기 위해 사용되며, IEE Std 610.12-1990에 기초하고, 개념 단계, 요구사항 단계, 디자인 단계 등의 단계로 구성되어있다.

2. Consideration for Producing an SDD (cont)

SDD within the life cycle

새로운 소프트웨어 시스템과 유지 하에 존재하는 시스템에 따라, 소프트웨어 시스템을 위해 사용되는 디자인과 수행은 그 시스템에서 이끌어진 요구사항을 만족시키는 것이 중요하다.

SDD는 디자인 단계 동안 수행되는 디자인 프로세스의 결과를 기록한다.

2. Consideration for Producing an SDD (cont)

Purpose of an SDD

SDD는 소프트웨어 시스템이 소프트웨어 요구 명세서 IEE Std 830-1998에서 확인되는 요구사항을 만족하기 위해 어떻게 설계되어야 하는가를 알려준다.

본질적으로, SDD는 수행 활동을 위한 세밀한 설계도가 된다.

완벽한 SDD에서, 각각의 요구사항은 하나 이상의 디자인 엔티티들을 추적할 수 있다.

3. Design description information content

Introduction

SDD는 만들어지기 위한 소프트웨어 시스템의 표현 또는 모델이며, 소프트웨어 시스템의 계획, 분석, 수행을 위해 필요한 정확한 디자인 정보를 제공한다. 시스템을 디자인 엔티티로 분할함을 나타내며, 엔티티들 사이에 중요한 속성과 관계를 기술한다.

디자인 문서는 속성들이 모든 엔티티들을 위해 특성화 될 때 비로소 완성된다.

3. Design description information content (cont)

Design entities

디자인 엔티티는 다른 요소들로부터 구조적, 기술적으로 구별되며, 개별적으로 이름 지어지고 관계 맺어지는 디자인의 요소이다.

디자인 엔티티는 소프트웨어 시스템 요구사항에 대한 분별 작업의 결과로서 일어난다.

엔티티는 시스템, 하위 시스템, 데이터 저장소, 모듈, 프로그램, 프로세스들로 존재한다.

엔티티들은 본질적으로 다를지라도, 공통적인 특성을 가지고 있으며, 디자인 엔티티 속성들에 의해 기술된다.

3. Design description information content (cont)

Design entity attributes

디자인 엔티티 속성은 고유의 특성이며, 엔티티에 대한 사실적인 성명서를 제공한다.

디자인 엔티티 속성은 디자인 엔티티에 대한 질문으로 생각되어 질 수 있으며, 이러한 질문들의 모임은 완전한 엔티티 문서를 제공한다.

디자인 엔티티 속성의 리스트는 모든 SDD들을 위해 요구되는 최소한의 집합이며, 이 속성들의 선택은 3가지 기준에 기초를 둔다.

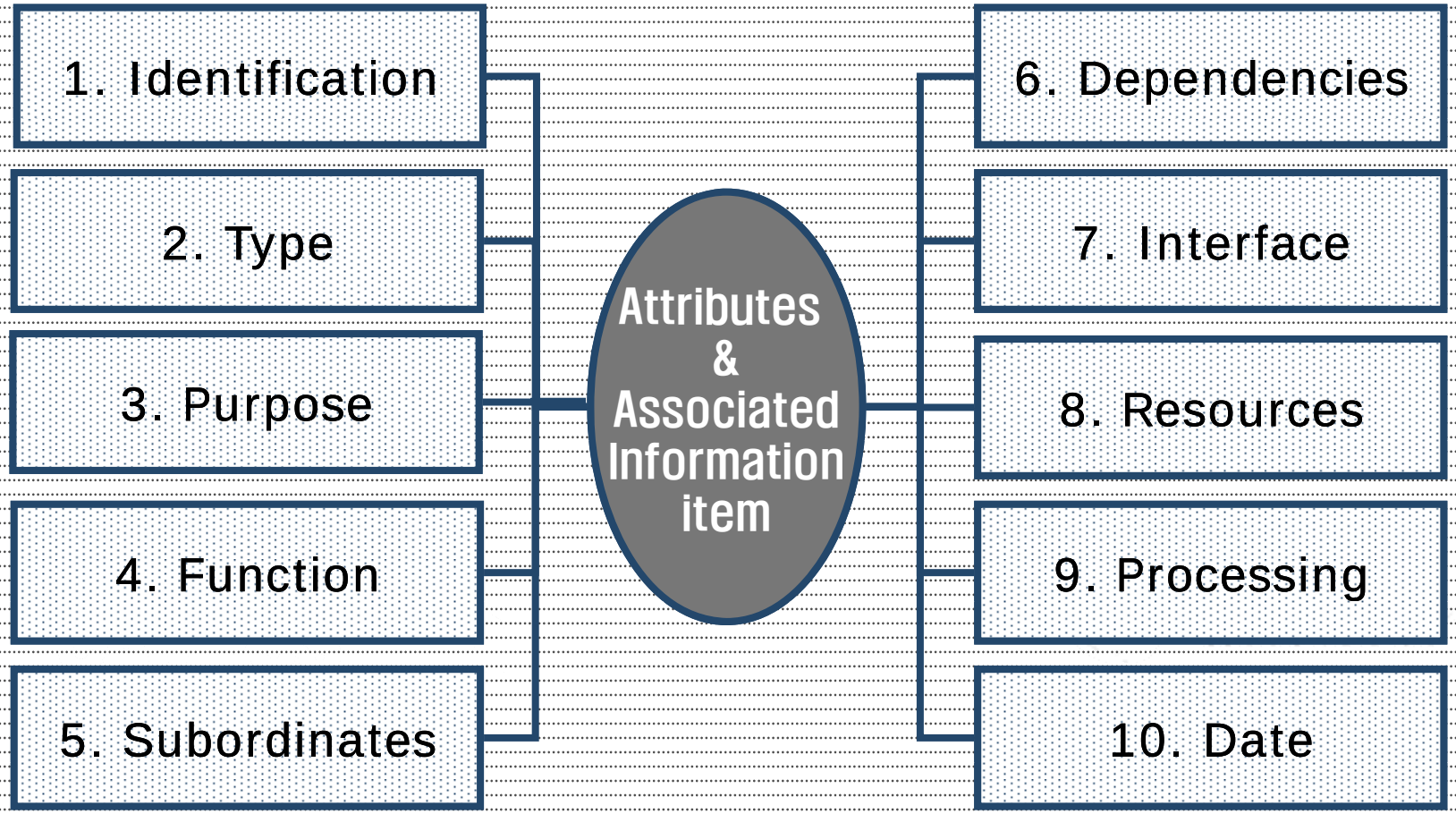
3. Design description information content (cont)

Design entity attributes (cont)

(a) 디자인 엔티티 속성은 모든 소프트웨어 프로젝트에 필수적이다.
(b) 속성값의 잘못된 명세는 개발되어진 소프트웨어 시스템의 실패가 된다.
(c) 속성은 본질적인 디자인 정보를 설명하는 것이지, 디자인 프로세스에 관련된 정보를 설명하는 것이 아니다.

3. Design description information content (cont)

Design entity attributes (cont)



4. Design description organization

Introduction

각각의 디자인 문서 사용자는 소프트웨어 디자인에 필수적이라 생각되는 것에 대한 다른 관점을 가지고 있다. 이러한 정보들은 문서로부터 인용하기가 어렵거나 비현실적이고, 받아들이기가 어렵다.

디자인 속성 정보를 구체화하도록 도와주는 관점의 인식을 소개한다.

디자인 엔티티의 권장 조직과 그들의 관련 속성들이 다양한 기술적인 관점으로부터 디자인 정보의 접근을 용이하게 한다. 이 권장 조직은 적응력이 좋으며, 문서, 디자인 언어 또는 데이터베이스 관리 시스템 등과 같은 미디어에서 사용된다.

4. Design description organization (cont)

Design views

디자인 뷰는 소프트웨어 프로젝트 활동의 필요한 것들에 보다 명확하게 만족되는 디자인 엔티티 속성 정보의 하위 집합이다.

각각의 디자인 뷰는 소프트웨어 시스템에 대한 별개의 관계를 나타내며, 정보접근과 통합을 단순화하는 간결하고 유용한 형식의 디자인에 대한 문서를 제공한다.

4. Design description organization (cont)

Design view	Scope	Entity attribute	Example representations
Decomposition description	디자인 엔티티로 시스템의 구분	확인, 종류, 목적, 기능, 종속	계층 분석 다이어그램, 자연어
Dependency description	엔티티와 시스템 자원 사이에 관계에 대한 문서	확인, 타입, 목적, 의존성, 자원	구조 차트, 데이터 플로우 다이어그램, 처리 다이어그램
Interface description	디자이너, 프로그래머 또는 테스터가 시스템을 구성하는 디자인 엔티티를 사용하기 위해 필요한 것들의 목록	확인, 기능, 인터페이스	Interface files. Parameter table
Detailed description	엔티티의 내부 디자인 세부사항에 대한 문서	확인, 처리, 데이터	플로우 차트, N-S chart, PDL

THANK YOU

4. Design description organization (cont)

1) Decomposition description

ⓐ Scope

디자인 엔티티로 소프트웨어 시스템의 구분을 기록한다. 시스템을 구축하는 방법과 각각의 엔티티의 목적과 기능을 기술한다.

확인, 타입, 목적, 기능, 종속성을 위한 속성 문서들이 포함된다.

4. Design description organization (cont)

1) Decomposition description

ⓑ Use

목적에 적합한 주요 디자인 엔티티의 시스템을 사용하려는 디자이너나 관리자들에 의해 사용된다.

Decomposition description의 정보는 계획, 모니터링, 소프트웨어 프로젝트의 제어를 위한 프로젝트 관리에 의해 사용된다. 또한 각각의 소프트웨어 컴포넌트와 그것의 목적과 기본적인 기능을 확인 할 수 있다. 다른 프로젝트 정보와 함께 개발 노력을 위한 비용, 직원, 스케쥴을 평가할 때 사용된다.

4. Design description organization (cont)

1) Decomposition description

Ⓒ Representation

계층 분석 다이어그램을 주로 사용한다. 이 다이어그램은 각각의 엔티티를 위한 목적과 기능의 기본적인 언어 문서들과 함께 사용된다

4. Design description organization (cont)

2) Dependency description

ⓐ Scope

엔티티들 사이에 관계들을 열거한다. 종속 엔티티들을 확인하고, 그들의 짝을 기술하며, 필수적인 자원을 확인한다.

디자인 엔티티들 사이에 상호작용을 위한 전략을 정의하고, 어떻게, 왜, 어디서, 어떤 레벨 시스템 작용이 발생하는가를 쉽게 인식하기 위한 필수 정보를 제공하며, 공유 정보, 실행의 규정된 순서, 또는 잘 정의된 파라메터 인터페이스와 같은 엔티티들 사이에 존재하는 관계들의 종류를 열거한다.

확인, 종류, 목적, 독립성, 자원을 위한 속성 문서들이 포함된다.

4. Design description organization (cont)

2) Dependency description

ⓑ Use

요구사항과 디자인 변경에 대한 영향을 평가하기 위해 시스템이 어떻게 동작하는가에 대한 전반적인 배경을 제공한다. 시스템 실패 또는 자원 병목현상을 야기하는 엔티티를 분리를 위해 관리자에게 도움을 준다. 다른 엔티티에 의해 필요되어지고 첫번째로 개발되어야하는 엔티티를 확인하므로써 시스템 통합 계획을 제시할 때 도와준다. 또한 통합 테스트 경우의 생산을 돕기 위한 통합 테스팅에 의해 사용된다.

4. Design description organization (cont)

2) Dependency description

Ⓒ Representation

일반적인 명세서 언어는 시스템 기능과 데이터, 그들의 상호 관계와 I/O 등의 명세서를 제공한다.

디자인 엔티티들 사이의 관계는 데이터 플로우 다이어그램, 구조 차트, 처리 다이어그램에 의해 나타내진다.

4. Design description organization (cont)

3) Interface description

ⓐ Scope

디자이너, 프로그래머, 테스터들이 엔티티에 의해 제공되는 기능들을 올바르게 사용하기 위해 알아야 하는 모든 것을 제공한다. 소프트웨어 요구사항 명세서에 제공되어 있지 않은 외부 & 내부 인터페이스의 세부사항을 포함한다.

각각의 엔티티를 위한 일련의 인터페이스 명세서들로 이루어져 있다. 확인, 기능, 인터페이스를 위한 속성 문서들이 포함된다.

4. Design description organization (cont)

3) Interface description

ⓑ Use

디자이너, 프로그래머, 구매자 그리고 테스터들 사이에 의무적인 계약을 할 때 사용된다. 엔트티들의 세부 디자인을 하기 전에 필요한 동의를 제공한다. 추가적으로, 인터페이스 문서는 소비자 문서를 제공하기 위한 기술적인 저자에 의해 사용되거나 소비자에 의해 직접적으로 사용된다. 마지막으로, 휴먼 인터페이스 뷰의 생산물이 된다.

디자이너, 프로그래머, 테스터들 사이에 협동 엔티티들을 어떻게 상호작용 할 지에 대한 계약을 확립한다. 다른 디자이너 또는 프로그래머가 그 엔티티가 상호작용하는 소프트웨어를 개발하기 위해 알아야하는 것을 포함한다.

4. Design description organization (cont)

3) Interface description

Ⓒ Representation

화면 형식, 유효한 입력, 결과 출력을 포함하는 각각의 엔티티들과의 통신을 위한 언어를 제공해야만 한다.

사용자에게 매우 명백하고, 소비자가 시스템을 어떻게 이해하는지에 대한 세부사항을 포함하는 엔티티들은 기능적 모델, 사용을 위한 시나리오, 세부적인 특징 집합들 그리고 상호작용 언어를 포함해야 한다.

4. Design description organization (cont)

4) Detailed design description

ⓐ Scope

각각의 디자인 엔티티의 내부적인 세부사항을 포함한다. 확인, 처리 데이터를 위한 속성 문서들이 포함된다.

4. Design description organization (cont)

4) Detailed design description

ⓑ Use

수행 이전에 프로그래머에 의해 필요한 세부사항을 포함한다. 또한 단일 테스트 계획을 산출할 때 도움을 주기 위해 사용될 수 있다.

4. Design description organization (cont)

4) Detailed design description

Ⓒ Representation

메타코드 또는 구조적 영어의 사용 또는 N-S 차트 나 플로우 차트와 같은 그래픽컬한 방법들을 포함한다.