Traceability

200310548 이정우
200911364 곽수미
200911372 김민하
200911373 김바울

Abstract

- Introduction
- System Life Cycle for Traceability
- Need for Traceability
- Problems and Issues Concerning Traceability
- Definition of Terms
- State If the Practice of Traceability
- Contemporary Traceability Practices
- An Ideal Process for Traceability
- Actual Practice for Implementing Traceability
- Return on Investment for Traceability
- Current Traceability Tools
- Common Tool Characteristics
- COMMERCIL CASE Tools for Traceability
- Future Trends and Conclusions

Introduction

- 성공적인 system development란?
  - Stakeholder의 needs와 requirements를 시스템에 반영하는 것

- Traceability란?
  - Traceability는 system requirements, design, code, test and implementation 관계를 이해하는데 필수적인 도움을 줌

  - 시스템에 대한 Stakeholder의 needs와 requirement를 만족시키기 위한 수단

System Life Cycle for Traceability Management

- life cycle
  - 일반적으로 목표한 시스템에 의해 결정됨
- life cycle models
  - spiral model, the evolutionary model, and the prototyping model

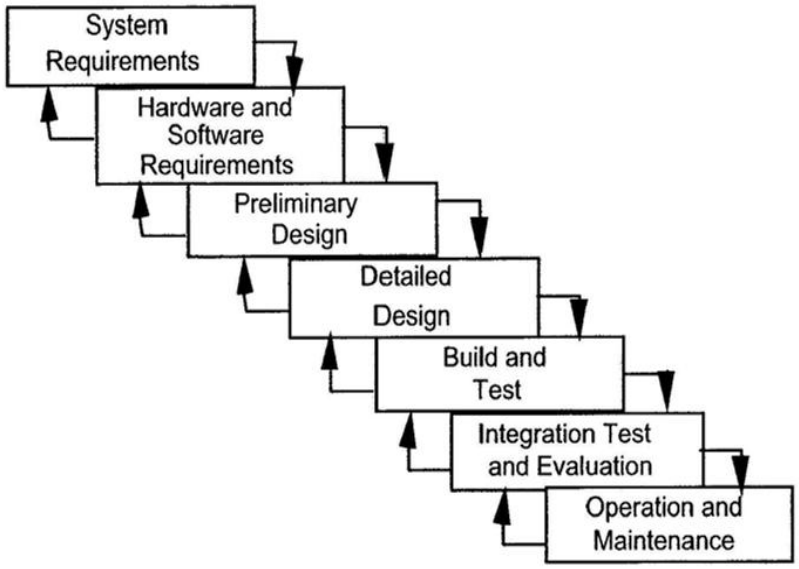

Figure 1. Typical system and software development life cycle.

System Life Cycle for Traceability Management

- Typical system and software development life cycle
  - system requirements
    - 보통 요구자가 natural language로 준비하여 개발자에게 제공함
  - Hardware and Software Requirements
  - Preliminary Design
  - Detailed Design
  - Build and Test
  - Integration Test and Evaluation
  - Operation and Maintenance

Need for Traceability

- Traceability의 필요성
  - verification과 validation
  - System 개발에 사용된 processes의 이해
  - quick access to information
  - 시스템 개발에 사용된 기술의 시각화
  - change control
  - development process control
  - risk control

Need for Traceability

- Traceability가 제공하는 것
  - Insight to nonbehavioral components such as
  - Quality, Consistency, Completeness, Impact analysis, System evolution and Process improvement
  - High-level의 동작의 영향을 자세하게 평가
  - Conflict detection

Problems and Issues Concerning Traceability

- 추적에 관한 PROBLEMS
  - 수동으로 요구 사항 문서에 요소를 추가해야 함
  - Traceability의 이득은 나중까지 확인되지 않음
  - Traceability는 수시로 잘못 이해되고, 자주 오용되고, 드물게 의도대로 실행 됨
  - 프로젝트의 복합성 문제
    - 항공, 통신, 네비게이션, 보안, 안전 등 각 학문마다 각각의 특성이 있기 때문

Problems and Issues Concerning Traceability

- 추적에 관한 ISSUES
  - 현재에는 single modeling method나 language로 효과적으로 크고 복잡한 시스템을 표현할 수 없기 때문에 여러 가지 관련된 모델링 방법이나 언어를 이해해야 함
  - 다른 분야의 toolsets 및 threads 간에 추적을 제공하는 시스템 속성과 사용된 분류 구성표가 필요함
  - 따라서, 검증과 확인을 위해, Traceability는 common denominator에 항상 집중해야 함

Definition of Terms

- Allocation
  - apportionment로 할당하거나 특정 모듈에 기능을 배분
- Audit
  - 소프트웨어 제품의 기능을 확인하기 위한 검사
- Behavior
  - 조건에 반응하여 하는 행위(ex activity, change)
- Bottom-up
  - 위로 이동하면 결정하는 방법
- Classification
  - 그룹화 또는 데이터를 구분하는 행위

Definition of Terms

- Flowdown
  - 이동 또는 상부에서 낮은 수준으로의 순환, 최고 수준에서부터 디자인에 코드까지 요구 사항을 추적
- Function
  - 시스템에서 작업을 수행할 때 다른 작업을 수행하기 위해 알고리즘의 동작의 수식을 제공할 수 있도록 대응
- Hierarchy
  - 계층구조, 계급 또는 순서 대로 분할되거나 분류
- Impact analysis
  - 한개에 하나의 접촉을 시험하거나 구성 요소 부분으로 검사
- Policy
  - Management나 procedure는 자료의 관계에 따라가게 된다.

Definition of Terms

- Requirement
  - 시스템이나 시스템 구성 요소가 갖추어야 할 조건
- Thread
  - 프로그램 수행 시 프로세스 내부에 존재하는 수행 경로
- Top-down
  - 위에서 아래로 큰 부분에서 세분화여 가는 방법
- Top-level requirement
  - 최상위 요구 사항, 보안 시스템 수준 요구 사항
- Traceability
  - 어떤 추상화 레벨의 모델 또는 코드가 보다 높은 추상화 레벨의 특정한 다른 모델로부터 기인하였음을 알고 인식할 수 있는 능력

Definition of Terms

- Traceability management
  - 최고 수준에서 처음부터 끝까지 추적의 방향을 지시
- Tree
  - 소프트웨어의 구성 요소

State If the Practice of Traceability

- Traceability 관리는그림 2에서 보이는 것처럼 프로젝트 개시에서 운영과 관리까지 전체 발달 수명주기에 적용함
  - 수동 및 자동 지원의 조합을 사용하여 추적 관리하여 요구자의 요구 사항의 만족의 보증을 제공함

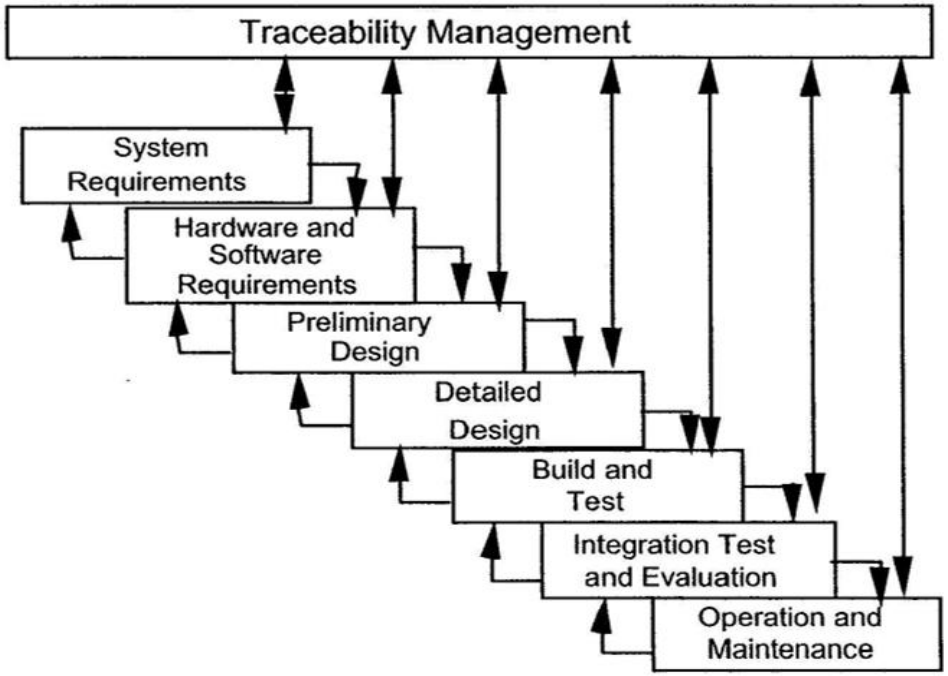

Figure 2. Traceability management across the system development life cycle.

Contemporary Traceability Practices

- Traceability 방법
  - 유일한 인식기를 할당하고 연결하는 방법
    - 고유한 식별 시스템 설정으로 앞으로 또는 뒤로 요구 사항에 따라 제품을 추적할 수 있음

- 복잡한 시스템을 위한 전형적인 Traceability 테이블

Table 1. Traceability matrix for multisegment system

SRD	SS	Segment 1	Segment 2	Segment 3	ICD
3.1.2.1	3.3.4.5	3.2.2.5.6	3.5.3.2		3.1.4.6.7
	3.3.4.6	3.2.2.5.7			3.1.4.6.8
		3.4.5.6.2			3.1.4.6.9
3.4.3.1	3.6.7.2	3.5.2.5.1	3.7.4.3.1	3.6.4.5.2	3.3.2.4.5
	3.8.4.3		3.7.4.3.2		3.3.2.4.7

Contemporary Traceability Practices

  - 이 테이블에서 시스템 요구 사항 문서 (SRD)의 개별 요구 사항은 수동으로 차례로 수동으로 시스템 세그먼트에 특정 규격에 링크되어 시스템 사양에서 더 자세한 시스템 요구 사항에 연결되어있음

- Traceability 테이블에서 대표된 체계는 그림 3에서 것과 같이 형성됨

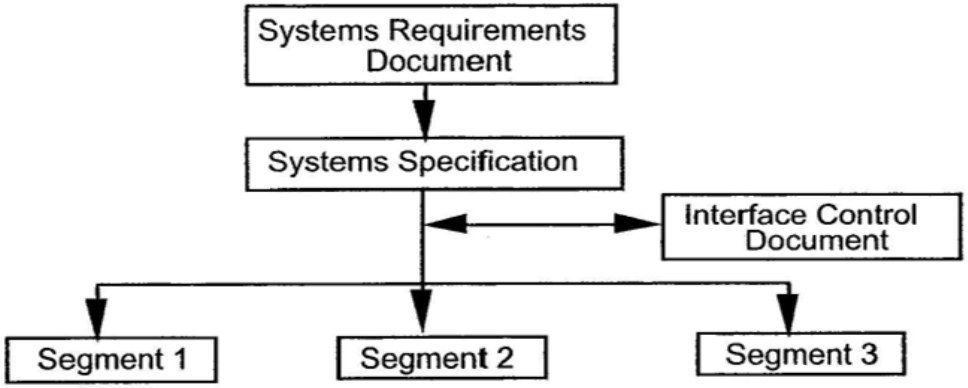

Figure 3. Typical requirements classification schema for a large, complex system.

Contemporary Traceability Practices

- System Requirements Document (SRD)
  - SRD는 관계자의 입력을 나타냄
- Systems Specification
  - 상세한 정보를 설계할 수 있음
- The interface control document (ICD)
  - Segments를 발생하는 모든 메시지에 대해 링크를 제공함

Contemporary Traceability Practices

- 대부분의 시스템 개발 프로그램에서는 필요조건이 추가, 변경, 삭제되기 때문에 시스템에 지속적인 변화가 있음
- 따라서, 늘 변화하는 requirements의 관리는 시스템 requirements가 어떻게 파생되어 흐르는지 제공하기에 아주 중요한 Traceability 기본이 됨

An Ideal Process for Traceability

- 무엇이 추적되어야 하는지 이해해야 할 것
  - 구조적인 관점
  - 분류 체계에서 시스템을 개발하기 위한 정의된 과정(process)
  - 만들어진 제품을 검증하고 명시하기 위해서도 정의된 과정(process)

An Ideal Process for Traceability

- 이상적인 Traceability process 단계
  - Identification
  - Architecture selection
  - Classification
  - Allocation
  - Flow-down

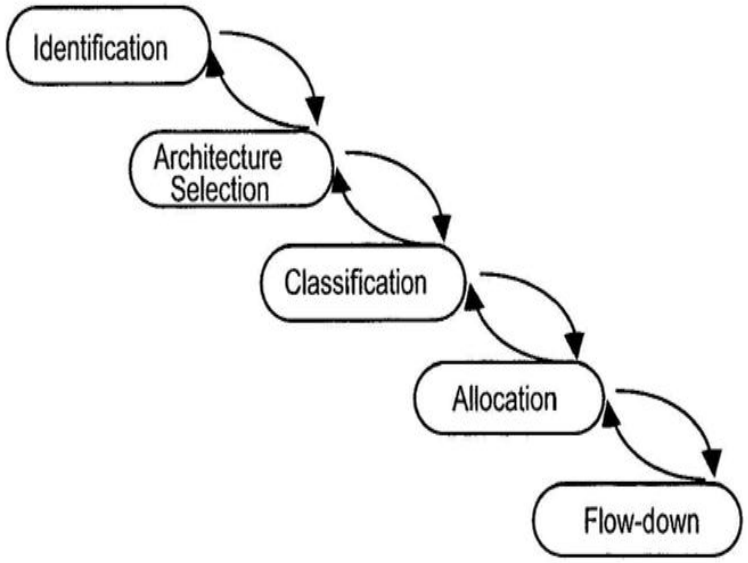

Figure 4. The ideal traceability process.

An Ideal Process for Traceability

- 이러한 top-down 접근은 큰 규모와 복잡한 프로젝트의 추적 관리(management of Traceability)에 가장 효과적이라고 입증됨
- 그러나, 이러한 접근은 기본적으로 상당한 시간의 투자와 능력 있는 인력을 요구하는 활동임

Actual Practice for Implementing Traceability

- 실제 Traceability
  - 추적은 매우 힘든 작업임
- 도메인 전문가들은 그림 3과 유사한 break-down 과정(process)을 따름

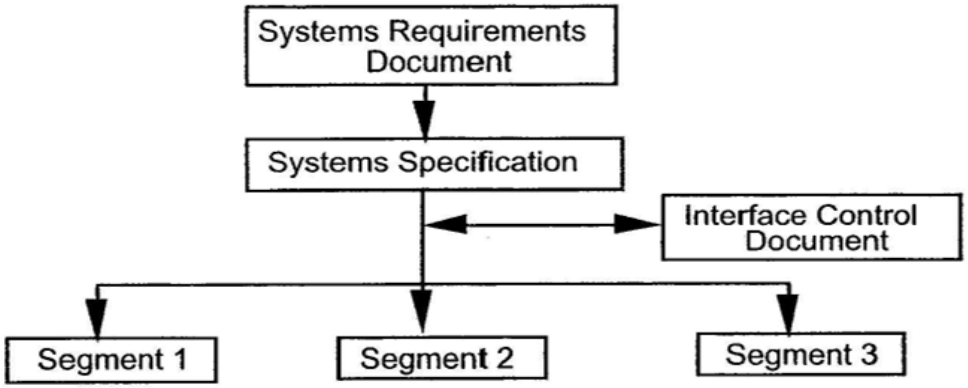

Figure 3. Typical requirements classification schema for a large, complex system.

Actual Practice for Implementing Traceability

- 적절한 시스템 구조가 식별되면, 분류 스키마 또는 구체적인 할당 요구의 목적을 가진 스키마가 준비되고 그러한 요구는 특정 단위에 할당됨
- 분류 스키마가 사용된 유형의 예
  - 프로젝트의 기능적 측면에 집중
  - 수행(performance)과 안전(security)에 집중
  - 이해 당사자 조직(stakeholder organization)에 집중

- 추적(Traceability)은 관점의 변화 및 요구사항의 변화로 인해 지속적인 과정(process)이 됨
- 이러한 다양한 견해를 확인하기 위해, 시스템 요구에 관한 추적 연계(trace linkage) 형태로 부터의 공통적인 basis가 있음

Actual Practice for Implementing Traceability

- 구문과 의미론적 정보는 추적(tracing)을 수행하기 위하여 필요함
  - 언어 의미론(Language semantics)은 요구사항의 맥락과 의미와 관련된 trace를 확인(보장)하기 위하여 필요함
  - 구문론은 특정한 단어와 문장(구), 문맥, 의미와 상관없이 추적하기 위하여 필요함
- 두 개를 통합하는 것으로 전체적인 Traceability가 제공될 수 있도록함

Actual Practice for Implementing Traceability

- 분류 스키마에 따른 할당
  - 사용가능한 CASE tool중 하나로부터 자동적인 지원을 받음
    - 이러한 결과로부터 Traceability matrices가 발생됨
    - CASE tool이 돕는 것에 따라 matrices가 자동적으로 준비되거나 직접 준비해야 함

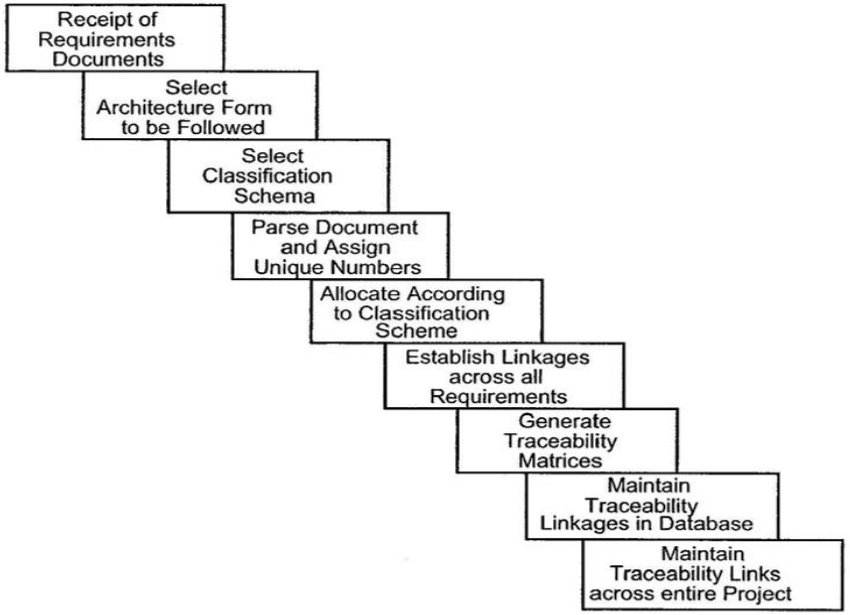

Figure 5. Steps to accomplish traceability.

Return on Investment for Traceability

- Traceability에 대한 투자 수익을 측정하는 것은 불가능함
  - 문서화 될 수 있는 실행에 관한 대부분의 비용과 이익은 비교 사례를 연구하지 않는 경우엔 확인하기가 매우 어렵움
- 실행 비용에 포함되는 것
  - 전문가의 노력
  - CASE tool의 초기비용
  - CASE tool의 유지비용
  - 수동적인 접근에서는
    - 구조적인 관점, 분류 스키마, 할당, 연계(linkage), 시스템 유지, 수리비용, 관리 등

Current Traceability Tools

- 현재의 Traceability Tools
  - Hypertext linking
  - unique identifier
  - 통사론적 유사 계수
  - 이것들의 조합

Current Traceability Tools

- Hypertext linking
  - 다른 요구사항에 연결되기 위한 "hotword" 또는 단어/구문은 수동으로 확인되고 하이퍼텍스트 툴에 입력됨
  - 링크는 자동적으로 Traceability를 제공하기 위한 툴에 의해 만들어지고 유지됨
- unique identifier
  - 프로젝트의 전체에 걸쳐 개별적 요구사항에 식별자가 할당됨
- 통사론적 유사 계수
  - 주어진 요구사항의 단어의 미리 정의된 세트가 또 다른 요구사항에서 발견되는지 아닌지 확정함
  - 유사성의 도가 상기에 미리 정의된 임계일 때, 2가지 요구사항을 추적함

Current Traceability Tools

- 각각이 가진 문제
  - 그들은 추적(tracing)이 발생하는 의미론 또는 문맥을 고려하지 않음
    - Hypertext linking
      - 단어가 이용되는 방법이나 배치에 배치에 관계없이 탐지된 텍스트를 찾음
    - unique identifier
      - 의미 또는 문맥에 관한 어떤 원근법으로, 확인되지 않는 그 요구사항에게 단지 액세스를 제공함
    - 통사론적 유사 계수
      - 추적되는 것은 요구사항의 의미와 문맥에 관해 무차별적이라는 점에서 연결되는 Hypertext linking과 같음

Common Tool Characteristics

- Traceability를 위해 최소한으로 생각된 약간의 공통 tool 특성이 있음
  - 사용자가 잘 이해 이해되야 함
  - 개발자가 이용하는 개발 환경의 특성과 일치함
  - 제공된 데이터를 받아들이고 이용해야 함
  - 능동 및 수동으로 다양한 활동과 서비스를 지원해야함

- 인간 의사결정이 분류 스키마와 시스템 구조 명칭에 필수적인 것처럼, Traceability 툴은 결코 완전히 자동화되지 않을 것임

COMMERCIAL CASE Tools for Traceability

- 일부 상용 도구는 하나의 단계에서 하나의 규칙으로 표현된 Traceability link information를 위해 개발되었고 반면에 다른 도구는 development life cycle에서 다른 활동을 하는 requirements에 연결하기 위하여 개발되었다.
  - 한 분야에서 작동하는 tool
    - Cadre Team Work for Real-Time Structured Analysis (CADRE)
  - 여러 분야로부터 정보를 연결한 tool
    - Requirements Traceability Manager (RTM)
    - SLATE
    - DOORS

COMMERCIAL CASE Tools for Traceability

- DOORS
  - 정보의 관리에 객체 지향 데이터 베이스를 이용
- RTM
  - 정보를 획득하고 관리를 제공하기 위해 관계형 데이터베이스 구조를 이용
- SLATE
  - 다중사용자, 클라이언트-서버, 시스템의 동적 표현을 제공한 객체 지향 방법을 이용

- 상업 tool에 이용된 또 다른 방법은 Hypertext 접근임
  - 키워드 또는 구문에 구분할 수 있는 식별자를 둠

Future Trends and Conclusions

- 미래의 Traceability는 자동 보조와 자연언어에서 Traceability의 모든 것을 제공하는 것임
- 진행중인 연구
  - 구조와 분류에 대한 엔티티의 자동화된 할당
  - 구조와 분류를 개발하는데 사용된 방법과 관계없는 Traceability
  - 최하위 수준으로 Requirements에서 제품 속성을 추적

Future Trends and Conclusions

- 현재 수동 기술로 시작에서부터 최종 생산품까지 가는 것이 어려운 것일지라도, 자동화된 할당과 분류를 위한 도움이 됨
- 각각의 접근에서, CASE 툴은 자동으로 tracing을 제공하지만, 의사 결정을 위한 것만 운영자 입력을 요구함
- 이러한 tool은 Traceability를 위한 연습의 현재 상태 위에서 중요한 진보를 나타냄