A study on the XML-based software Components Specification Method and Supporting Design

Joon-Bum Park°, Soo-Lyul Oh, Han-Suk Choi

Division of Information Computer Engineering, Mokpo National University

요 약
본 논문은 컴포넌트 S/W의 저장, 관리, 유동 활성화를 위하여 컴퓨터의 명세서를 구현화하고 구현화된 컴퓨터의 명세서를 표준 모바이아인 XML 기반으로 모델링하는 것이다. 클래스 라이브러리의 한계점을 극복하고 소프트웨어 재사용성을 창출시키기 위한 단위로 만들어진 컴퓨터 단위의 재사용 기법의 XML 기반의 컴퓨터 명세서로 안해 더욱 활성화 될 수 있을 것이다.
또한, 컴퓨터의 메타데이터 및 리소스 정보, 외부 인터페이스들의 논리적 구조 및 시맨틱 표현 기법을 연구하고 컴퓨터의 명세서 표현을 위한 메타데이터 모델을 근거로 컴퓨터 명세서를 효과적으로 생성할 수 있는 GUI 기반 명세서 기반 도구를 설계한다.

1. 서론
소프트웨어 규모는 커지고 있으나, 소프트웨어 개발 기술의 발전 속도는 하드웨어 발전 속도에 비해 뒤틀린 실정이다. 대부분의 소프트웨어 프로젝트들은 개발 시간이 지연되거나 예산을 초과하거나, 고객이 원하는 기능들의 소프트웨어를 생산하지 못함으로 인해 실패하는 경우가 많다. 이러한 소프트웨어의 개발과 유지보수 비용의 증가로 인해 소프트웨어 제작이 악재가 발생하게 되었다.[1][2][3] 이에 대응하기 위해 소프트웨어 개발과 유지보수 비용이 줄이기 위해 개발이 필요하다고 규정한 설정[4][5][6][7]

2. 컴퓨터의 명세서 구조
2.1 컴퓨터의 명세서 구조
컴퓨터에 대한 명의는 다양한. Rational 사의 Philippe Krutchen은 "컴퓨터는 잘 정의된 아키텍처 상에서 어떠한 기능을 수행하는 시스템 복잡적으로서 때


2.2 컴퓨터의 명세서 구조
(1) 컴퓨터의 명세서 구조 분석
컴퓨터의 명세서는 <그림>과 같이 3가지 영역으로 구분된다.
3. XML 기본 보안 모델

3.1 보안 모델 명세 문서의 정의 (DTD)
앞의 2.1의 보안 모델 명세 구조는 <그림1>, <그림2>, <그림3>, <그림4>와 같이 전체 보안 모델 문서 (DTD)를 정의 하였다. 각 모델에 대한 DTD는 다음과 같다.
(1) 유스케이스 모델
<!ELEMENT Usecase_Diagram (Usecase_Module,
Usecase_Actor*, Usecase_System*, Usecase_Usecase*,
Usecase_Generalization*, Usecase_Extends*,
Usecase_Entity_Position*, Usecase_Relation_Position*)>
<!ELEMENT Usecase_Usecase Module
(Usecase_Module_Name,
Usecase_Module_Document)
<!ELEMENT Usecase_Module_Name
(#PCDATA)>
<!ELEMENT Usecase_Module_Document
(#PCDATA)>
(2) 객체 모델
<!ELEMENT Class_Diagram (CD_Base,CD_Module,CD_Class*,
CD_Operation*, CD_Attribute*,
CD_Generalization*, CD_Dependency*,
CD_Type*, CD_Aggregation*, CD_Composition*,
CD_Role*,
CD_Entity_Position*, CD_Relation_Position*)>
<!ELEMENT CD_Base EMPTY>
<!ATTLIST CD_Base
CD_Base_Items_Count CDATA #IMPLIED
CD_Base_Version CDATA #IMPLIED>
(3) 컴포넌트 모델
<!ELEMENT Component_Diagram (CPD_Base,
CPD_Module, CPD_Component*, CPD_Interface*,
CPD_Node*, CPD_Dependency*, CPD_Entity_Position*,
CPD_Relation_Position*)>
<!ELEMENT CPD_Base EMPTY>
<!ATTLIST CPD_Base
CPD_Base_Items_Count CDATA #IMPLIED
CPD_Base_Version CDATA #IMPLIED>
3.2 XML 기본 보안 모델에 DTD를 이용하여 XML 문서를 작성하였다.
<!ELEMENT Usecase_Diagram>
<!ELEMENT Usecase_Usecase Module
(Usecase_Module_Name,
Usecase_Module_Document)
<!ELEMENT Usecase_Module_Name
(#PCDATA)>
<!ELEMENT Usecase_Module_Document
(#PCDATA)>

(2) 컴포넌트 모델에서 다이어그램 구조의 변경
각 컴포넌트의 다이어그램은 유스케이스 모델에서 유스케이스 다이어그램, 객체 모델에서 클래스 다이어그램, 컴포넌트 모델에서 컴포넌트 다이어그램으로 구성된 다이어그램 구조를 나타내고 각 클래스들의 속성과 행위를 기술한다. 컴포넌트 다이어그램은 소프트웨어의 물리적 단위의 구성과 연결상태를 나타내게 된다. 시리즈 다이어그램은 단일 객체의 시간적 계주로 하여 시간의 효과를 나타내어 메시지의 순서에 역할을 두고 표현한다.
4. XML 기반 콤포넌트 명세서 생성 지원도구 설계

4.1 지원도구 요구사항 분석

 컴퓨터시스템에 Editor는 컴퓨터 개발자의 UML기술을 이용하여 개발과 동시에 컴퓨터시스템을 작성할 수 있도록 개발한다. 컴퓨터시스템의 구조화된 DTD를 이용하여 컴퓨터시스템의 설계상적 정의를 기하고 수정, 삭제, 저장할 수 있어야 한다. 컴퓨터시스템에서 Editor 개발을 위한 요구사항은 다음과 같다.

 첫째, UML을 이용한 컴퓨터시스템을 가능하도록 환경을 제공한다.
 둘째, 컴퓨터시스템에서 Editor 개발을 위한 구현 환경을 철저히 분석하여 원도우 기반의 허용한 사용자 인터페이스를 제공한다.

 4.2 지원 도구 설계

 컴퓨터시스템의 특성은 아래와 같다
- 데이터그램의 작성 및 편집
- 데이터그램 부가기능
- Drag & Drop 편집기능
- 미리 보기 화면이나 웹 브라우저에서 보기기능을 별도로 제공한다.
- 데이터그램과 트리 구조의 작업이 일치
- 기능 및 화면설계

 컴퓨터시스템에서 Editor에서 제공되는 주요 기능은 다음과 같다.

 1. 과일 생성

 세부설계: 새로운 컴퓨터시스템 작성
 연기: XML문서를 분리하면서 요구사항을 UML로 변환 저장: 컴퓨터시스템을 XML기반으로 작성
 다른이용을 저장: 활성화된 정보 UML 데이터그램을 컴퓨터시스템으로 저장

 인체: 활성화된 정보의 컴퓨터시스템을 인체
 단기: 활성화된 정보 단기
 모두단기: 열린 모든 정보 단기
 접대기: 모든 정보 단기 저장 여부를 묻는 후 프로그램을 종료한다.

 2. 편집메뉴

 실행취소: 실행한 명령을 취소한다.
 갈라내기: 선택된 부분을 지우고 기록시킨다.
 복사하기: 선택된 부분을 기록시킨다.
 지키기: 갈라내거나 복사하기 한 내용을 삽입한다.
 지우기: 선택한 그룹은 지우는.

 3. 보기 메뉴

 기본도구상자: 기본 도구모음 Toolbar을 보여줄 것을 선택한다.
 컴퓨터시스템에서 보기: 컴퓨터시스템을 보여준다.
 DTD문서보기: 컴퓨터시스템의 DTD구조를 보여준다.

 4. 창 메뉴

 창설명: 스탬프모양, Cascade등 기본 창 정렬을 지원한다.
 모두닫기: 열려있는 창을 모두 최소화한다.

 5. 도움말

 요청: 도움말을 찾는다.
 도움말: 도움말의 내용을 보여준다.

 6. 참조문헌