

무기체계 설계/형상정보 관리 시스템을 위한 메타 데이터 모델링

Meta Data Modeling for Weapon System Design/Configuration Data Management System

김기백*
Kim, Ghiback

ABSTRACT

In general, weapon system design/configuration data consist of system structure information which is linked to part information, documents and drawings. For configuration management, version and revision control are necessary and access control of users to information should be managed for information security. Configuration data of weapon systems have various kinds of different meta data which are contained in the structure as well as attributes of parts and documents information. If neutral types of meta data models be used for building configuration management system, they can be applied to many different kinds of weapon systems with a little customization. In this paper, five meta data models are supposed and implementation results of them by using CBD(component based design) methodology are presented.

주요기술용어(주제어) : Meta Data Model, Weapon System Design/Configuration Data, Component, GBL: General Break down Lists, LCN: LSA(Logistic Support Analysis) Control Number

1. 서론

무기체계의 연구개발 생산 및 배치운용 단계에서 무기체계의 형상관리는 효율적인 연구개발과 전력 증강에 필수적인 관리활동이다. 특히, 지속적으로 변화하고 개선되는 연구개발 과정에서 무기체계의 형상관리는 더욱 중요하다 할 수 있다. 이미, 민수 분야에서

는 제품 개발에 형상관리의 중요성을 깨달아 PDM(Product Data Management) 도구를 이용하여 제품 생명주기의 전 과정인 연구개발 단계부터 생산까지 제품의 형상과 관련된 자료를 관리하고 있다.

정보 관점에서 무기체계의 연구개발 과정도 경우도 민수의 경우와 크게 다르지 않으며, 무기체계의 종류에 따라 형상자료 또한 다양하게 변화될 수 있다. 특히, 무기체계에 따라 결정되는 분류 또는 속성관련 자료를 동적으로 정의하기에 어려운 점이 있어 민수에 활용되는 PDM 도구와 같은 소프트웨어를 도입하여 활용하는데 어려움이 있었다.

† 2004년 5월 4일 접수 ~ 2004년 6월 17일 심사완료
* 국방과학연구소(Agency for Defense Development)
주저자 이메일 : ghiback@premail.co.kr

무기체계의 형상정보는 구조 관점에서 무기체계의 형상을 구조 형태로 표현하는 GBL, WBS, LCN Family Tree가 있으며, 이들 구조 정보에 연결된 구성품에 관련된 정보, 그리고 구성품을 공급하는 공급 업체에 관련된 정보로 구성된다. 또한, 무기체계와 관련된 규격서, 연구보고서 등 각종 문서 자료도 구조 정보와 연결된다. 이들 자료들은 상호 밀접한 관계를 통하여 무기체계의 형상을 표현한다^[1].

무기체계의 형상을 관리하기 위하여, 위에서 식별된 자료를 토대로 무기체계와 관련된 형상정보를 모델링하여 민수의 PDM 도구와 같은 기능을 갖는 소프트웨어를 개발할 수도 있다.

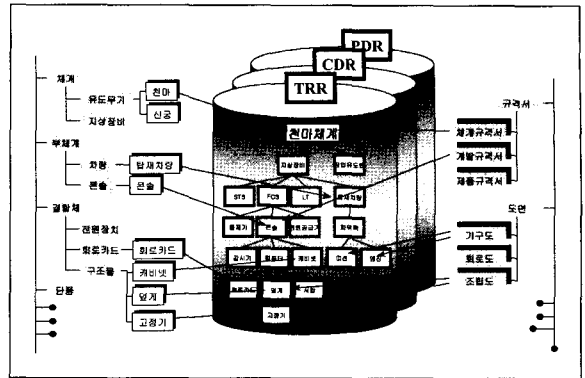
그러나 무기체계 마다 독특하게 정의되는 분류 정보, 속성 정보를 가변적으로 처리하고, 지속적으로 변화하는 연구개발 환경에 적응적으로 대처하기 위하여 무기체계의 설계/형상정보 모델을 개발하기 위한 무기체계에 중립적 형태의 메타 데이터 모델^[2]의 필요성이 제기되었다. 따라서 본 연구에서는 다양한 무기체계의 설계/형상정보 모델에 중립적인 메타 데이터 모델을 제시하고 이를 삼성SDS의 CBD 방법론인 INNOVATOR CBD 방법론 1.0^[3]을 적용하여 구현하였다.

2. 무기체계 설계/형상정보

가. 무기체계 설계/형상정보

무기체계의 설계/형상정보는 무기체계를 하드웨어 및 소프트웨어 관점에서 물리적으로 구분한 구조로서 무기 체계의 분할 구조를 트리 형태로 표현한 GBL, 체계 개발에 사용되는 모든 사업의 구성 요소를 세분화하여 업무요소 및 업무요소 간의 상하관계를 표현한 WBS, 종합군수지원 요소개발에 필수적인 군수지원 분석과정에서 활용되는 무기체계를 분류한 LCN Family Tree 가 중심이 된다.

그림 1의 무기체계 설계/형상정보 모델에서 알 수 있듯이 무기체계의 구조정보는 구조를 구성하는 각 노드가 구성품을 표현하는 독립적인 구성품 정보와 연계되고 또 무기체계의 특성을 표현하는 규격서, 연구보고서, 도면과 같은 문서와 연계된다. 그리고



[그림 1] 무기체계 설계/형상정보 모델

이들 자료는 무기체계 연구개발의 주요 사업단계 (SDR, CDR, TRR 등등) 별로 베이스라인을 형성하여 관리된다.

구성품 정보에는 각 구성품에 대한 상세한 정보가 포함되는데, 구성품의 기본정보, 버전정보, 관련 문서/도면 정보, 관리정보, 가격정보, 특성치정보 등이 관리된다. 특히, 특성치 정보는 관련 구성품에 종속되는 정보로서 무기체계에 적용되는 구성품에 따라 다양한 형태로 정의된다.

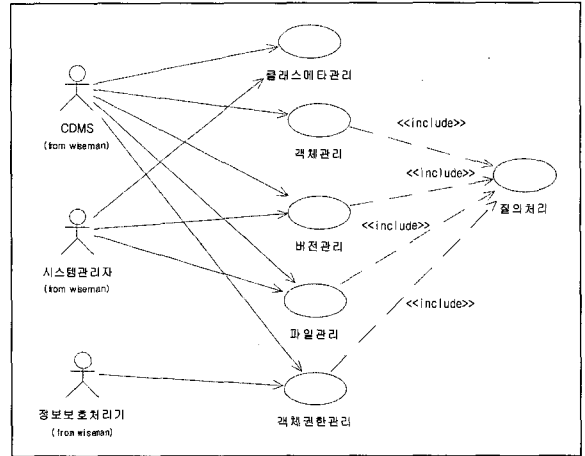
문서/도면 정보에는 GBL, WBS 또는 구성품에 연결된 문서나 도면에 대한 기본정보, 버전정보, 관리정보, 색인정보 등이 관리된다. 색인항목은 구성품의 특성치정보와 유사한 특징을 갖는데, 문서/도면의 분류마다 적합한 색인정보가 정의된다. 문서/도면의 분류 또한 적용되는 무기체계에 따라 서로 다르게 정의된다.

이와 같은 형태로 표현되는 무기체계의 설계/형상정보는 다음과 같은 공통적인 특징을 갖는다.

- 구조 형태의 정보
GBL, WBS, LCN Family Tree, 문서/도면의 분류, 구성품의 분류 등을 표현하기 위한 구조 형태의 정보
- 리스트 형태의 정보
구성품의 공급업체 정보와 같이 테이블 또는 리스트 형태로 표현되는 정보
- 형상정보간의 링크 정보
GBL과 구성품, 구성품과 문서/도면, 구성품과 공

급업체간의 관계와 같이 다양한 형태로 정의되는 정보간의 관계 정보

- 가변 속성
구성품의 특성치 정보나 문서/도면의 색인 정보와 같이 무기체계의 분류에 따라 정의되는 가변형 속성 정보를 관리해야 하는 특징
- 참조 리스트 정보
문서 형태, 통화 단위, 문서 구분 등과 같이 가변 속성에 적용되는 데이터로써 무기체계의 특성에 따라 정의되는 데이터 목록
- 권한 제어
데이터 보안측면에서 구성품 또는 문서/도면에 대한 사용자의 접근 권한을 제어해야 하는 특징
- 파일 관리
문서와 도면에 첨부되는 물리적인 파일을 관리해야 하는 특징
- 볼트(Vault) 관리
파일을 저장소에 독립적으로 관리할 수 있도록 파일 저장소를 관리해야 하는 특징
- 버전 제어
구성품 또는 문서/도면 같은 대상의 버전을 관리해야 하는 특징



[그림 2] 유즈케이스 다이어그램

3. 메타 데이터 모델링

무기체계의 설계/형상정보를 효율적으로 모델링하기 위한 메타 데이터 모델은 앞 절에서 식별한 8가지의 특징을 지원할 뿐만 아니라 클래스 형태로 모델링이 지원되도록 설계되어야 한다. 그림 2에 메타 데이터 모델을 위한 최상위 단계의 유즈케이스를 식별하였다.

메타 데이터 모델을 통하여 모델링 된 설계/형상정보 모델을 관리하는 클래스 메타관리, 형상정보 객체를 관리하는 객체관리, 객체의 권한을 관리하는 권한관리, 형상정보와 연계된 파일을 관리하는 파일관리, 객체의 버전을 관리하는 버전관리 유즈케이스가 식별되었다.

가. 메타 데이터 모델링

설계/형상정보를 모델링을 지원하기 위한 메타 데이터 모델로 구조 클래스, 데이터 클래스, 관계 클래스, 구조관계 클래스, 참조 클래스 등 5개의 클래스가 식별되었다.

구조 클래스는 설계/형상정보 모델에서 GBL, WBS, 구성품의 분류와 같이 트리 형태의 자료를 표현하기 위한 클래스이다. 트리의 각 노드에는 객체 속성을 동적으로 추가하거나 변경할 수 있고, 파일 또한 첨부할 수 있다.

데이터 클래스는 관계형 데이터 모델의 테이블과 같은 특징을 갖는다. 2차원 배열인 테이블 형태의 데이터를 모델링 할 수 있다. 각 속성별로 키(Key), 필수입력(Not Null), 데이터 타입(문자, 숫자, 날짜), 데이터 길이, 참조 클래스 등의 제약조건을 정의할 수 있다.

관계 클래스는 클래스간의 관계를 정의하는 클래스이다. GBL과 WBS, 구성품과 문서같이 상호 밀접한 관계를 갖는 클래스간 관계를 정의 한다.

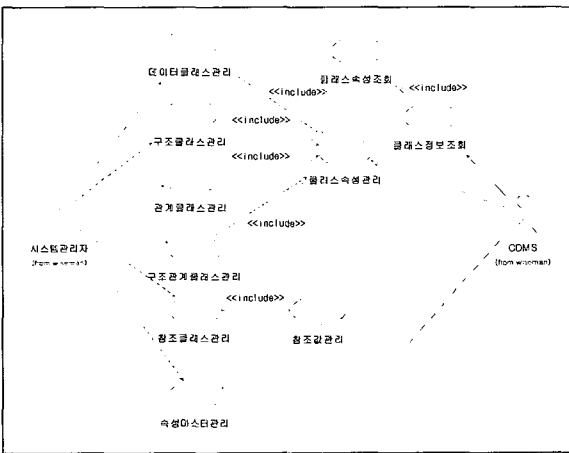
구조관계 클래스는 객체간의 상하 관계를 표현하기 위한 클래스이다. 구성품을 이용하여 체계 구조를 표현하는 자료구조에 적합한 모델로써 민수에서 사용하는 BOM(Bill of Material)을 표현할 수 있다.

참조 클래스는 구조, 데이터, 구조관계 클래스의 가변 속성의 참조 데이터로 사용될 클래스의 데이터 모델이다. 참조 데이터란 시스템의 운영중의 동적으로

정의되고 변경되어 사용되는 데이터 목록을 의미한다. 이 클래스를 가변 속성의 입력 대상 데이터로 연결하여 사용하면 자료를 목록에서 선택하여 입력함으로써 입력을 쉽게 할 수 있으며 그 결과 입력 오류를 방지하는 효과를 얻을 수 있다.

메타 데이터의 유즈케이스를 식별한 그림 3에 클래스 속성을 통합 관리하는 클래스속성관리와 클래스 정보를 조회하는 클래스정보조회 그리고 참조클래스의 참조값관리가 추가로 식별되어 있다.

표 1에 메타 데이터 모델에서 지원하는 클래스의 특징에 대하여 요약하였다. 각 클래스별로 버전관리 기능, 파일관리 기능 등을 선택적으로 적용할 수 있도록 모델링 되었다.



[그림 3] 메타 데이터 유즈케이스 다이어그램

[표 1] 메타 데이터 지원 클래스

구조클래스	트리형태의 구조 자료를 표현하는 클래스
데이터 클래스	리스트 형태의 데이터를 표현하는 클래스
관계 클래스	두 객체 간의 관계를 표현하는 클래스
구조관계 클래스	객체 간에 구조적인 상하 관계를 표현하는 클래스
참조 클래스	가변 속성에 활용될 데이터 목록을 표현하는 클래스

나. 메타 데이터 모델 설계

1) 클래스 다이어그램 작성

기 식별한 유즈케이스 다이어그램과 메타 데이터 모델을 기반으로 클래스 다이어그램이 작성되었다. 그림 5는 분석 단계의 클래스 다이어그램으로 메타 데이터 모델 보여주는 클래스 다이어그램이다.

상위 클래스인 ClassMeta를 상속받아 구조클래스(StructureClass), 데이터클래스(Data Class), 관계클래스(RelationClass), 구조관계클래스(StructureRelationClass), 참조클래스(DictionaryClass)가 아래에 표현되어 있다.

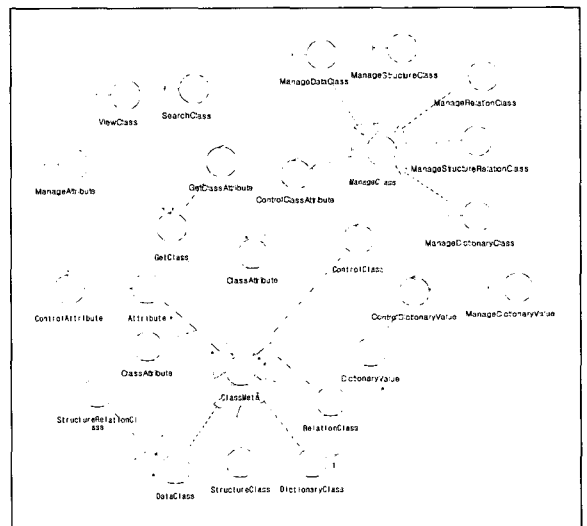
클래스 속성을 통합 관리하는 클래스로 Attribute 클래스가 식별되어 있다.

그림 6은 메타 데이터 모델의 인스턴스인 객체를 관리하는 클래스 다이어그램을 보여준다. 상위 클래스인 Object를 상속받아 각 데이터 모델과 관련된 객체에 대한 클래스가 표현되어 있다.

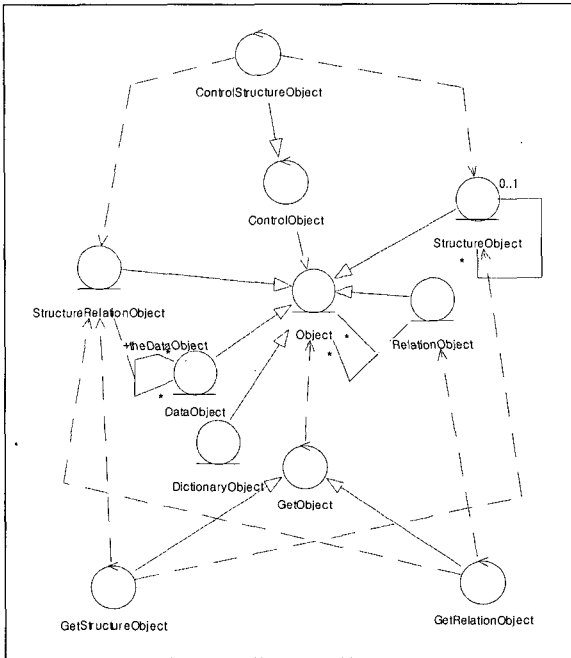
이외에 객체의 권한관리, 버전관리, 파일관리, 질의 처리에 대한 클래스 다이어그램이 식별되었다.

2) 컴포넌트 식별

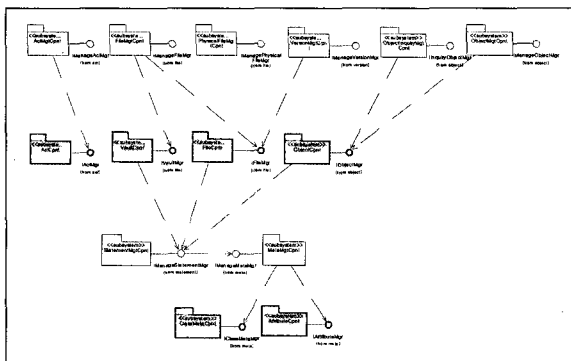
메타 데이터 모델은 여러 형태의 무기체계 설계/형상정보 관리 시스템의 지속적인 발전에 따른 활용성을 높이기 위하여 컴포넌트 형태로 설계되었다. 컴포



[그림 5] 클래스 메타관리 클래스 다이어그램



[그림 6] 객체 관리 클래스 다이어그램



[그림 7] 컴포넌트 아키텍처 다이어그램

먼트 식별을 위하여 유즈케이스 다이어그램과 클래스 다이어그램을 바탕으로 시스템 인터페이스와 비즈니스 인터페이스를 식별하였다.

시스템 인터페이스는 서로 관련이 있는 유즈케이스를 묶어서 표현하였으며, 비즈니스 인터페이스는 클래스 다이어그램에서 코어 클래스와 그것에 종속적인 클래스를 묶어서 나타내었다. 다음으로 시스템 인터페이스를 어플리케이션 컴포넌트로 비즈니스 인터페이스를 비즈니스 컴포넌트로 식별하였다^[3].

메타 데이터 모델의 컴포넌트 아키텍처 다이어그램인 그림 7에는 총 14개의 컴포넌트가 식별되었는데, 8개의 어플리케이션 컴포넌트가 황색으로 6개의 비즈니스 컴포넌트가 회색으로 표현되어 있다.

3) 컴포넌트 인터페이스^[4]

다섯 가지 클래스 타입에 따른 객체에 대한 인터페이스를 단순하게 정의하기 위하여 매소드 오버라이딩을 적용하였으며, 조회 인터페이스의 경우 매소드 오버로딩을 적용 하나의 인터페이스를 통하여 다양한 조회 패턴을 처리할 수 있도록 설계하였다.

가) 비즈니스 컴포넌트

비즈니스 컴포넌트는 메타 데이터 모델의 처리 논리를 구현한 컴포넌트로 메타 클래스를 관리하는 ClassMetaCpnt 컴포넌트, 클래스의 속성을 관리하는 AttributeCpnt 컴포넌트, 클래스의 객체를 관리하는 ObjectCpnt 컴포넌트, 객체의 접근 권한을 관리하는 AclCpnt 컴포넌트, 파일에 대한 메타정보를 관리하는 FileCpnt 컴포넌트, 볼트를 관리하는 VaultCpnt 컴포넌트 등 6개의 컴포넌트로 구성되어 있다.

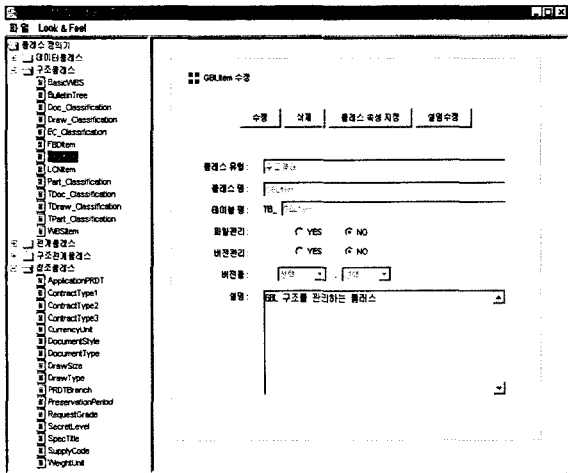
나) 어플리케이션 컴포넌트

어플리케이션 컴포넌트는 비즈니스 컴포넌트의 인터페이스를 묶어 어플리케이션의 요청을 처리하는 인터페이스 컴포넌트로서 어플리케이션으로부터 클래스와 속성에 대한 조작 요청을 처리하는 MetaMgtCpnt 컴포넌트, 어플리케이션으로부터 객체에 대한 조작 요청을 처리하는 ObjectMgtCpnt 컴포넌트, 어플리케이션으로부터 객체에 대한 제어 요청을 처리하는 AclMgtCpnt 컴포넌트, 어플리케이션으로부터 파일과 볼트에 대한 조작 요청을 처리하는 FileMgtCpnt 컴포넌트, 물리적인 파일의 송수신 요청을 처리하는 PhysicalFileMgtCpnt 컴포넌트, 어플리케이션으로부터 객체의 버전에 대한 조작 요청을 처리하는 VersionMgtCpnt 컴포넌트, 어플리케이션으로부터 객체에 대한 질의 요청을 처리하는 ObjectInquiryMgtCpnt 컴포넌트, 객체에 대한 조작 요청을 물리적인 데이터를 조작하는 요구로 변경하는 StatementMgtCpnt 컴포넌트 등 8개의 컴포넌트가 있다.

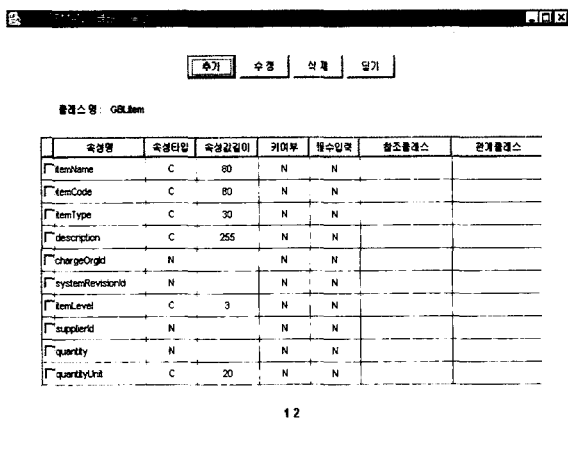
각 컴포넌트별 세부 인터페이스 목록은 부록에 첨부하였다.

4. 메타 데이터 모델 구현

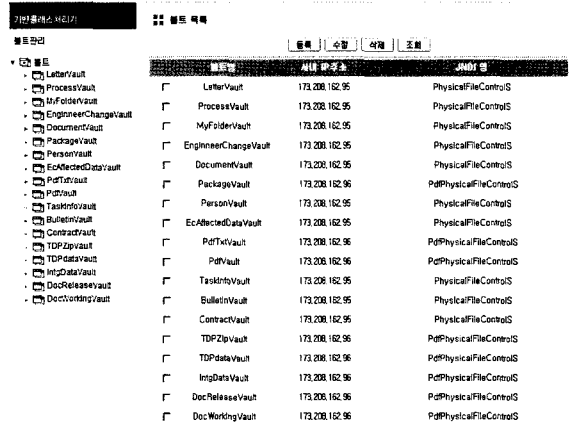
메타 데이터 모델은 무기체계의 설계/형상정보 모델을 정의하는 클래스 정의기와 파일 저장소를 정의하는 볼트 정의기 그리고 어플리케이션의 요청을 처리하는 8개의 컴포넌트와 4개의 비즈니스 컴포넌트로 구현되었다.



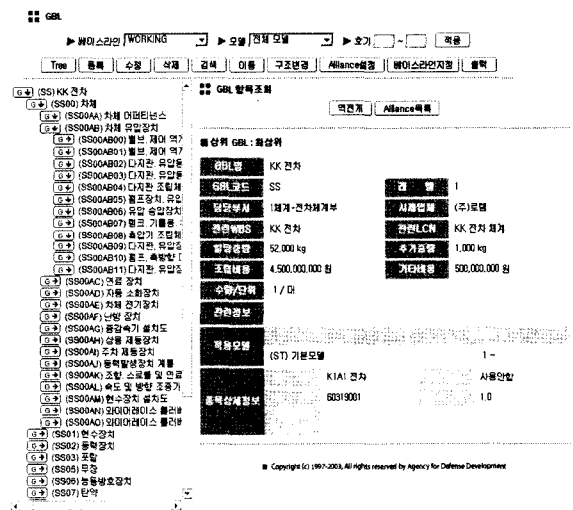
[그림 8] 클래스 정의 화면



[그림 9] 속성 정의 화면



[그림 10] 볼트 관리 화면



[그림 11] 무기체계 GBL 정보 (예)

그림 8은 설계/형상정보 모델을 정의하는 클래스 정의기의 화면으로, 구조클래스인 GBL에 대한 클래스 정의 화면이며 그림 9는 클래스의 속성 정의 화면이다.

그림 10은 웹으로 구현된 볼트를 관리하는 볼트 정의기의 화면이다.

가. 메타 데이터 모델의 적용
메타 데이터 모델은 국방과학연구소에서 개발한 무기체계 설계/형상정보 관리 시스템(WISEMAN:

구성품 정보 조회

구조

이력 | 관련GBL | 관련LCN | 달기

기본정보

종류(종류)	KK전차 차체
종류(종류)	Body
품 명	60039282
NSU	
중량(중량)	None
중 량	51900
중 량 단	
중량단위	kg
입력명 / [CAGE 코드] / [국명]	
품번 / [품명] / [비전]	
품번 / [품명] / [비전]	
구분 / [번호] / [재목] / [비전]	

가격정보

단 가	작성중입니다
단가수준	작성중입니다 %

관리정보

단 가	1.4	등록자	관리자
승용권	공용	등록일	2003/12/05 19:26:53

[그림 12] 무기체계 구성품 정보 (예)

도면 정보 조회

무장

이력 | 관련WBS | 관련LCN | 달기 | 관련구성품

기본정보

도면번호	AVI-전차장일자-003	1.0
도 명 명	전차장일자 간섭	
도면호기		
도면대수		
관리정보		
승용권(승용권)		
도면종류		
품 명	전차장일자 간섭 체크를 위한 시뮬레이션 결과	
변경 사유		
첨부파일	HFE_전차장일자.avi	

관리정보

등록자	박삼준	등록일	2003/12/05 14:58:46
승용권장자	관리자	승용권등록일	2004/01/27 20:21:38
승용권	관리자	유효일	

색인항목

관리사명	
도면개봉자명	

[그림 13] 도면 정보 (예)

Weapon system information Integration for System Engineering and MANagement)의 데이터 모델링 및 데이터 관리를 위한 기반 소프트웨어 활용되어 무기체계 개발에 적용되고 있다.

그림 11, 그림 12, 그림 13은 구현이 완료되어 시범 운영중인 무기체계 설계/형상정보 관리 시스템의 화면으로 GBL 정보, 구성품 정보, 도면 정보를 보여 주고 있다.

5. 맺 음 말

성공적인 무기체계의 연구개발을 위하여 지속적으로 변화하고 발전되는 연구개발 환경에서 무기체계의 형상관리는 매우 중요한 요소이다. 그러나 이와 같이 변화되는 환경에서 다양한 무기체계의 형상관리에 대한 요구사항을 만족하는 설계/형상정보를 모델링 하는데 어려움이 있었다.

본 연구에서는 이러한 문제점을 해결하기 위하여 무기체계의 설계/형상정보를 모델링을 지원하는 메타 데이터를 모델링 하였고 이를 컴포넌트 형태로 구현 하였다.

메타 데이터 모델을 이용하여 설계/형상정보를 모델링 함으로써 메타 데이터가 제공하는 기능을 활용하여 무기체계 설계/형상정보를 쉽게 모델링하고 구현할 수 있으며, 또한 데이터가 저장되고 운영되는 환경에 독립적으로 시스템을 개발할 수 있다.

향후 개선될 연구개발 절차에 따라 새롭게 보완될 시스템도 본 논문에서 제시한 메타 데이터 모델을 이용하여 데이터를 모델링하여 시스템을 개발한다면, 기존에 구축된 데이터와 호환성을 갖추고 동일한 아키텍처 상에서 일관성을 갖도록 시스템을 확장할 수 있다.

참 고 문 헌

[1] 정기원, 김철환, 임춘성 “연구개발형상 관리체계 시험구현”, CESD-409-980064, 연구보고서, 국방과학연구소, 1998.

[2] 김기백 “WISEMAN 시스템의 정보공유 기반 소프트웨어 요구규격 식별 및 설계”, IEDC-409-020527 국방과학연구소, 2002, pp.59~61.

[3] “Innovator CBD 방법론”, 삼성SDS, 2002

[4] “WISEMAN 2차시제 SDMS 소프트웨어 컴포넌트 설계”, 국방과학연구소, 2004.

부록: 컴포넌트별 인터페이스

1. 비즈니스 컴포넌트

가. ClassMetaCpnt 컴포넌트

- getClass(), getClasses(): 클래스의 상세 정보 조회
- createClass(): 클래스의 생성
- updateClass(), updateClasses(): 클래스의 정보 수정
- deleteClass(), deleteClasses(): 클래스 삭제
- createClassAttribute(): 클래스 속성 등록
- updateClassAttribute(): 클래스 속성 변경
- deleteClassAttribute(): 클래스 속성 삭제
- getClassAttributes(): 클래스 속성 조회
- getMetaData(): 클래스의 메타정보 조회
- createDictionaryValue(): 참조 클래스의 객체 생성
- updateDictionaryValue(): 참조 클래스의 객체 수정
- deleteDictionaryValue(): 참조 클래스의 객체 삭제
- getDictionaryValue(), getDictionaryValues(): 참조 클래스의 객체ID를 이용한 조회

나. AttributeCpnt 컴포넌트

- createAttribute(): 속성 마스터의 클래스 속성 정보를 추가

- updateAttribute(), updateAttributes(): 속성 마스터의 클래스 속성정보 수정
- deleteAttribute(), deleteAttributes(): 속성 마스터의 클래스 속성정보 삭제
- getAttribute(), getAttributes(): 속성 마스터의 클래스 속성 조회
- getMetaData(): 클래스의 메타정보 조회

다. ObjectCpnt 컴포넌트

- getObject(), getObjects(): 조건을 이용하여 선택한 객체의 상세정보 조회
- getDictionaryValues(): 참조 객체의 값을 조회
- getTreeObjects(): 트리의 하위 노드 검색
- getChildren(): 노드의 하위 객체정보 조회
- getParents(): 노드의 상위 객체정보 조회
- getRelationObjects(): 연결되어 있는 상대방 객체의 정보 조회
- getObjectID(): 객체 아이디를 생성
- createObject(), createObjects(): 객체를 데이터베이스에 생성
- updateObject(), updateObjects(): 객체 수정
- deleteObject(), deleteObjects(): 객체 삭제
- copyObject(): 객체 복사
- moveObject(): 구조 객체의 노드 이동

라. AclCpnt 컴포넌트

- createAuthorization(): 객체권한정보 설정
- deleteAuthorization(): 객체권한정보 해제
- checkAuthorization(): 객체권한종류 조회

마. FileCpnt 컴포넌트

- getFile(), getFiles(): 파일 상세정보 조회
- createFile(): 파일 등록
- updateFile(), updateFiles(): 파일정보수정
- deleteFile(), deleteFiles(): 파일 삭제
- copyFile(): 파일 복사
- moveFile(): 파일 이동

바. VaultCpnt 컴포넌트

- getVault(), getVaults(): 볼트 정보 검색

- createVault() : 볼트 정보 등록
- updateVault(), updateVaults() : 볼트 정보 수정
- deleteVault(), deleteVaults() : 볼트 정보 삭제
- moveVault() : 볼트의 노드 이동
- assignVault() : 클래스에 볼트를 할당
- revokeVault() : 클래스에 할당된 볼트를 해지
- getTargetVault() : 클래스의 볼트정보조회

2. 어플리케이션 컴포넌트

가. MetaMgtCpnt 컴포넌트

- ClassMetaCpnt 컴포넌트와 AttributeCpnt 컴포넌트의 모든 인터페이스를 포함

나. ObjectMgtCpnt 컴포넌트

- getObjectID() : 객체의 아이디를 생성
- createObject(), createObjects() : 객체를 데이터베이스에 생성
- updateObject(), updateObjects() : 객체 수정
- deleteObject(), deleteObjects() : 객체 삭제
- copyObject() : 객체 복사
- moveObject() : 구조 객체의 노드 이동

다. AclMgtCpnt 컴포넌트

- AclCpnt 컴포넌트의 모든 인터페이스를 포함

라. FileMgtCpnt 컴포넌트

- FileCpnt 컴포넌트와 VaultCpnt 컴포넌트의 모든 인터페이스

마. PhysicalFileMgtCpnt 컴포넌트

- beginPutBuffer() : 파일 전송 시작
- putBuffer() : 버퍼만큼 파일 블록 전송
- endPutBuffer() : 파일 전송 완료
- beginGetBuffer() : 파일 수신 시작
- getBuffer() : 버퍼만큼 파일 블록 요청
- endGetBuffer() : 파일 수신 완료

- checkDirectory() : 해당 디렉토리의 존재여부 확인 및 필요시 디렉토리 생성

바. VersionMgtCpnt 컴포넌트

- transfer() : 객체를 릴리즈 상태로 변경
- checkIn() : 객체를 체크인
- checkOut() : 객체를 체크아웃
- undoCheckOut() : 객체의 체크아웃 취소
- release() : 객체를 릴리즈
- newRelease() : 객체를 뉴 릴리즈
- getObjectStatus() : 객체의 상태정보 조회
- setVersioningRule() : 객체의 버전관련 규칙을 설정
- getVersioningRule() : 객체의 버전관련 규칙 조회

사. ObjectInquiryMgtCpnt 컴포넌트

- getObject(),getObjects() : 조건을 이용하여 선택한 객체의 상세정보 조회
- getDictionaryValues() : 참조 객체의 값을 조회
- getTreeObjects() : 트리의 하위 노드 검색
- getChildren() : 노드의 하위 객체정보 조회
- getParents() : 노드의 상위 객체정보 조회
- getRelationObjects() : 객체와 연결되어 있는 상대방 객체의 정보 조회

아. StatementMgtCpnt 컴포넌트

- makeInsertStatement() : 객체 생성 구문을 분석하여 물리적인 데이터 조작 구문 생성
- makeUpdateStatement() : 객체 변경 구문을 분석하여 물리적인 데이터 조작 구문 생성
- makeDeleteStatement() : 객체 삭제 구문을 분석하여 물리적인 데이터 조작 구문 생성
- makeSelectStatement() : 객체 검색 구문을 분석하여 물리적인 데이터 조작 구문 생성