

원자력 해체시설 특성관리 시스템을 위한 CBD 프로세스의 적용 방안

(CBD process applying for DEFACS)

조운형[‡] 박승국[§] 최윤동[¶] 문제권^{*}
(Woonhyoung Cho) (Seungkook Park) (Yundong Choi) (Jeikwon Moon)

요 약 원자력 시설 해체 사업은 해체 시 방사성 물질이 발생하기 때문에 시설 해체에 있어서 일반적인 해체 공법을 사용하기 어렵다. 그렇기 때문에 원자력 시설의 해체 계획을 수립하는데 해체 대상 시설의 성격을 조사하고 파악하는 것이 매우 중요하며 해체 대상 원자력 시설의 특성 자료 조사는 해체 폐기물의 양을 예측하고 해체 사업의 비용 산정에 크게 활용이 된다. 한국원자력연구원에서는 이를 목적으로 해체사업 대상 시설의 특성자료를 관리하는 시스템 DEFACS(Decommissioning Facility Characterization DB System)을 개발하였다. 그러나 원자력 시설 해체에는 시간이 오래 걸리기 때문에 부득이하게 해체 사업 중에 시스템을 개발하였고 이는 지속적인 요구사항의 변경이 발생하는 원인이 되었다. 이러한 이유로 개발에 있어서 일반적인 개발 프로세스를 적용키 어려웠던 바, 본 논문에서는 개발 중 요구사항 변경에 대한 효율적인 대응을 하기 위하여 기존의 CBD(Component Based Development) 프로세스를 CD(Component Development)와 CBSD(Component Based Software Development)로 구분하여 변경사항에 대한 핸들링을 하나의 컴포넌트로 다루고 컴포넌트 별로 CBD를 재적용하여 재귀적으로 프로세스를 핸들링한다. 이로써 컴포넌트 변경에 대한 전체 시스템의 변경점을 최소화하고 컴포넌트와 프로세스의 독립성을 강화함으로써 요구사항 변경으로 인한 프로세스의 중지를 최소화 하였다.

키워드 원자력 해체시설 특성관리 시스템, 원자력, 제염, 해체, CBD, 개발 프로세스, 소프트웨어공학

Abstract Characteristic of decommissioning target facility investigate and understand is very important. because radioactive materials occurs in the decommissioning and dismantling, so it is difficult to use a general dismantling method. Decommissioning nuclear facilities, the characteristics of the target of research to predict the amount of decommissioning waste, decommission projects costing is largely utilized. For this purpose, we developed DEFACS(Decommissioning Facility Characterization DB System) that manage characteristic of decommissioning target facility. But nuclear facility decommissioning takes long time. so we inevitably developed system during decommissioning works, it occurs many system changes. For this reason, it is difficult to apply general development process, so we take CBD process that divide CD(Component Development) and CBSD(Component Based Software Development) for handling change of requirement. it make Component of the overall system for changes to minimize changes by strengthening the independence of components and processes due to changes in requirements were to minimize stopping of the process.

Key words DEFACS, nuclear, decommissioning, dismantling, CBD, development process, software-engineering

1. 서 론

전 세계적으로 수명을 다한 원자력시설이 늘어나고 있으며, 위험성 제거와 재활용 측면에서 시설을 해체하게 되는데 이때 제염 및 해체 활동에 대한 관리의 중요도는 매우 높게 인지되고 있다. 원자력 시설 해체에는 미국, 영국, 프랑스, 독일, 일본, 벨기에를 비롯하여 이미 많은 국가들이 이미 수행을

[‡] 정회원 : 한국원자력연구원 제염해체연구부
whcho@kaeri.re.kr

[§] 비 회원 : 한국원자력연구원 제염해체연구부
skpark2@kaeri.re.kr

[¶] 비 회원 : 한국원자력연구원 제염해체연구부
ydchoil@kaeri.re.kr

^{*} 비 회원 : 한국원자력연구원 제염해체연구부
njkmoon@kaeri.re.kr

논문접수 : 2012년 3월 2일

심사완료 : 2012년 3월 26일

완료한 해체사업의 경험을 토대로 새로운 해체 사업 수행을 위한 데이터베이스 시스템을 개발하여 운영하고 있다. 이에 한국원자력연구원에 서도 연구용 원자로 2호기에 대한 해체 사업을 수행하면서 해체 사업 전반에 대한 정보와 자료 들을 구축, 응용을 위한 해체 사업관리시스템 (DECOMMIS)을 제작하여 사용 중에 있다. 이는 해체 작업 과정에서의 인력소요, 작업 결과 및 방사선 안전관리 자료와 동시에 폐기물의 발생, 처리 및 포장 등의 관리 자료를 처리할 수 있도록 함으로써 폐기물의 발생 및 처리 과정에 관한 이해 증대 뿐만 아니라 폐기물 관리가 좀 더 효율적으로 이루어지도록 하는데 기여할 수 있도록 한다. 하지만 DECOMMIS는 해체사업시행 도중에 개발되어 제염 해체 작업활동, 폐기물 관리, 오염검사 및 오염도 정보 등은 축적 관리 하고 있지만 해체대상 시설에 대한 정보는 포함하고 있지 않다. 시설 처리의 경우 가장 나중에 이루어졌고 방사성 물질에 오염 되어 있기 때문에 이를 제염하는 과정 및 관리가 지속적으로 이루어지고 있기 때문이다. 때문에 이러한 해체 대상 시설물들을 관리하고 나아가 해체대상에 대한 해체계획을 수립하는데 결과물 예측과 같은 문제를 해결하기 위하여 시설 특성 정보를 관리하는 시스템 DEFACS(Decommissioning Facility Characterization DB System)을 개발하였다. 그러나 한국에서는 해체 사업이 처음 시도되었기에 연구용 원자로 1,2호기에 맞추어 시스템을 개발하여야 했고 그렇기 때문에 해체 사업 중에 시스템을 개발 하였던 바, 지속적으로 요구사항에 대한 변경점이 발생하였기에 시스템 설계 단계에서부터 기존의 개발 프로세스를 적용키 어려웠다. 이에 본 논문에서는 지속적으로 변경되는 요구사항과 변수를 효율적으로 처리하기 위하여 CBD(Component Based Development) 프로세스를 CD(Component Development)와 CBD(Component Based Software Development)로 구분하여 변경된 요구사항에 대한 효율적인 반영과

함께 웹 응용 시스템인 DEFACS에 적용시킨 방법에 대해 논의한다.

본 논문은 다음과 같이 구성된다. 제 2절에서는 본 논문의 대상이 되는 시스템을 제작하는데 필요한 개발 프로세스의 기법에 대하여 소개한다. 제 3절에서는 제 2절에서 언급한 기법을 변경한 개발 프로세스에 대해 제시한다. 제 4절에서는 실제 시스템에 적용시킨 사례를 제시하며 마지막으로 제 6절에서는 사례 적용과 추후 논의점으로 결론을 맺는다.

2. 개발 프로세스 기법들

소프트웨어 개발 프로세스는 기본적으로 폭포수 모델, 프로토타입 모델, 나선형 모델, 일정 중심 모델 등에서 시작하여 근래에 이르러 시스템의 발달과 콘텐츠의 발달, 사용자의 요구수준 증가 및 수요 증가 등의 이유로 소프트웨어 자체가 양적으로 질적으로 매우 커지게 되었다. 이러한 이유로 다양한 관점으로 프로세스 기법들이 제시되었으며 DEFA CS는 웹 응용 프로그램이고 지속적인 요구사항의 변경에 관점을 두었기 때문에 웹 개발 프로세스와 요구사항에 대한 핸들링을 위하여 컴포넌트 레벨로 프로세스를 관리하는 CBD에 대하여 알아본다.

2.1 웹 개발 프로세스

일반적인 소프트웨어 개발 프로세스와는 달리 웹 개발 프로세스는 웹의 특성상 개발기간이 짧고 자원이 한정되어 있다. 때문에 표현의 레벨보다는 목적과 기능에 대한 설계가 중요하며 화면에 대한 디자인이 선행되어야 한다. 웹사이트 개발 프로세스는 그림 1. 에서 보는바와 같이 크게 7가지의 업무로 나뉘는데 기획 단계에서는 요구사항에 대한 분석과 함께 서비스 흐름, 데이터 정의, 운영방안, 개발 범위 등이 정해진다. Web Page 기획단계에서는 전체 구성에 대한 세부 페이지들을 기획한다. 사이트맵,

관계도, 진행방법 등 기능에 대한 상세한 기획과 분석이 이루어진다. Web Page Design 단계에서는 Web Page 기획 단계에서 나온 산출물들을 기준으로 Web Page 디자인 작업을 수행한다.

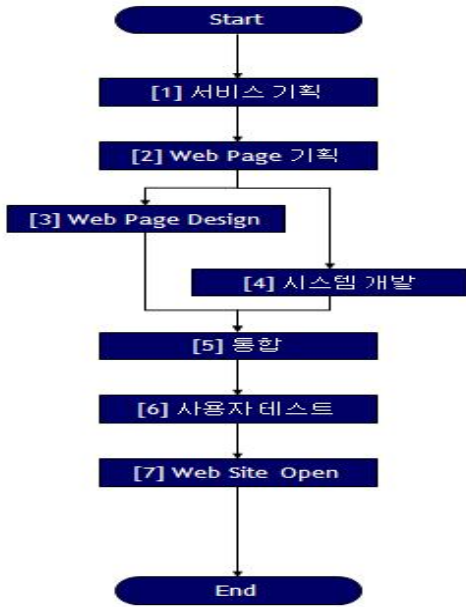


그림 1. 웹사이트 개발 프로세스[1]

디자인에 대한 규칙 정의와 함께 GUI 구성을 한다. 시스템 개발 단계에서는 앞서 기획을 기준으로 전체 시스템에 대한 분석, 설계를 수행한다. 시스템 개발 규칙 정의와 함께 프로세스 및 데이터베이스 설계가 포함된다. 통합과정은 디자인과 개발된 시스템을 기능을 기준으로 통합하는 과정이다. 통합 과정이 완료되면 사용자테스트를 통하여 오류검출과 사용자레벨에서 수행을 한다. 모든 과정이 끝나면 최종적으로 웹사이트 오픈을 하는데 DEFACS와 같은 경우는 내부 시스템이고 사업 특성상 요구사항이 지속적으로 변하기 때문에 이와 같은 프로세스를 그대로 적용키 어렵다. 또한 서비스 대상 웹사이트가 아니고 데이터에 대한 입출력, 관리, 제어, 응용 등이 목적이기 때문에 일괄적인 개발보다는 요구사항에 맞추어 단위별로 개발하여야 한다.

2.2 CBD(Component Based Development)

CBD(Component Based Development)는 재사용(reuse)과 조립(assembly)이 가능한 소프트웨어 자산을 컴포넌트라는 것으로 만들고 그것들을 기반으로 시스템을 개발하여 개발의 생산성 향상과 비용의 절감을 추구하고자 하는 개발방식의 Trend이다. 그리고, 그러한 CBD 개념을 적용하기 위한 역할(role), 작업(task), 산출물(work product) 등을 체계적으로 정리해 놓은 것이 CBD방법론이다.

CBD는 방법론이기 때문에 실질적으로 적용키 위한 여러 기법들이 존재한다. Catalysis[2]는 1992년 Desmond D'Souza와 Allan Wills가 개발한 CBD 방법론인데 UML과 UML 확장 메커니즘을 적용하여 모델을 표현한다. 이론적인 접근법을 제시하고 있으나 학문적이어서 쉽게 적용하기 어려운 면이 있고, 프로세스에 대한 정립이 필요한 단점이 존재한다. Advisor (SBD96)[2]는 John Dodd가 Catalysis를 채택하여 제공한 CBD 방법론이다. 컴포넌트를 명세(specification)와 구현(implementation), 패키지(package)로 분리하는 개념을 적용하여 UML과 UML 확장 메커니즘으로 표현하였으며 개발 프로세스의 트윈 트랙을 정의하였다. 어플리케이션 개발 트랙과 컴포넌트 공급 트랙으로 분리하여 컴포넌트를 조립하여 어플리케이션을 개발하는 프로세스와 컴포넌트를 개발하는 프로세스를 별도로 정의하여야 한다. Select Perspective[2]는 Stuart Frost와 Paul Allen이 제안한 CBD 방법론이다. 컴포넌트 모델은 UML로 표현하였으며 정렬/설계/조립의 기본 단계를 반복적·점진적으로 수행하는 프로세스를 정의하였다. LUCID 기법을 통해 실용적이고 실질적인 프로세스를 제공한다. RUP[2]는 Booch, Rumbaugh, Jacobson이 제안한 방법론이며 초기에는 객체지향 개발 방법론으로 소개되어 지금은 컴포넌트 개발 방법론으로 진화하였다. UML로 모델을 표현하고 개발 프로세스와 관리 프로세스를 통합하여 프로세스를 정의하였다. 반복적·점진적인 개발을 강조하였으며 각 반복을

계획하고 수행하는 지침들까지 포함하고 있다. UML Component[2]는 2000년 John Cheesman과 John Daniels가 컴포넌트 시스템의 아키텍처와 의존성을 명세화하는 기법을 제시하였다. UML을 이용하여 컴포넌트를 모델링하는 기법이 구체적으로 소개되어 있어 실질적인 컴포넌트 모델링 기법을 제시하며 현재 널리 쓰이고 있는 방법중 하나이다. Korbra[2]은 Atkinson 외의 다수의 연구진이 컴포넌트 기반의 Product-Line Engineering을 실현하는 방법과 기법, 도구 등을 정의하였다. UML로 컴포넌트 모델을 표현하였으며 Product-Line 개념을 적용하여 컴포넌트를 작성하고 유지하며, 배치하는 전략을 소개하였다. 본 논문에서는 이러한 방법론들에 대하여 적용을 하였던 바, 개발의 포커스가 이미 정립된 요구사항에 대한 컴포넌트의 활용이 아니기 때문에 기본적인 CBD의 개념을 수정하여 사용하였다.

3. CD와 CBD

본 절에서는 서론에서도 밝혔듯이 DEFACS는 사업의 진행에 따라 요구사항이 수시로 변하는 바, CDB를 CD와 CBSD로 나누어 적용하였다. 그림 2.는 분리된 프로세스에 대한 정보를 나타낸다.

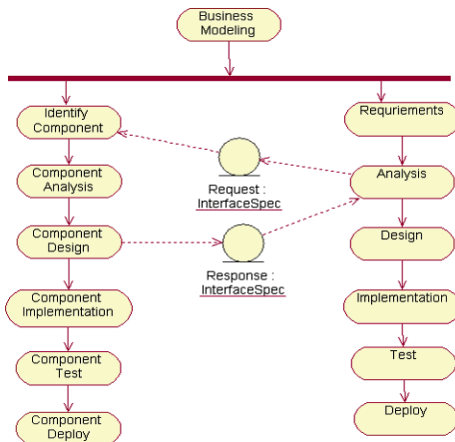


그림 2. CD와 CBSD 프로세스

그림에서 왼쪽은 컴포넌트를 개발하는 과정(CD)을 나타낸 것이고 오른쪽은 컴포넌트를 사용하는 시스템을 개발하는 과정(CBSD)을 나타낸 것이다.

CBSD에서 CD쪽으로 연결되어 흘러가는 모습은 기존의 컴포넌트가 존재하는 형태가 아니라 CBSD를 추진하면서 새로운 컴포넌트를 개발하고자 하는 CBD의 초기단계 모습이 된다.

CD쪽의 Requirements과정은 개발하고자 하는 컴포넌트의 인터페이스를 정의하는 것과 동일한 작업이 되고, 이미 개발이 되었거나 개발하고자 결정된 인터페이스의 내용에 대해서 CBSD쪽으로 정보가 전달되어야 CBSD쪽에서는 원하는 컴포넌트의 정보를 얻을 수가 있다. 이 역할을 Interface Specification이 맡아서 두 프로세스의 교량 역할을 하게 된다. CBD에서 컴포넌트 개발을 위한 첫 단계는 우선 무엇을 컴포넌트로 개발할 것인가를 정하게 되는 컴포넌트 정의단계인데 이것은 어떠한 방법론을 전개할 것인가에 따라 그 시점이 다르게 된다. Domain의 Entity정보를 중심으로 컴포넌트를 개발하고자 하는 방법론에서는 Requirements 단계에서 부터 컴포넌트를 결정할 수 있고, 요구사항에 대한 분석을 거쳐 분석 결과물을 검토하면서 컴포넌트를 정의하는 방법론을 다르게 되면 분석 단계 이후가 된다. 그래서 위의 그림에서 표현되는 Interface Specification의 Request 시점과 Response 시점은 여러 프로세스에서 발생할 수가 있다. 프로젝트 조직의 환경과 채택한 방법론의 내용에 따라 그 시점이 조정되어야 한다. DEFACS는 요구사항이 추가되는 시점에서 CBSD가 전개되면서 CD쪽으로 갔다가 다시 오는 흐름이 전개시킴으로써 CBD의 프로세스가 제대로 동작시킨다.

4. DEFACS 적용

위에서 언급했듯이 Interface Specification의 Request 시점과 Response 시점은 프로젝트와 방법

론에 따라 달라진다. CBD의 방법론들은 UML의 내용이 달라질 뿐 UML을 사용하는 것은 동일한데, DEFACS는 기본적으로 UML Component의 방법을 따른다. DEFACS의 개발 방식은 시스템의 갖춰야 할 가장 낮은 레벨의 기본적인 컴포넌트들을 구성 후 요구사항에 따라서 수정하는 방식을 취한다.

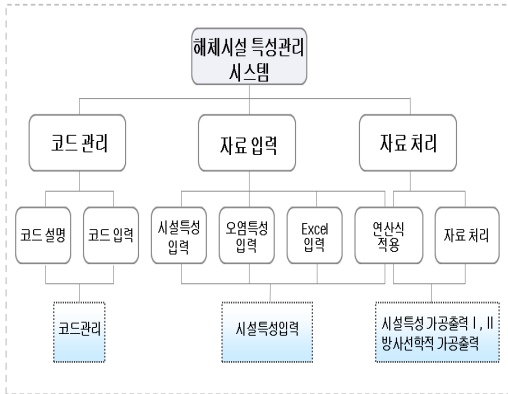


그림 3. DEFACS 기능 구조도

그림 3.은 DEFACS의 구조를 나타낸 것인데 이중 Code definition과 Data Processing의 요구사항이 계속 바뀌게 되었다. 이를 해결하기 위해 중점으로 둔 것이 컴포넌트의 독립성과 전체 프로세스에 대한 영향력을 최소화 하는 것이다. DEFACS는 시설에 대한 방대한 양의 데이터를 근간으로 구동되는 시스템이기 때문에 데이터베이스가 가장 큰 부분을 차지하며 시설의 무게나 부피, 선량 등과 같이 기능에 종속적으로 관련된 데이터가 수정된 경우에는 전체 데이터에 대한 부분이 수정이 되어야 한다. 이는 전체 CBSD에 영향을 주기 때문에 이를 해결하기 위하여 각각의 CD안에서 또다시 CD와 CBSD로 구분하여 최저레벨의 CD의 처리를 최우선시 함으로써 재귀적으로 처리되게 하였다. 이러한 방식을 위하여 더미데이터베이스를 사용하였다.

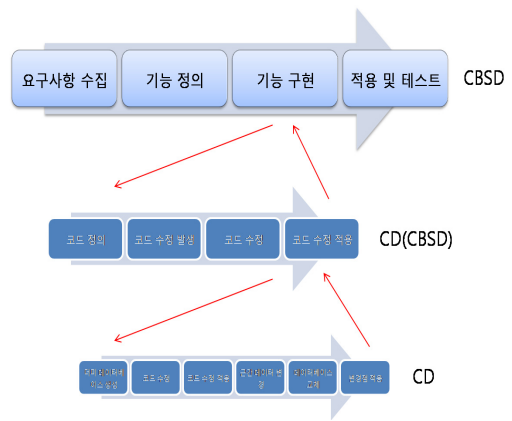


그림 4. 재귀 스케줄링 방식

그림 4.를 보면 기본적인 CBSD와 CD외에 CD에서 다시 CBSD와 CD로 구분하는 방식이 나타나 있다. CBSD는 전체적인 구성만 되어 있고 기능이 필요한 컴포넌트들을 전부 CD로 구성한다. 이 때 CD를 구성하는 요건이 변경되게 되면 이 때 CD를 CBSD로 간주하고 이를 처리하기 위한 컴포넌트를 CD로 생성하여 처리하게 된다. DEFACS에서 변경점이 발생하는 CD는 앞에서 언급했듯이 Code definition과 Data Processing인데 Code definition을 예로 들어 설명한다.

Code definition의 경우 크게 코드를 정의하고 생성하는 과정이라고 할 수 있는데 코드 정의 시 일반적인 코드를 작성 후에 해체사업 과정에 따라 코드가 변경되는 상황이 발생한다. 이 때 모든 코드가 한번에 변경되는 것이 아니고 위에 언급했듯이 코드 별로 종속성이 발생하기 때문에 하나의 CD로써 처리하기가 어려워진다. 그래서 Code definition이라는 CD를 다시 CBSD로 간주하고 변경되는 코드에 따라 수정을 해줘야한다. 이때 필요한 것이 더미데이터베이스이며 이를 Code definition에 대한 CD가 된다. 그림 4.에서 코드의 수정이 발생하게 되면 이에 해당하는 코드와 그에 종속된 데이터에 대한 더미데이터베이스를 생성한다. 이때 생성되는 데이터베이스는 기존의 데이터베이스의

미러 형태가 아닌 각 코드와 그에 종속된 데이터가 된다. 이 후 변경 점이 수정되면 이를 기존의 데이터베이스에 적용시키는 방식이다. 해당 방식은 기존 데이터베이스의 ACID를 만족시키는 동시에 Concurrency control을 가능하게 해준다.

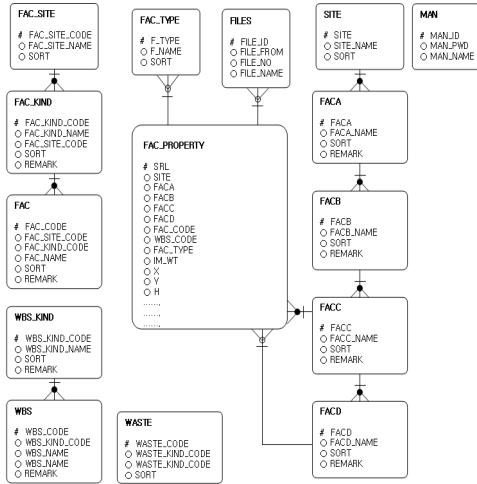


그림 5. 기반 ERD

코드가 한번에 여러 개가 바뀌는 경우에도 더미 데이터베이스로 인해서 독립성이 유지되므로 이를 처리하기 위한 개별적인 비용에만 영향을 주게 되어 전체 프로세스 처리에는 영향을 최소화 할 수 있게 한다. 또한 변경이 되는 코드들에 대한 확인 작업이 쉬워지고 전체 개발 프로세스의 진행이 수월해진다. 이에 대한 기반이 되는 UML은 그림 5.에 나타나있다.

5. 결 론

본 논문에서는 국내에서 처음 시도된 원자력 시설의 해체 사업에 따른 해체정보관리 시스템(DEFACS를 개발하는데 있어서 일반적인 개발 프로세스를 적용시킬 수 없으나, 객체지향 컴포넌트 기반 방법인 CBD를 웹 개발 프로세스와 사업 특성에 맞추어

개발하였다. 요구사항의 변경으로 인한 컴포넌트의 미완성을 CD와 CBSD의 프로세스로 분리하고 변경 점이 잦은 컴포넌트에 대한 부분을 CBSD로 간주하여 그에 대한 처리를 CD로 분류하여 재귀적으로 처리되게 하였다. 이를 위하여 더미데이터베이스를 사용하였으며 이는 전체 프로세스에 주는 영향을 최소화하고 동일 컴포넌트에 대한 변경 점이 자주 발생하여도 효율적으로 처리됨을 확인할 수 있었다. DEFACS는 현재에도 계속 데이터 입력 및 업그레이드가 이루어지고 있으며, 본 시스템을 이용하여 도출된 정보 및 자료는 국내에서는 최초로 수행된 원자력시설 해체사업의 평가 자료로 활용중에 있다. 본 시스템의 활용은 그림6, 그림7에 나타나있다.

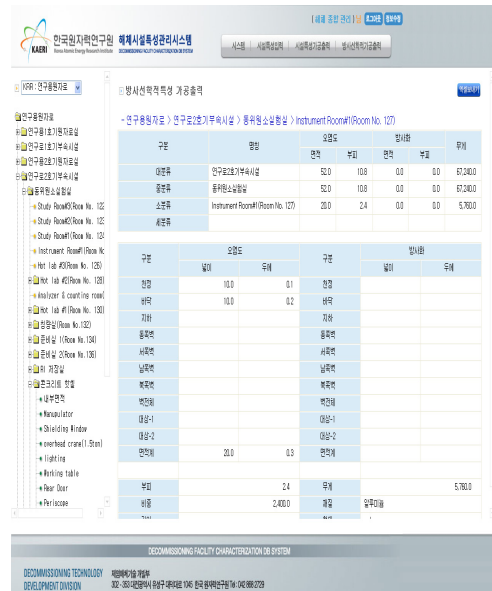


그림 6. DEFACS 사용자 인터페이스

이와 같은 시스템을 통하여 도출된 자료들은 추후 타 원자력시설의 해체 계획 수립 및 설계의 기초자료로 활용되며 본 시스템의 개발 방식은 추후 현재 개발중인 후속 시스템인 정보/자료 평가 시스템과 모델링 시스템에도 동일하게 적용시켜 개발 중에 있다.



최 윤 동

현재 한국원자력연구원 근무



문 제 권

현재 한국원자력연구원 근무