

# 고속전철 시스템엔지니어링 전산체계 구축

## Design of System Engineering Database for KHSR

황희수\*\*, 이태형\*, 전용렬\*, 현승호\*\*, 정홍채\*\*\*

Hwang, Hee-Soo Lee, Tae-Hyung Jeon, Yong-Ryul Hyun, Seung-Ho Chung, Heung-Chai

---

### ABSTRACT

This paper will describe the design of database for system engineering of High Speed Railway Systems, focusing on the ability to capture a full thread of traceability, from the inception of a system requirement, to the test results that validate that the designed system meets criteria defined by the requirement. The database includes data models for requirement, function, component, work, work process, resource, etc.. These data models are utilized for the efficient management of data related system design, development work, and validation.

---

### 1. 서론

복잡한 시스템 개발에 있어서 시스템엔지니어에게 가장 중요한 것은 최종적으로 개발된 시스템이 이해 당사자들과 소비자의 요구를 만족시키고 시스템 개발이 시행착오 없이 진행하기 위하여 개발초기에 기술적인 위험부담 요소들을 식별하여 대처하는 것이다. 시스템의 요건은 소비자의 요구인 동시에 최종 개발된 시스템이 성공적으로 개발되었나를 평가하는 기준이 된다. 따라서, 시스템개발에 있어서 요건관리는 시스템엔지니어의 중요한 임무이다. 요건관리는 요건에 관한 정보의 수집, 관리와 보급으로 구성되며, 그 핵심은 요건들을 추적, 관리하는데 있다. 요건의 추적성을 통해 요건 분석 단계로부터 시작, 시스템 설계와 실행단계에서 새롭게 도출되거나 변화되는 요건을 효과적으로 관리하여 개발되는 시스템의 성공확률을 높일 수 있다. 최근 들어 무기체계, 통신, 운송, 우주항공 등의 복잡한 시스템 개발 시 수작업을 통하여 수행하던 요건관리를 그 복잡도가 날로 증가함에 따라 자동화된 전산도구를 사용하여 할 수 있도록 많은 도구들이 개발되고 실제로 적용되어 왔다.

고속철도 시스템은 차량, 기계부품, 전기, 전자, 제어, 정보통신, 토목기술 등이 종합적으로 적용되는 대형 복합 시스템의 하나로써, 시스템의 개발 요건이 체계적으로 관리되어야만 하는 전형적인 예라고 할 수 있다. 요건은 하위 계약자와의 계약조건에 반영될 뿐만 아니라 과제 수행근거가 되며 시스템의 통합 검증의 근거로서 모든 개발단계의 기본사양이 된다. 고속전철 시스템 기술

---

\* 한국철도기술연구원 고속철도기술개발사업단 시스템개발팀 주임연구원 정희원

\*\* 한국철도기술연구원 고속철도기술개발사업단 시스템개발팀 선임연구원 정희원

\*\*\* 한국철도기술연구원 고속철도기술개발사업단 시스템개발팀 책임연구원, 정희원

개발 과제 자체가 크고 복잡하기 때문에 많은 하위 계약자(또는 과제)가 존재한다. 이러한 하위 과제에 적절히 시스템 요건과 작업을 분배하고 그 결과를 평가 검증할 수 있는 체계가 갖추어져야만 한다. 이를 통하여 전체 시스템의 통합 시 생길 수 있는 문제들을 사전에 고려하여 각 서브 시스템에 반영할 수 있으며, 모든 하위 개발자들이 일관된 시스템 요건과 작업 목표를 가지고 일에 임할 수 있는 기초가 된다. 고속철도는 시스템의 요건, 시스템의 구조, 기능들이 복잡하며, 시스템 요건과 하위 과제의 연계관리도 상당히 복잡하기 때문에 적절한 시스템엔지니어링 도구의 사용 없이는 추적관리에 드는 인적비용 및 시간이 만만치 않으며 오류가 생기기 쉽고 매우 비효율적이다. 따라서, 시스템 요건에 정의된 목표 성능을 달성하고 시스템간의 불일치를 최소화 하기 위해 체계적인 시스템 통합 절차를 제공할 수 있는 시스템엔지니어링 체계를 구축하고, 이를 데이터베이스화하여 종합적으로 연계관리를 할 수 있는 전산 시스템을 구축하는 것이 본 연구의 목표이며, 이를 위해 Ascent Logic사의 RDD-100이란 시스템엔지니어링 전산 도구를 사용하였다.

## 2. 시스템 엔지니어링 데이터베이스 설계

연계 관리를 위해 구축할 시스템 엔지니어링 전산 체계 데이터베이스에는 요건, 인터페이스, 검증, 시스템 구조, 작업 구조, 자원, 조직 등에 대한 데이터를 데이터 모형 형태로 입력한다. 데이터 모형은 크게 시스템 설계 부분, 시스템 검증 부분과 시스템엔지니어링 관리 부분으로 구분할 수 있다. 그림1은 시스템 엔지니어링 전산 체계에 사용될 데이터 모형에 대한 개요를 보여주는 그림이다.

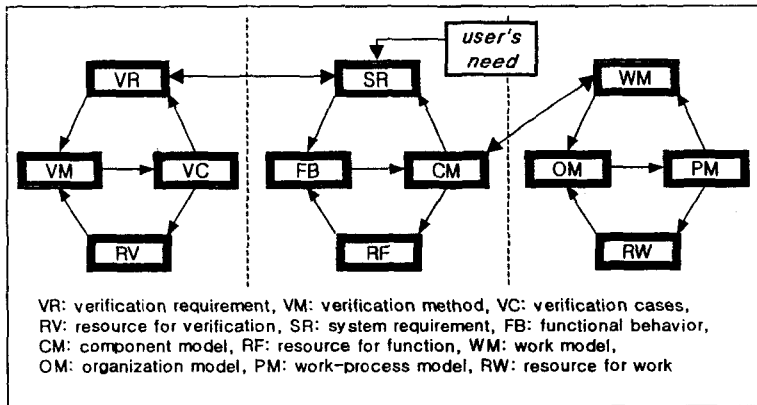


그림1. 시스템 엔지니어링 전산 체계 데이터 모델

이 그림에서 가운데는 시스템 자체에 대한 데이터 모형이고 왼쪽은 시스템 검증을 위한 데이터 모형, 오른쪽은 시스템엔지니어링 관리를 위한 데이터 모형이다. 시스템 모형은 시스템 요건(SR), 시스템 거동(FB), 시스템 구조(CM), 부가적으로 시스템의 기능이 수행될 때 소모되거나 생산되는 자원(RF) 모형으로 나눌 수 있다. 검증 관리를 위한 모형에는 검증 요건(VR), 검증 방법(VM)와 검증 사례(VC)로 구성된다. 시스템엔지니어링 관리를 위한 데이터 모형은 작업 모형(WM), 작업을 수행하는 조직(OM), 작업 일정(PM)과 작업에 필요한 자원(RW) 모형으로 구분된다. 각각의 모형은 다른 모형과 상호 연관 관계를 가지고 있으며, 시스템 설계 데이터 모형은 차후 이 시스템과 관련된 상세 설계, 개발, 통합, 검증, 유지보수 등 시스템 생애주기 동안 발생하는 모든 작업의 기초로 사용된다. 이 데이터 모델에 따라 저장된 데이터는 고속철도 시스템엔지니어링을 위한 데이터베이스를 구성하며, 개별 데이터들의 연계성을 통해 관련 항목으로의 추적, 관리가 가능하게 된다. 따라서, 이렇게 구축된 시스템엔지니어링 전산 체계는 "시스템엔지니어링에 관한 모든 지식의 축적"이란 목적 이외에 향후 이를 재사용하여 기술의 전수 및 기술 수출을 위한 기본 자료로써도

그 활용 가치가 높다. 이러한 목적을 달성하기 위해서는 RDD-100의 자료 저장 구조에 적합한 데이터 모형을 설정하고 이들의 연계성을 정의하는 것이 중요하다.

## 2.1 시스템 요건 모형

시스템 요건(데이터) 모형은 기본적으로 트리 구조를 가진다. 사용자와 고객으로부터 정리된 최상위 요건으로부터 시스템 상세 설계자가 이해할 수 있고 구현할 수 있는 시스템 수준에서의 요건을 시스템 요건이라고 한다. 시스템 요건은 최상위 요건으로부터 하위 요건으로 계층적으로 분해되며, 분해된 하위 요건들은 분해하기 전 상위 요건으로의 관계만을 가지기 때문에 다른 가지에서 파생된 시스템 요건과는 일반적으로 관계를 가지지 않아야 한다. 시스템 요건 모형의 말단 노드에 있는 요건은 상세 설계자(하드웨어/소프트웨어 설계자)가 이해할 수 있는 수준이어야 하며, 명확해야 하고 검증이 가능해야 한다. 전통적인 시스템엔지니어링에서, 시스템 요건은 사용자의 필요를 만족시키기 위해 시스템이 가져야 할 기능을 주로 다루며, 시스템의 운영자(사용자)와 시스템간의 상호작용에 중점을 두어 분해된다. 이러한 요건들을 '기능적 요건'이라고 부르며 다시 세부적인 요건으로 분해된다. 요건의 분해는 하향식의 계층적 방법에 의해 이루어지면서 점차 확실해지고 구체화된다. 이는 시스템 거동 모델을 도출해가는 과정에서 더 상세히 분해되기도 하고, 빠진 요건을 추가하기도 하면서 더 정확해 진다. 시스템 요건 모형에서 각 요건은 시스템 거동 모델에서 하나의 기능 모델로 반영된다. 본 연구에서는 시스템 기본 사양과 개발 보고서를 기준으로 요건을 분석한 후, 통합 시스템 수준에서 요건을 관리할 수 있도록 하는 체계를 구축하였다.

## 2.2 시스템 거동 모형

요건을 확인하는 하나의 방법은 고객이 원하는 여러 가지 시나리오를 설정해서 고객과 시스템 설계자가 상호 이해를 바탕으로 시스템 기능을 확인해 가는 방법론이 있다. 이를 '시나리오에 기반한 설계'라고 하며 시스템의 요건을 도출하는 유용한 방법으로 사용되고 있다. RDD-100에서도 OBSE(Object-Based Systems Engineering)이란 유사한 방법을 사용하고 있다. 요건은 이 시나리오를 모델링하는 과정에서 추가되거나 그 의미가 명확해지며, 전체적으로 정확하고 완전한 시스템 요건을 도출해 낼 수 있다. 이를 위한 시스템 거동 데이터 모형은 RDD-100의 Behavior Diagram(BD)에 의해 만들어진다. BD는 이들 기능이 시나리오에 따라 어떤 순서로 수행되는지, 그리고 어떤 입력과 출력을 가지고 있는지를 기술하기 위한 방법론이자 도구이다. RDD-100에서 시스템 거동 모형은 DVF(Dynamic Verification Facility)라는 시뮬레이션 엔진을 통해서 실행될 수 있으며, 이를 통해 시스템 기능의 완전성 여부를 판정할 수 있다.

## 2.3 시스템 구조 모형

시스템 구조 모형은 시스템의 개념적인 구조에 관한 것으로 RDD-100의 컴포넌트 모델링 방법에 해당한다. 컴포넌트 모델은 크게 시스템을 구성하는 컴포넌트의 계층 구조와, 컴포넌트간에 정보를 주고받는 링크 모델로 이루어져 있다. 링크는 단순히 메시지를 교환하는 통로일 뿐만 아니라 컴포넌트들을 이어주는 물리적인 연결 고리가 될 수도 있다. 하나의 컴포넌트는 물리적으로 구분된 시스템 내의 구성요소를 의미할 수도 있지만, 물리적으로 분리되어 있어도 시스템의 어떤 기능을 담당하는 다수의 물리적인 요소들을 통칭하는 논리적인 것을 의미하기도 한다. 따라서, 일반적으로 컴포넌트 모델은 시스템 구조라 부르기도 한다. 하나의 컴포넌트는 하나 이상의 기능을 시스템 거동 모형으로부터 할당받아 이를 수행한다. 또한 시스템의 비기능적 요건, 예를 들어서 시스템의 성능, 신뢰성, 가용성, 치수 등은 해당 컴포넌트에 연결되며 컴포넌트에 대한 설계 사양으로 작용한다. 본 과제에서는 고속전철 시스템의 구조에 대한 설계가 일정 부분 끝났기 때문에 시스템의 물리적인 구조인 PBS(Product Breakdown Structure)를 기초로 하여 시스템 구조(컴포넌트 모형)를 작성하였다. 시스템의 거동 모델과의 관계는 아직 설정되지 않았지만, 시스템 요건 모

형과는 연계되어 있어서 어떤 요건이 어떤 컴포넌트(PBS내의)에 적용되는지, 어떤 컴포넌트에 관계되는 요건에는 어떤 것이 있는지를 추적, 관리할 수 있다.

## 2.4 자원 모형

자원 모형은 시스템의 동적 거동과 관련하여 소모되거나 생산되는 자원을 모형화 한 것이다. 하나의 자원은 초기 값을 가지고 있으며, 시스템의 기능을 수행하는 동안 소모되거나 생산되는 것을 시스템 자원에 더하거나 빼는 간단한 구조이다. RDD-100의 DVF를 통하여 시스템 거동 모형을 시뮬레이션 할 때, 하나의 기능이 수행될 때마다 자원의 현재 값이 변하며, 만일 자원이 고갈 되면 시뮬레이션은 멈추게 되고 이를 알려준다.

## 2.5 작업 모형

작업 모형으로는 WBS(Work Breakdown Structure) 모형을 사용하며, 설계된 시스템 사양으로부터 실제 시스템의 설계, 구현, 시험검증 등 시스템과 관련하여 해야 할 일들을 계층적으로 분해한 트리 구조의 작업도이다. 작업모형에 관련된 작업들은 패키지로 묶어서 관리되고 이는 하나의 하위과제로 할당된다. 계층 구조를 가지고 있기 때문에 전체 시스템 과제는 작업 단계별로 여러 개의 하위 과제들에 의해 수행될 WBS상의 작업들이 성공적으로 이루어져야 하나의 작업 단계가 끝나게 된다. WBS는 시스템 모형과 직접적인 연관관계를 가진다. 과제에서 수행되는 어떤 작업도 시스템 사양과 관련하여 반드시 연관 관계를 가져야 한다. WBS는 기본적으로 시스템 구조 모형 및 시스템 요건과 연관관계를 가져야 한다. 궁극적으로 WBS는 시스템의 요건을 모두 만족하기 위해서 수행되어야 한다.

## 2.6 조직 모형

작업을 수행하기 위해서는 이를 수행하는 작업 주체가 반드시 존재해야 한다. 조직 모형은 바로 이에 대한 모형이다. 여러 작업 주체들이 서로 상호 작용하면서 작업을 수행하기 때문에 이는 시스템 구조 모형과 대응관계를 가진다.

## 2.7 작업 공정 모형

작업 공정 모형은 작업모형에서 정의된 작업의 선후 관계, 작업간에 산출물들의 입출력 관계, 각 작업에서 걸리는 시간, 각 작업의 세부적인 작업순서 등을 모델링한다. 이러한 과정에서 WBS에서 빠진 작업을 추가하거나 보완하기도 한다. DVF를 통한 시뮬레이션을 통하여 전체 작업이 수행되는 기간, 하나의 작업이 길어졌을 때 전체적으로 작업이 얼마나 지연되는지 등을 확인할 수 있으며, 자원모형과 관련하여 주어진 자원이 적절하게 사용되는지 등을 미리 확인할 수 있다.

## 2.8 인터페이스 관리 모형

인터페이스 분석을 통해 주요 관리 항목을 도출하고, 개별 과제의 과업 진척에 대한 정보를 수집하여 인터페이스 관련 문제점을 조기 도출하며, 도출된 문제를 인터페이스 협의체에서 해결하며 그 과정에서 발생한 사항을 관리하기 위해 인터페이스 관리 모형을 만든다. 이 모형은 시스템 요건 모형에 인터페이스 요건으로 추가되며, 각 요건은 관련 설계 사양, 컴포넌트, 검증 요건 및 검증 결과 등에 연계되어 추적, 관리될 수 있도록 한다.

## 2.9 검증 관리 모형

검증 관리는 시스템 요건을 검증하기 위해 검증 요건의 수립, 검증 방법의 지정 및 검증 결과로 구성된다. 이들은 시스템 요건, 시스템 구조 등에 연계되어서, 시스템 요건으로부터 검증 결과에 이르기까지 추적, 관리할 수 있도록 한다. 현재에는 검증 관리를 위한 데이터베이스 모형에 대

한 설계만이 진행되었지만, 향후 검증 요건, 검증 방법 등을 DB화하고 검증 결과 등에 연계성을 갖도록 관리 체계를 구축할 것이다. 검증 요건을 검증하는 방법에는 크게 시험평가, 시뮬레이션, 해석, 검사와 데모가 있다. 이 가운데 시험평가 방법에 의한 검증을 관리하기 위해서는 다음과 같은 일이 이루어져야 한다.

- 입증해야 할 요구조건 형태에 따라 분류된 시험항목 트리를 구성.
- 시험항목을 요구조건, 기능분해도, PBS 및 WBS에 연계
- 소스문서에서 요구조건, 기능, PBS, WBS 및 시험항목까지를 연계해서 추적, 관리.

### 3. 시스템 설계 관련 데이터 관리

산발적으로 흩어져 있고 연계성을 갖고 있지 못한 데이터를 RDD-100 DB형태로 통합 정리하여 상호 연계성을 갖고 추적, 관리할 수 있도록 하는데 시스템엔지니어링 관리 체계 구축의 목적이 있다. 시스템엔지니어링은 크게 '시스템 수준의 설계', 설계된 시스템 사양서로부터 시스템을 만드는 전 과정을 관리하는 '시스템엔지니어링 관리'와 설계된 시스템을 검증하는 검증 관리로 나누어 생각할 수 있다. 그 중 시스템 설계 데이터는 시스템이 반드시 만족해야 할 시스템 요건 모형과 시스템 요건 중 기능적 요건을 프로세스 모델로 표현한 시스템 거동 모형, 시스템의 요건에서 기술된 기능을 수행하기 위해 구체적으로 설계되어야 하는 개념적 시스템 컴포넌트(구조) 모형으로 크게 나뉘며 부가적으로 시스템의 컴포넌트가 기능을 수행할 때 소모되는 자원 모형으로 나뉘어 생각할 수 있다. 시스템 설계 데이터는 기본적으로 고객과 사용자의 필요를 구체화하여 시스템 요건을 도출하는 과정으로부터 시작된다. 시스템 요건 도출은 그림4와 같은 과정을 거치면서 더 이상 분해할 수 없고 명확하게 정의될 때까지 구체화 된다. 이 때 이루어진 모든 의사 결정 내용도 함께 추적 관리된다.

구체화된 기능적 요건으로부터 여러 가지 시나리오를 만들어 가면서 시스템의 기능적인 요건을 시스템 거동 모델로 정형화시키는 과정이 이어서 진행된다. 이 과정에서 누락되었거나 덜 분해된 시스템 요건을 보완하게 된다. 이러한 과정을 거친 다음 시스템의 요건과 시나리오, 즉 거동 모형이 완성되면 거동 모형의 각 기능을 담당할 컴포넌트를 생성하고, 컴포넌트 구조를 설정한 다음 각 기능을 할당하는 구조 설계 단계를 수행한다. 또한, 시스템 요건으로부터 시스템 구조 모형의 컴포넌트가 가져야 할 성능과 같은 비기능적 요건을 할당하기도 하고 역으로 요건을 구체화하기도 하면서 시스템 요건 및 구조 모형을 보완한다. 이러한 과정을 통해서 최종적으로 상세 설계자에게 넘길 시스템 사양이 결정되며, 상세 설계자는 시스템 구조 모형을 중심으로 작업을 수행하는 것이 일반적이다.

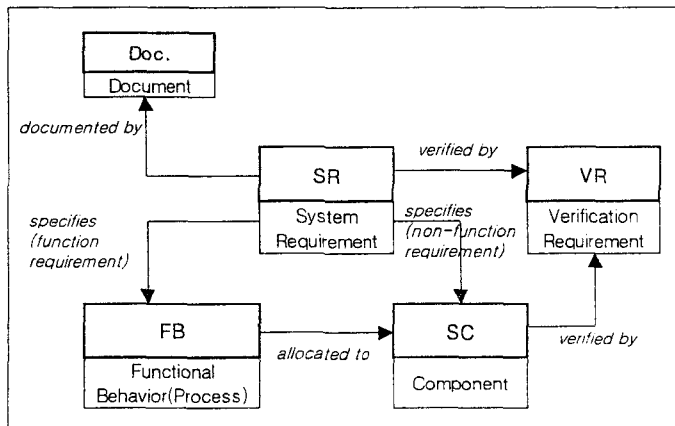


그림2. RDD-100내에서 시스템 설계 관련 데이터의 관리 체계

그림2는 시스템 설계와 관련한 데이터 관리 체계를 위해 RDD-100에서 사용한 개체 유형과 이들을 연계할 관계 유형이 보여진다. 이 그림에서 설계 관련 데이터는 RDD-100의 스키마 가운데 적절한 개체 유형(Element Type)을 활용하여 추적, 관리한다. 시스템 설계 공정에서 요건 분석에서 발생하는 데이터는 요건 개체 유형을 통해 입력되고 분해된다. 기능 분석과 관련해서는 RDD-100의 Behavior Diagram을 이용하며, 컴포넌트 구조 설계는 Component Diagram을 이용한다. 이들 사이의 상호 연계성은 각 개체간에 잘 정의되어 있는 링크(Relation Type)들에 의해 정의된다.

#### 4. 개발 과제 관련 데이터 관리

과제 관련 데이터 관리는 시스템 사양으로부터 이를 만족하는 시스템을 획득하는데 관련된 모든 설계, 구현, 검증, 운용과 관련된 작업을 정의하는 일로부터 시작된다. 시스템의 성격에 따라 이 프로세스는 다를 수 있다. IEEE 시스템엔지니어링 표준에서도 일반적인 프로세스를 제시하고 있지만, 반드시 표준에 제시된 것을 따르라는 것은 아니며 특성에 맞게 재단해서 사용할 것을 권고하고 있다. WBS가 완성되면 이를 어떤 프로세스로 수행할 것 인지가 먼저 분석되어야 한다. 이러한 작업공정 모형이 대략적으로 완성되면 WBS를 어떻게 분할해서 과제화 할 것 인지와 각각에 투입될 자원을 결정하게 된다. 마지막으로 과제 수행의 주체가 결정되면 작업을 수행을 위한 과제 체계가 완성된다. RDD-100을 이용하여 과제 관리 모형을 만들고 이를 관리하는 과정은 시스템 설계 경우와 유사하다. 과제에서 수행할 WBS는 별도의 WBS란 개체 유형을 만들어서 입력하며, 계층적인 구조를 가질 수 있다. WBS의 작업순서/작업일정은 RDD-100의 Behavior Diagram을 이용하여 관리한다. 작업을 수행할 주체인 조직은 Component 개체 유형을 이용하되, component\_type이란 속성을 Organization으로 지정한다. 규모가 큰 작업은 과제 단위로 만들어져 수행되므로 과제에 대한 정보를 담기 위하여 ProjectInfo라는 유형을 만들어서 관리하도록 한다. 시스템 설계 관련 데이터와 연관은 WBS를 통해 work for/worked with라는 관계 유형에 의해 이루어지기 때문에 각 작업에 연관된 시스템 요건이나 컴포넌트에 대한 추적성을 갖는다. 이런 추적, 관리를 통해 작업의 정당성 또는 작업의 목표를 쉽게 도출할 수 있다. 그림3에 여기서 설명한 시스템엔지니어링 관리 체계를 위한 RDD-100의 개체 유형과 각 개체의 연계를 위해 사용된 관계 유형이 보여진다.

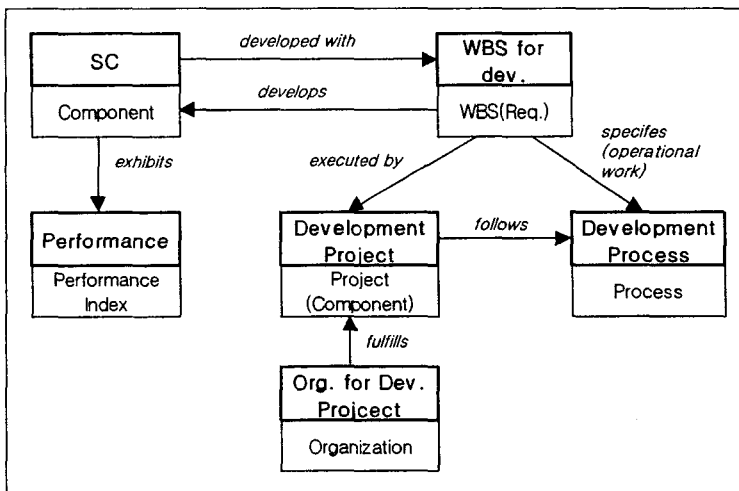


그림3. RDD-100내에서 시스템엔지니어링 관리를 위한 체계

## 5. 검증 관련 데이터 관리

요건의 생성에서부터 요건에 의해 정의된 기준을 만족시키도록 설계된 시스템을 검증(validation)하는 검증 결과까지를 연계해서 추적, 관리할 수 있는 방안을 설명한다.

### 5.1 기존의 방법

과거에는 전후 참조 검증 표에 의해 요건과 이에 관련된 검증 요건을 상호 연계하는 방법을 사용하였다. 그림4와 같은 전후 참조 검증 표에는 검증 요건에 의해 사용될 방법(검사, 데모, 시험, 시뮬레이션, 해석 등)을 포함한다. 이런 표에 의한 방법은 상당히 많은 반복적인 행위를 필요로 하기 때문에 소모적이다. 즉, 시험 평가 계획 등에서 정의되는 검증 요건을 설계 사양에 연계하는 작업을 사람이 직접 해야 하며, 요건이 변하면 이런 작업은 반복되어야 한다. 또한 이렇게 완성된 것도 검증 내용과 검증 방법 이외의 것은 추적, 관리할 수 없다. 검증 방법과 검증 결과 등을 연계하기 위해서는 추가적인 노력이 필요하게 된다. 따라서, 시스템 엔지니어는 대단히 많은 문서와 반복적인 행위라는 짐을 지게 된다.

시스템 요건 번호	시스템 요건 이름	검증요건 번호	검증 요건 이름	검증 방법
3.2.1.1	요건 A	4.2.1.1	요건 A	데모
3.2.1.2	요건 B	4.2.1.2	요건 B	시험
3.2.1.3	요건 C	4.2.1.3	요건 C	해석
3.2.1.4	요건 D	4.2.1.4	요건 D	검사
3.2.1.5	요건 E	4.2.1.5	요건 E	N/A

그림4. 전후 참조 검증 표의 예

최근에는 계약자가 요건을 설계 검증 시험이나 절차에 연계하여 관리하는 것이 요구되고 있다. MIL-STD-2167A에 의하면 계약자는 소프트웨어 시험 기술 문서(MIL-STD-2167)에서 찾아낸 시험 사례를 만족시키는 소프트웨어 요구 사양과 인터페이스 요구 사양에서 요건의 추적성을 문서화해야 하며, 더 나아가 모든 요건은 문서화되고 시험 결과에 연계되어야 한다고 적고 있다. 상업적인 분야에서도 요건 수립 초기 단계부터 이들 요건을 검증하는 결과까지를 추적하는 개념을 표준화해 왔다. IEEE P1220에 의하면 검증은 모든 요건이 분해와 할당 과정을 통해 추적되어서 부품, 컴포넌트 또는 하위 시스템의 검증 활동이 시스템 요건의 만족을 지원하도록 검증이 이루어져야 한다. 추가로, 검증 과정은 시스템 통합과 그 과정에서 요건에 제품이 부합하도록 보증하는데 필요한 시험 행위를 포함한다. IEEE-1220과 ISO9000에 따르면 공급자는 최종 제품이 규정된 요건을 만족시키는 명확한 증거를 위해 품질 계획이나 절차에 준해서 모든 시험을 수행하도록 하고 있다. 또한 공급자는 제품이 허용 기준을 만족시킴을 입증하는 필요한 모든 검사와 시험 증거와 기록을 유지해야 한다.

### 5.2 RDD-100에 의한 방법

RDD-100은 entity-relation으로 이루어진 데이터베이스이다. 즉, 데이터베이스는 엘리먼트(element) 유형으로 이루어진 스키마(schema)로 구성되며 엘리먼트 유형(객체)은 다른 엘리먼트 유형과 관계(relation)를 갖는다. 스키마는 Ascent Logic사가 만든 엘리먼트 유형과 관계로 구성되어 있다. 엘리먼트 유형은 자신을 정의하는데 필요한 일련의 속성을 갖는다. 모든 엘리먼트 유형은 자신을 정의하는 고유한 속성들을 갖는다. RDD-100 데이터베이스는 핵심적인 스키마를 확장해서 새로운 엘리먼트 유형과 관계를 추가할 수 있다. 따라서, RDD-100은 소스 문서로부터 만들어진 요건으로부터 이들 요건을 검증하는 시험 결과를 연계하여 추적, 관리하는데 효과적으로 사용될 수 있다. 그림5와 같이 RDD-100의 데이터베이스를 확장해서 검증 결과 엘리먼트로 Verification Cases라는 엘리먼트 유형을 추가한다. 추가된 엘리먼트 유형의 속성을 정의하고 다른 엘리먼트 유형(System Requirement, Verification Requirement, Verification Method)과의 관계를

설정한다. 시스템 요구사항에 해당하는 요건 엘리먼트는 이들을 검증하는데 사용되는 검증 요건 엘리먼트와 verifies/is verified라는 관계를 갖도록 하며, 검증 요건 엘리먼트는 검증 방법 엘리먼트와 has the verification method of라는 관계를 갖는다. 검증 결과 엘리먼트는 검증 요건 엘리먼트와 validates/is validated by라는 관계를 가지며 검증 방법 엘리먼트와 confirms/is confirmed by라는 관계를 갖는다.

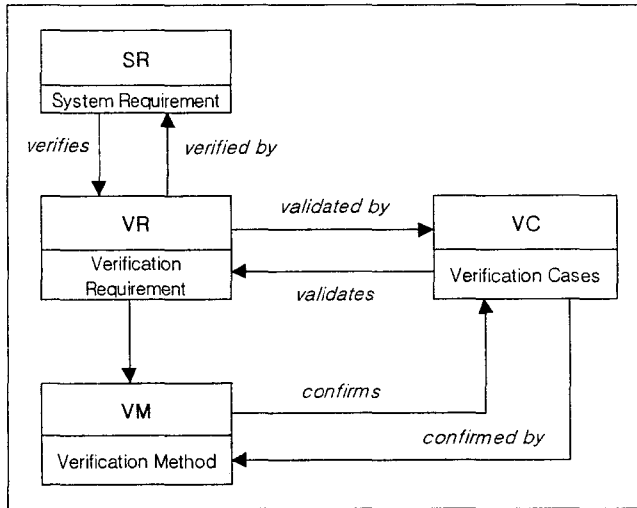


그림5. RDD100내에서 시스템 검증 관리 체계

## 6. 결론

시스템의 초기 요건 생성에서부터 요건에 의해 정의된 기준(설계 사양), 이를 만족시키도록 설계된 시스템과 이의 검증을 위한 시험평가 및 결과까지를 연계해서 추적, 관리할 수 있는 시스템 엔지니어링 기법을 개발하기 위하여 시스템 요건관리 및 개발관리를 위한 전산 체계(프레임워크)를 구축하였다. 이를 통해, 현재까지 추적성이 거의 없이 산재해 있던 시스템 요건, 시스템 구조, 작업 내용(WBS), 과제, 조직, 개발공정 등 시스템엔지니어링 데이터들이 연계성을 갖고 추적될 수 있게 되었다. 이들 시스템엔지니어링 데이터의 DB화와 추적성을 위해 시스템엔지니어링 전산 도구인 RDD-100을 사용하였다.

이상에서 설명한 시스템 엔지니어링 전산 체계는 다음과 같은 면에서 시스템 엔지니어링에 유용하게 활용될 수 있을 것이다.

- 시스템간의 불일치를 도출하고 이를 추적 관리함으로써 적기에 문제를 해결.
- 명확한 검증을 위해 설계, 제작 및 시험 단계에서 추적, 관리되어야 할 사항을 도출.
- 요건, 기능, PBS, WBS와 검증 요건을 연계 관리해서 작업의 우선순위, 작업에 대한 책임, 성공 및 실패 여부 판정

시스템 엔지니어링 전산 체계를 사용함으로써 다음과 같은 효과를 얻을 수 있다.

- 최종 시스템이 요건을 만족시킬 가능성의 증대
- 불일치 사항의 조기 도출 및 해결, 인터페이스의 명확한 정의, 공유 정보의 활용 증대를 통해



위험을 감소시킴.

● 요건, 설계 및 검증(시험) 내용의 문서화 및 의사결정 과정의 추적

고속철도 시스템엔지니어링 전산 체계를 완성하기 위해서 (1) 시스템 요건 및 시제차 상세사양의 일관성 유지와 효율적 관리를 위해 이들을 통합화하고 RDD-100에 의한 번호체계 사용, (2) 시제차를 위한 시스템 구조 모형과 번호체계 보완, (3) 요건의 만족 여부를 검증할 검증 요건의 수립, (4) 검증 요건, 검증 방법 및 검증 결과의 DB화와 연계 추적성 확보, (5) RDD-100을 사용하지 않는 하위과제와 시스템엔지니어링 과제간의 정보전달 체계의 수립, (6) 컴포넌트 단위로 이루어진 성능 평가 연계와 같은 작업이 추가적으로 이루어져야 한다.

후기: 본 연구는 'G7 고속전철기술개발 프로젝트'에서 '고속전철 시스템 엔지니어링 기술개발' 과제의 3차년도 연구 가운데 일부로 수행되었다.

**참고 문헌**

1. System Engineering Handbook, INCOSE Release 1.0, January 1998.
2. Ascent Logic Corporation, Introduction to Object-Oriented Systems Engineering, RDD-100 Training Course #142.
3. Byron Smith, "The Application of a Tool Like RDD-100 to Accomodate Traceability from Requirements to Test Results," 6th Annual International Symposium of The International Council on System Engineering(INCOSE96) Proceedings CD ROM, Los Angeles, 1997..
4. M. E. Sampson, "Driving Test and Verification From SLATE", 7th Annual International Symposium of The International Council on System Engineering(INCOSE97) Proceedings CD ROM, Los Angeles, 1997.
5. Loyd Baker and Ann Christian, "Requirements Development & Management Using Models," 8th Annual International Symposium of The International Council on System Engineering(INCOSE98) Proceedings CD ROM, Vancouver, July 26-30, 1998.
6. 고속전철 시스템 엔지니어링 기술개발, 한국철도기술연구원, 고속전철 기술개발 3차년도 연구보고서, 1999년 10월.