2000년도 대한전기학회 하계학술대회 논문집 2000.7.17-20

한국형 고속전철 시스템엔지니어링 데이터베이스 구축

황희수, 이태형, 현승호, 정홍채, 김대욱
한국철도기술연구원

System Engineering Database for Korea High Speed Rail System Development

H. S. Hwang, T. H. Lee, S. H. Hyun, H. C. Jeong, D. W. Kim

Abstract - 고속철도는 상당히 복잡한 시스템이기 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 최소로 하기 위해 체계적인 시스템 통합 절차를 제공할 수 있는 시스템엔지니어링 데이터베이스 구축이 필요하다. 이 데이터베이스는 시스템 요건부터 시스템 기능, 컴포넌트, 평가 요건, 평가 결과까지 추적성을 갖도록 구축되어서 연계 관리를 가능하게 한다.

1. 서 론

고속철도 시스템은 차량, 기계부품, 전기, 전자, 제어, 정보통신, 토목기술 등이 종합적으로 적용되는 대형 복합 시스템의 하나로서, 시스템의 개발 요건이 체계적으로 관리되어야만 하는 전형적인 예라고 할 수 있다. 요건은 하위 계약자와의 계약조건에 반영될 뿐만 아니라 과제의 수행근거가 되며 시스템의 통합 검증의 근거로서 모든 개발단계의 기본사양이 된다. 고속전철 시스템 기술 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 서브시스템에 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 최소로 하기 위해 체계적인 시스템 통합 절차를 제공할 수 있는 시스템엔지니어링 체계를 구축하고, 이를 데이터베이스화하여 종합적으로 연계관리를 할 수 있도록 하는 것이 본 연구의 목표이다.

2. 요건 관리

고속철도시스템 개발과 같은 대형 프로젝트의 시스템 엔지니어링은 개발 전주기 동안 사양의 일관성을 유지하고 개발된 시스템이 사양을 만족시키는 것을 보증해야 하는 과업을 갖는다. 시스템 엔지니어링의 주요 목적 가운데 하나는 요구사항을 효과적으로 관리하는 것이다.

2.1 요건관리 기술 및 개념

요건 관리 기술은 시스템 엔지니어링을 수행하는 동안 지속적이고 다양한 순서로 적용된다. 요건 기술에는 문제 분석, 시스템 정의, 시스템 정의 교정, 프로젝트 범위 관리, 요구사항 변경관리가 있다. 여기서는 새로운 프로젝트의 첫 단계에 적용하는 순서로 요건 기술을 설명한다.

문제 분석: 초기 이해 당사자의 필요를 이해하고 최상위 수준의 해결책을 제안, 문제 분석 동안 합의점을 도출하며, 초기 해결 범위와 제약조건을 기술 및 사업 측면에서 결정한다.

시스템 정의: 시스템을 정의한다는 것은 이해 당사자의 필요를 이해해서 개발 시스템에 대한 의미 있는 기술로 바꾸어 조직화하는 것이다. 시스템 정의 초기에는 요구사항, 문서 형식, 요구사항 수준, 우선 순위와 평가 수준, 위험도, 프로젝트 범위 등을 결정한다.

프로젝트 범위 관리: 이것은 요구사항에 의해 결정되며 개발기간, 인력, 예산과 같은 이용 가능한 자원을 조화시키는 것이다. 범위 관리는 프로젝트의 범위를 더 작고 관리하기 쉬운 형태로 나누거나 차츰 키워 나가는 활동의 연속이다. 요구사항의 포함 여부를 결정하는 기준으로 우선 순위, 필요한 노력과 위험 같은 요구사항의 속성을 사용한다.

시스템 정의 교정: 최상위 수준의 시스템 정의와 초기 과업 범위를 잘 이해하면 시스템 정의를 보다 정확하게 할 수 있어서 보다 효율적으로 자원을 투입할 수 있게 된다. 시스템 정의를 교정할 때 두 가지 중요한 점을 고려해야 한다. 즉, 최상위 수준 시스템 정의를 더 자세하게 기술하고 시스템이 이해 당사자의 요구에 부합하는지 검증하는 것이다.

요구사항 변경 관리: 초기에 아무리 요구사항을 신중하게 정의했더라도 요구사항은 변한다. 요구사항 변화를 관리하는 데는 기준 출발점의 설정, 요구사항의 이력 추적, 추적에 유리한 의존성 결정, 항목들의 추적 관계 설정, 버전 관리와 같은 행위들이 포함된다. 변경 관리는 지정된 팀이 수행하며 변경을 승인하는 프로세스를 설정하는 것이 중요하다.

요구사항 관리 기술을 프로젝트에 적용하기 위해서는 프로젝트 참가자 모두가 다음과 같은 요구사항 관리 개념을 이해하고 있어야 한다.

요구사항 종류: 시스템이 커지고 복잡해질수록 요구사항의 종류는 더 많아진다. 요구사항 종류는 간단하게 말해서 요구사항들의 클래스이다. 요구사항의 종류를 인지함으로써 시스템공학 팀은 다수의 요구사항을 의미 있고 관리하기 쉬운 그룹으로 분류할 수 있다. 프로젝트에서 요구사항의 여러 종류를 설정하는 것은 팀 구성원이 변경 관리를 원활하게 하고 의사소통을 명확하게 하는데 도움을 준다. 요구사항의 종류는 더 세분화되거나 다른 종류로 분해될 수 있다. 요구사항은 분석과 설계를 위해 더 이상 분해할 수 없지만 검증이 가능한 형태로 분해될 수 있다. 검증 요건은 시스템 요건으로부터 도출되고 특정한 검증 절차로 분해된다.

상호 기능적 팀: 요구사항 관리에는 필요한 전문지식이 개발 프로세스에 기여할 수 있도록 모든 구성원이 포함되어야 한다. 개발 관리자, 생산 관리자, 분석가, 시스템 엔지니어와 심지어 사용자까지 포함되어야 한다.

추적성: 요구사항의 한 문장이라도 단독으로 존재할 수 없다. 이해 당사자의 요구사항은 그것을 만족시키기 위해 제안된 제품의 특성과 관련이 있어야 한다. 제품의 특성은 기능 및 비기능적 거동과 관련된 요구사항에 연관된다. 시험 사례는 검증 및 확인이 필요한 요구사항에 연관된다. 요구사항들은 서로 독립적이거나 의존되어 있다. 시스템 엔지니어링 팀이 변화의 영향을 결정하고 시스템이 기대에 부응한다는 것을 확신하기 위해서는 이런 추적성 관계를 정의하고 문서화해서 관리해야 한다. 추적성은 요건 관리에서 구현하기가 쉽지 않지만 변경 될 대상을 관리하고 변경에 따른 영향을 파악하는데는 매우 유용하다. 변경 관리를 보다 효과적으로 하기 위해서는 이런 추적성에 의해 변경이 관리되어야 한다.

변경 이력: 환경 변화, 새로운 요구사항 도출, 새로운 기술의 도입 등 변경 요인은 많기 때문에 변경은 필연적인 것으로 봐야 한다. 프로젝트 요건의 버전을 기록하는 것은 팀 리더가 프로젝트 변경 사유를 알도록 하는데 도움을 준다. 개별 요구사항이 변경될 때마다 그 이력을 이해하는 것이 중요하다. 무엇이 왜, 언제, 그리고 누구에 의해 변경되었는지 알 수 있어야 한다.

2.2 요건관리 프로세스

개발 수명 주기 동안 설계와 일치하고 완전한 설계를 유지하는데 드는 비용은 크다. 또한 요구사항이 적절하게 수행되지 않는다면 설계에 의해 발생한 오류들에 의해 프로젝트는 심각한 위험을 감수하게 될 수 있다. 여기서는 설계에서 불일치를 감소시킬 수 있는 방법에 대해 설명한다.

2.2.1 일관성 유지

요구사항의 품질이 프로젝트의 성공에 직접적으로 영향을 주기 때문에 품질이 보증된 요구사항을 개발하는 것이 무엇보다 중요하다. 이를 위해 요구사항 도출에 비중이 있는 전산 도구를 사용해서 요구사항을 분석하고 요구사항을 관리할 수 있는 프로세스를 확립해왔다. 품질이 보증된 요구사항이란 것은 그 자체로 완전하고, 일관성이 있으며 증명이 가능해야 한다.

RDD의 사용을 통해, 요구사항이 스키마로 정의된 구조에 의해 유지되기 때문에 요구사항의 일관성 검사를 자동화할 수 있다. RDD는 ERA(Entity-Relationship-Attribute)로 이루어진 모델로 스키마를 지원하며 잠재적인 오류를 감소시키기 위해 일관성을 검사할 수 있는 기능을 제공한다. 따라서, 시스템 엔지니어는 기술적인 측면에서의 일관성에만 초점을 맞출 수 있다. 일관성을 유지한다는 것은 요구사항이 정의된 명확한 기준에 부합한다는 것을 의미한다.

요구사항의 일관성에는 다음과 같은 네 가지가 있다.

문장 일관성(Textual Consistency): 요구사항을 위해 최소 기준을 구성할 중요한 단어를 예를 들면, 모든 고속전철 시스템 요구사항은 요구사항 문장에 동사 'shall'을 갖는다.

정적 일관성(Static Consistency): 이것은 요구사항이 자원이나 자신의 상위 또는 하위 요구사항과 정당한 관계를 갖고 있는지를 확인한다. 또한 어떤 필요에 의해 정의된 관계에 일관성이 있도록 해준다.

동적 일관성(Dynamic Consistency): 이것은 정의된 모델의 구조가 논리적으로 옳음을 보증한다. 물론, 이것이 모델이 요구된 행동을 보인다는 것을 의미하는 것은 아니다. 그러나 의미적으로 모델에 오류가 있을 가능성은 줄어든다.

결합 일관성(Composition Consistency): 시스템 요건의 문맥에 내포된 기술적인 내용에 대한 엔지니어링 평가를 위한 것이다. 여기에서 강조되는 것은 요구사항이 완전한지, 검증 가능한지, 실현이 가능하고 타당성이 있는지를 보는 것이다.

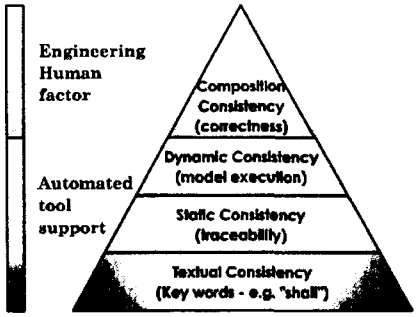

그림1. 요건에 대한 일관성 관리 항목

그림1에서처럼 일관성의 아래 세 범주는 일관성 확인을 위해 공식화된 규칙을 검사하는 개념적인 토대를 제공한다. 이러한 토대는 일관성 검사의 자동화를 가능하게 한다. 세 부류의 일관성 검사를 자동화함으로써 시스템 엔지니어가 개발 및 요구사항의 기술적 내용에 더 많은 시간을 투자할 수 있도록 해준다. 즉, 시스템 엔지니어는 결합 일관성을 확인하는 것에만 노력을 집중할 수 있다. 반면에, 문장 일관성, 정적 일관성 및 동적 일관성에 대해 RDD의 일관성 검사 기능이 자동으로 데이타베이스를 검사한다.

2.2.2 일관성 점검 프로세스

요구사항 개발 프로세스에 오류를 범하기 쉬운 부분이 있다. 고속전철 시스템의 요구사항 각각은 5내지 10개의 연관된 링크를 갖는다. 엔지니어는 각 요구사항을 RDD 데이타베이스 안에 정의된 관계 요소를 통해 소스 문서, 검증 요건, 검증 방법, 거동 모델, 세부 시스템 요구사항들에 연결해야 한다. 요구사항의 불일치는 설계 결함, 불완전한 인터페이스 정의, 검증할 방법의 미정의, 하위 수준 통합과 시험에 대한 연계성 부족 등에 기인한다. 일관성 검사는 요구사항 불일치를 최소로 하는데 그 목적이 있다. 일관성 검사 방법과 그것을 절차화하는 것에 대해 논의해 보자.

1) 일관성 검사 규칙의 설정

요구사항 추적성과 완전성은 일관성 검사의 기준을 형성한다. RDD와 같은 데이터베이스 시스템에서는 사용할 스키마의 정의와 이들의 연계성이 일관성 검사를 위한 기준이 된다. 스키마가 정의되면 스키마 사용 방법에 대한 규칙을 정의하게 된다. 예로, 다음과 같은 규칙이 있을 수 있다. 요구사항은 다수의 상위 요구사항을 가질 수 없다. 거동 모델에서 메시지는 함수 사이에 입출력된다.

2) 변경 절차의 설정

요구사항을 효과적으로 관리하는데 필요한 핵심 요소는 변경 프로세스와 요구사항 기준 출발점(baseline)을 운영하는 것이다. 이 프로세스는 엔지니어가 사전에 정의된 절차에 따라 변경 패키지를 제공해야 한다. 변경 패키지의 내용은 요구사항과 관련이 있는 정보를 담고 있다. 시스템 엔지니어의 일 가운데 하나는 요구사항이 기술적으로 유효한지를 확인하고 일관성을 유지하기 위한 지침을 제공하는 것이다. 일관성 지침에 따라, 변경 패키지의 일관성 검사를 위해 자동화 할 부분을 찾고 이를 RDD의 일관성 검사 규칙에 코딩한다. 변경 패키지에 대한 일관성 평가를 수동으로 한다면 이는 복잡하고 번거로우며 많은 시간이 소비된다. 고속철도 시스템엔지니어링에서 상위 수준의 요건 관리만 고려한다고 해도 변경 패키지는 수 천개의 데이터베이스 구성 요소에 연관되는 복잡한 데이타 구조를 갖는다. 따라서, RDD내에서 변경 패키지에 대한 일관성 검사를 자동으로 수행해서, 필요한 관계가 구현되었는지 확인할 수 있는 것은 커다란

이점이 될 수 있다. 시스템 엔지니어는 변경에 대해서 검사할 일관성 항목을 커스토마이징 할 수 있으며 발견된 불일치 사항을 자동으로 보고서화 할 수 있다. 이 보고서는 관련 요소와 불일치 내용을 포함하며 검사 활동에 도움을 준다. 이를 통해 시스템 엔지니어링 팀은 변경 패키지의 기술적인 내용에 초점을 맞출 수 있게 된다. 일단 변경 패키지가 정확하고 일관성에 문제가 없으면 다음 단계로 요구사항 검사를 수행한다. 이것은 변경된 요구사항이 완전한지, 검증 가능한지, 타당하고 일관성이 있는지를 결정하는 것이다. 아래 그림2는 변경 패키지의 불일치를 해결하는 과정을 보여준다.

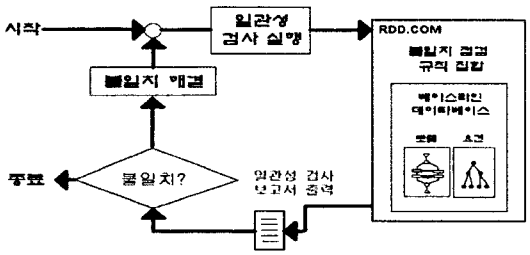

그림2. 불일치 해결 프로세스

3. 시스템 설계 데이터베이스

이 절에서는 상위 시스템 레벨에서 시스템 요건의 탄생에서부터 평가 결과에 이르기까지 추적성을 부여할 수 있는 시스템 엔지니어링 데이터베이스에 대해 기술한다.

3.1 추적 관리 배경

과거에 계약자는 상세 사양을 검토하기 위해서는 관련 문서를 정의하고 있는 상호 참조 표를 이용하였다. 이 표는 시스템 요건과 그것을 평가할 평가 요건이 연계되어 있다. 이 표에는 개괄적인 평가 방법이 포함된다. 시스템 설계가 진행되는 과정에서 이 표를 유지하고 관리하는데는 상당한 노력이 필요하다. 변경이 수시로 발생하며 해당 내용을 일일이 찾아서 변경하는 일은 소모적이며 반복적인 작업이다. 변경을 한다해도 그것의 완전성이나 일관성을 확인할 수 있는 방법이 없었다.

검증 및 확인(V&V)에서 RDD는 시스템 거동 모델을 이용할 수 있다[7,8]. 거동 모델은 개발 중인 시스템의 기능을 표현한다. 이 모델은 실행할 수 있는 시스템 요건이며 시스템의 거동을 시험할 수 있는 시험 계획의 개발을 용이하게 해준다. 실행 모델은 시험 계획에 따라 시험 대상 장비에 입력을 줄 수 있다.

시스템 거동의 검증은 시험 계획에 정의된 시험 조건에 따라 수행되어야 한다. 그러나 아직까지 RDD 거동 모델은 시험 대상에 실시간으로 데이터를 줄 수 없다. 또한 시스템이 복잡할 경우 거동 모델의 복잡성이 증대하기 때문에 시스템 요건을 하나의 모델로 만드는 것은 실용적이지 못해 보인다. 따라서 시스템 요건의 탄생부터 해당 요건의 검증까지를 연계해서 관리할 수 있는 실용적인 방법이 필요하다.

3.2 추적성 구현

실용적인 해결책으로, 요건의 탄생으로부터 시험 결과 및 판정에 이르기까지를 추적 관리할 수 있도록 RDD 데이터베이스를 구현하였다. RDD 데이터베이스는 다수의 구성 형태(Element Type)로 이루어진 스키마로 구성된다. 이들 스키마는 서로 연계성을 갖도록 관계(Relation) 구성 요소를 갖는다. 각 요소는 자신의 특성에 맞는 속성을 갖는다. RDD는 기본 스키마를 확장할 수 있으며, 이런 능력을 활용하여 요건 탄생부터 시험 평가까지 모든 관련 항목을 데이터베이스화 할 수 있고 관계 요소를 이용하여 추적성을 부여할 수 있다.

고속전철 시스템의 경우, 검증 결과를 관리할 수 있는 스키마를 새로이 만들었으며 데이터베이스내의 다른 구성 요소와 연계시키기 위해 필요한 속성을 정의하였다. 그림3은 고속전철 시스템 엔지니어링 데이터베이스 모형으로 입력 문서에서 시스템 사양, 컴포넌트 구조, 평가 요건, 평가 결과 등이 서로 연계되어 추적 관리될 수 있음을 보여준다.

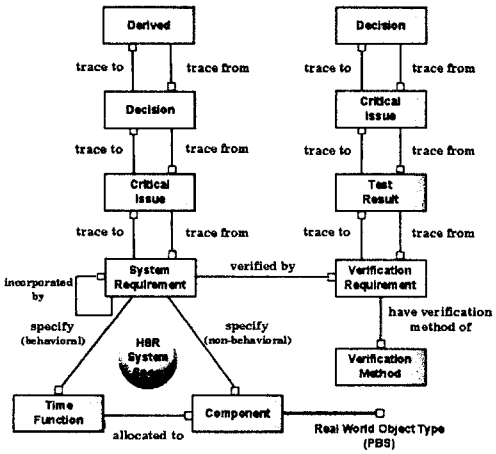

그림3. 고속전철 시스템 엔지니어링 데이터베이스 모형

4. 결 론

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 시스템엔지니어링 기법을 개발하기 위하여 시스템 요건관리 및 개발관리를 위한 전산 체계(프레임워크)를 구축하였다. 이를 통해, 현재까지 추적성이 거의 없이 산재해 있던 요건들, 시스템 컴포넌트, 작업 내용, 과제, 조직 등 시스템엔지니어링 데이터들이 연계성을 갖고 추적될 수 있게 되었다.

후기: 본 연구는 'G7 고속전철기술개발 프로젝트'에서 '고속전철 시스템 엔지니어링 기술개발' 과제의 4차 년도 연구 가운데 일부로 수행되었다.

[참 고 문 헌]

[1] A. H. Anderegg, A concurrent process for development and validation of operational requirements, INCOSE96
[2] D. Locke, Implementing an effective requirements management process, pp. 187-192, ICSE99, August 9-12, Las Vegas.
[3] I. Plastow, How do you model a system?, INCOSE97
[4] H. S. Hwang, et al., Design of system engineering database for KHSR, *Proceedings of Korean Society for Railway*, pp. 121-129, Nov. 1999.
[5] H. S. Hwang, et al., Application of a system engineering tool to high speed railway system development project, COMPRAIL 2000, 11-13 Sep. 2000, Bologna, Italy
[6] L. Baker, A. Christian, Requirements development & management using models, INCOSE98
[7] M. Alford, RDD support for V&V, Technical notes, 1993.
[8] M. Alford, Behavior based system test planning, *Proceedings of the 3rd Annual International Symposium of the National Council of System Engineering*, 1993.
[9] S. Datto, C. Weaver, D. Parfitt, and M. Rothstein, System engineering life cycle management with traceability, pp. 61-66, ICSE99, August 9-12, Las Vegas.