

동시적 업무수행을 위한 통합된 시스템 설계 프로세스 연구

김진훈[†], 박영원

아주대학교 시스템공학과 박사과정, 아주대학교 시스템공학과 교수

Research of Integrated System Design Process for Concurrency Design Activity

Jin-Hoon Kim[†], Young-Won Park

Department of Systems Engineering, Ajou University

Key Words : Concurrent Engineering, Process integration, Process definition, Process model, Design Structure Matrix.

ABSTRACT

This study describes process and method to establish concurrent and/or sequential schedule planning for the system design process of the EIA632 standard. For this purpose, the study suggest process and method that are context analysis of standard process, interface definition from activity decomposition, integration of related activities, and definition of concurrent and/or sequential work flow. The proposed process and method will contribute to minimize time loss that is emerged from activities iteration.

1. 서론

CMMi, ISO15288, EIA632, IEEE 1220등과 같은 표준들은 시스템 개발과 관련된 요구사항들을 제시하고 있다. EIA632(1999)에서는 이러한 완전하고 일관성 있는 요구사항들은 실현가능하고 비용 효과적인 시스템의 솔루션을 제공할 수 있도록 하는 요구사항의 수립 및 진개를 지원하며, 비용, 일정 및 리스크의 제약사항 내에서 고객 요구를

만족시키는 것이 표준의 목적임을 밝히고 있다. 그러나 이와 같은 표준들은 기업이 적용할 수 있도록 구체적인 수행방법이나 절차를 기술하고 있지는 않다.

업무의 절차가 계획되지 않은 상태에서 표준의 적용은, 수많은 업무들과 이들 간의 복잡한 인터페이스로 인해, 누락되거나 중복된 업무가 수행될 수 있으며, 동시공학(Concurrent Engineering)적 접근이 제대로 계획되지 않아, 시스템 개발을 효율적으로 운용하지 못하여 개발기간 연장, 개발비용

[†] 교신저자 jhkim00@iae.re.kr

증대와 같은 문제를 발생시킬 수 있다. 만약, 표준의 프로세스 중 일부분을 적용하고자 하는 경우에는 있어서는, 적용하고자 하는 요구사항과 연계된 기타 요구사항을 식별하여 잘 조정되어 통합된 절차를 수립하여 제품개발 업무에 적용해야 기대되는 효과를 얻을 수 있을 것이다.

본 연구에서는 시스템을 엔지니어링 하는 프로세스의 표준인 EIA 632로부터 시스템을 설계하는 프로세스를 기업에 적용하기 위하여, 순차적이며 동시적 업무를 계획하는 프로세스와 방법을 연구하였다. 특히, 표준에서 시스템을 설계하는 프로세스 일부분을 적용하고자 하는 경우에는 연관된 기타 프로세스의 요구사항을 식별하고, 인터페이스를 정의하여야 한다. 이를 위해, 정황분석 절차와 DSM(Design Structure Method) 방법을 업무간의 인터페이스를 고려한 순차적인 업무 계획을 수립하기 위해 연구하였다.

1.1 표준 프로세스의 산업 적용 개념

CMMi, ISO15288, EIA632, IEEE1220 등은 시스템 또는 제품을 엔지니어링하기 위해 적용되는 표준들이다. 이들 표준들은 DoD 표준 MIL498을 근간으로 시스템엔지니어링 민수 표준으로 발전되었다[5]. CMMi는 시스템 엔지니어링에 대한 평가모델로 엔지니어링 결과에 대한 평가 요구사항을 다루고 있다. ISO15288, EIA632, 그리고 IEEE1220은 수명주기 적용단계와 프로세스 요구사항의 상세화 수준에 의해 구분된다. ISO15288은 시스템 수명 주기 전체를 대상으로 다루고 있으나 위의 3개 프로세스 중 내용의 상세화 수준이 가장 낮다. EIA632는 개념단계, 개발단계, 그리고 생산단계에 적용되며,

ISO15288보다 요구사항이 더 구체적이다. IEEE1220은 개발단계에 해당하는 시스템 설계에 해당되는 시스템엔지니어링 프로세스 표준이다.

EIA632 표준 개발에 참여한 Martin(2000)은 일단, 산업에서 적용할 표준을 선정하면, 해당 표준은 정책과 절차를 수립하는 요구사항이 되며, 정책과 절차에 적합하도록 선정된 프로세스를 적용하기 위한 계획과 일정이 수립됨으로써 프로젝트가 구현된다고 가이드하고 있다. 그러나 표준에서는 프로세스 요구사항을 어떻게 구현하는지에 대한 구체적인 내용을 제시하지 않고 있다. EIA632(1999)에서는 개발자가 기업의 정책과 절차에 부합되면 업무 적용에 용이한 방법과 도구를 직접 선정하거나 정의하도록 하기 위해 구현방법을 구체적으로 제시하지 않는다고 설명하고 있다.

따라서 표준에서 제시된 프로세스를 산업에 적용하기 위해서는, 기업이나 프로젝트에서 적용하고자 하는 표준 프로세스로부터 식별된 업무들을 적절히 통합하고, 순차적인 업무 흐름으로 계획과 일정을 수립할 필요가 있다. 아울러, 업무의 누락이나 뒤늦은 업무의 반복을 제거함으로써, 효율적이며 기업에 적절한 통합된 프로세스를 확보하기 위한 방안 또한 마련되어야 할 것이다.

특히, 시스템 설계 단계에서 이루어진 업무들은 동시적이며, 반복적으로 이루어져야 할 업무가 많은 단계이다. Tyson(1998)에 의하면, 시스템 설계 단계에서 나타나는 업무반복은 의도적(intentional)인 경우와 비의도적(unintentional)인 경우로 분리되는데, 의도적인 업무는 원하는 해결방안으로 수립할 수 있는 유용한 정보를 만들기 위해 상호 연계된 설계 프로세스를 수행하는 경우

이며, 비의도적 업무는 프로세스 진행상에 실수나 시험의 실패 등의 잘못된 경우로 새로운 정보가 유입되어 반복이 발생하는 경우로 설명하고 있다. Clausing(1994)도 “창의적 반복”과 “원칙적 반복”, 그리고 “역기능 반복”으로 설계 단계에서 나타나는 업무 반복 현상을 분류하고 있다. 이 또한, 의도적, 비의도적 관계로 크게 분류할 수 있다.

본 연구에서는 시스템엔지니어링 표준의 시스템설계 프로세스에서 발생하는 비의도적인 반복은 업무를 분해하여 순차적 업무를 계획할 수 있도록 하고, 의도적인 반복은 동시적인 업무로 계획할 수 있도록 하였다. 연구대상이 되는 표준을 선정함에 있어, ISO15288의 내용은 적용대상 범위가 광범위하고 시스템설계 프로세스에 대한 내용의 상세화 수준이 낮기 때문에, 상세업무를 분석하기 위해서는 ISO15288의 근원 표준이 되는 EIA632를 다시 참고해야 한다. IEEE 1220은 시스템설계 프로세스를 직접적으로 다루고 있어 이미 인터페이스 정황이 명확하기 때문에, 본 논문의 시스템설계 절차와 방법을 적용함에 있어 연구 결과의 효과성이 잘 나타나지 않는다. 이에 최종적으로 수명주기 단계와 업무상세수준에서 EIA632의 시스템설계 프로세스가 연구 대상으로 적절한 것으로 판단하였다. 그러나 기업에서 어떤 표준을 적용하는 것이 바람직하다는 판단은 본 논문에서 제안되지 않는다.

1.2 프로세스 설계 절차와 방법

Tyson(2002)은 프로세스를 시스템의 한 종류로 간주하고 있다. 따라서 프로세스의 업무도 시스템적으로 분해 및 통합이 가능하며, 효과적이고 효율적인 업무를 설계할

수 있다고 주장하고 있다. Dennis(1999)는 시스템을 정의하고 새로운 가치를 만들어가는 프로세스는 분해(decomposition)와 정의(definition)의 반복 과정이며, 정황분석을 통해 시스템 개발의 경계(Boundary)가 정의된다고 설명하고 있다. 본 논문에서는 EIA632 표준에서 제시되고 있는 프로세스를 시스템으로 간주하고, EIA632의 시스템 설계 프로세스를 순차적 업무로 계획하는 절차를 다음과 같이 수립하였다.

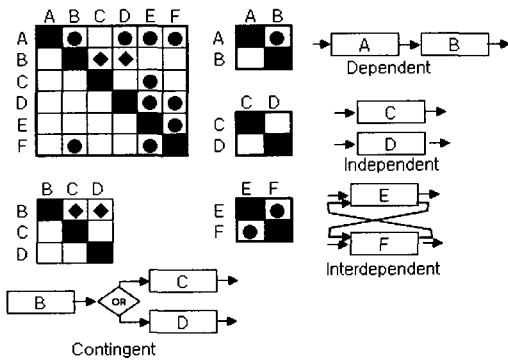
- 표준 프로세스로부터 시스템설계 업무 식별
- 정황분석(타 프로세스와의 인터페이스 식별)
- 업무 분해를 통한 입출력 정의
- 통합 및 업무순서 정의

또한, 정황분석, 업무분해 및 입출력 정의, 그리고 통합 및 순차적 업무정의를 위한 방법으로 DSM(Design Structure Matrix)을 적용하였다. DSM은 분해와 통합이 가능하고, 피드백되는 정보의 흐름을 표현할 수 있는 간단한 모델이다. Tyson(2002)은, 일반적으로 알려진, 간트차트(Gantt Chart), PERT, CPM, 흐름도(Flow chart)나 IDEF 다이어그램에서는 데이터의 반복성을 표현하기 위한 피드백을 표현할 수 없거나, 한 눈에 확인하기 어려운 표현의 복잡성에 비해 DSM은 매트릭스 형태로 한 장에 표현할 수 있는 장점을 가지고 있다고 설명하고 있다.

앞에서 제시된 절차와 방법은 표준의 최상부 프로세스에서 한 단계씩 한 단계씩 하부 업무로 분해하여 모든 인터페이스가 고려된 업무를 식별하고, 이들과 통합됨으로써 종합적인 일정을 계획하는 시스템적 접

근 방식이다. 특히, 최근에 CMMi로 인해 부각된 시스템엔지니어링 표준인 EIA632를 프로젝트에 적용할 수 있도록 업무 계획 절차와 방법을 연구하였다는 점에 중요한 의의가 있다고 할 수 있다.

DSM은 행과 열로 구성된 정사각 형태의 매트릭스이며, DSM에 표현된 활동과 이의 관계 예제를 Figure 1에 나타내었다.



<Figure 1> DSM 활동 및 관계 표현

Figure 1에서 DSM에서 좌측 상단에서 우측 하단까지의 대각선에 위치한 각 칸들은 순서가 정해지지 않은 활동들을 나타내고 있다. 대각선에 위치한 칸들을 제외한 매트릭스의 나머지 부분들은 다른 프로세스의 요소 또는 활동들 간의 의존적 관계, 즉 정보의 흐름을 나타낸다. 만약 정보가 흐름이 시계방향으로 설정된 경우, 각 칸에서 우측으로 정보가 나오고, 아래 방향으로 정보가 입력된다는 것을 의미한다. 이러한 방향 규칙은 반대인 시계 반대방향으로도 정의될 수 있다. 정보 흐름에 대한 예로 Figure 1에서 설명하면, A는 B, D, E, F에 정보를 제공한다. B는 A와 F로부터 정보를 받고, C, D에 정보를 제공한다. DSM에서는 활동들의 관계가 독립적(Independent), 종속

적(Dependent), 상호의존적(Interdependent), 그리고 조건적(Contingent)인 관계를 Figure 1에서와 같이 나타낼 수 있다. 한 예로, Figure 1의 C, D와 같은 독립적 관계는 단독적이며 동시에 업무 수행이 가능한 관계를 보여주고 있으며, Figure 1의 E, F와 같은 상호의존적 관계는 동시적 업무 수행 관계를 나타내고 있다. 동시 공학적 관점에서는 상호의존적 관계가 유관 부서와의 협력 업무를 통해 업무수행에 따른 질적 효과와 개발 기간을 단축할 수 있다는 점에서 더 의미 있는 관계라고 할 수 있다.

DSM의 장점인 피드백을 표현할 수 있다는 점이다. 한 예로, Figure 1의 F는 A, D, 그리고 E로부터 정보를 받아, B와 E에 정보를 제공한다. A에서 F까지가 순차적인 활동 흐름이라고 보았을 때, F에서 B에 제공된 정보로 인해, C, D, E, 그리고 F까지의 업무 반복이 크게 발생할 수밖에 없음을 확인할 수 있다. 이를 개선하기 위한 방법을 DSM에서 제공하고 있는데, 매트릭스의 대각선 방향의 각 칸에서 행과 열의 활동이 교차하는 것을 유지하는 상태에서, 행과 열을 동시에 이동시킨다. 이러한 결과는 활동의 순서가 바뀌면서, 정보 흐름관계 또한 위치가 변한다. 이러한 이동을 통해 종속적이고 의존적인 관계에 있는 활동들이 인접하도록 활동의 순서를 정할 수 있다[6, 7, 8]. Tyson (1998)은 Figure 2와 같은 사례를 통해, 행과 열의 이동을 통해 의존적인 관계에 있는 활동이 근접하게 됨으로써, 반복에 따른 시간 손실이 개선됨을 나타내었다. Figure 2의 위 그림의 업무순서는 Get Socks, Get Shoes, Put on Socks, Put on Shoes, Inspect Shoes의 순이다. 그러나 Inspect Shoes에서 오류가 발생되면, Get

Shoes까지 행위가 되돌아가게 되고, 다시 Put On Socks, Put On Shoes 및 Inspect Shoes까지 진행되어 완료된다. Figure 2의 아래 그림의 경우, Get Socks, Get Shoes 후, Inspect Shoes 행위가 발생하고, 만약 오류가 발견되었을 때는 한 단계 위인, Get Shoes로 곧바로 되돌아 가서 수정되고, 이후 행위가 진행되어 완료된다. Figure 2의 위 그림과 아래 그림을 전체 행위에 걸리는 시간으로 보았을 때, 동일한 업무에 대해 아래 그림이 위 그림보다 시간 손실이 적다는 것을 알 수 있다.



<Figure 2> DSM 행과 열의 이동 사례

2. 통합된 시스템 설계 프로세스

EIA632 표준에서는 기술적 관리, 획득 및 공급, 시스템 설계, 제품 구현 및 배치, 그리고 기술적 평가의 5개의 최상부 프로세스와 이의 하부에 13개의 프로세스, 그리고 33개의 요구사항을 제시하고 있으며, 각 프로세스를 구현하기 위해 수행되어야 할 대표적인 업무들을 정의하고 있다. 본 연구에서 정의하고자 하는 시스템설계 프로세스는 이해당사자의 요구사항을 식별하여 시스템 기술요구사항을 개발하는 요구사항 정의 과

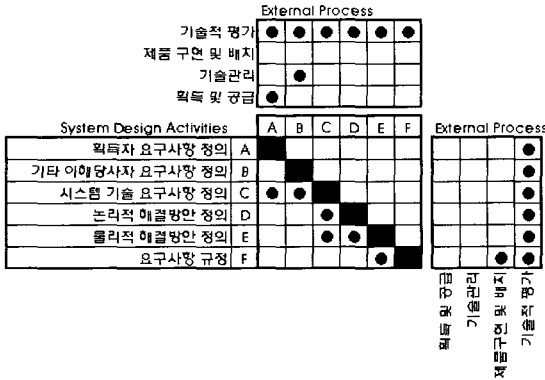
정과, 정의된 기술요구사항으로부터 기능적, 물리적 해결방안을 도출하여 규정된 요구사항을 개발하는 해결방안 정의 과정으로 구성된다. 또한, 시스템 설계 프로세스와 연관된 기술적 관리, 제품 구현 및 배치, 획득 및 공급, 그리고 기술적 평가 프로세스들과의 인터페이스 되는 업무들은 시스템 설계 프로세스에 필요한 정보를 제공하거나, 시스템 설계 프로세스로부터 나오는 결과물을 분석 및 평가하거나, 또는 결과물을 통해 시스템을 구현하는 기능을 가지고 있다.

이러한 시스템설계 프로세스의 최상부 업무를 분석하여 관련된 타 프로세스 업무를 식별하며, 식별된 업무들 간의 인터페이스를 정의하는 과정이 다음과 같이 DSM 방법을 통해 표현되고 전개된다.

2.1 정황(Context) 분석

EIA 632 표준의 시스템설계 프로세스에는 반드시 만족되어야 할 6개의 요구사항을 제시하고 있다. 이 프로세스 요구사항은 수행해야 할 업무가 된다. 예를 들어, EIA 632의 요구사항 14번(EIA 632에서는 33개의 요구사항이 각각 일련번호를 가지고 있다.)인 획득자 요구사항의 내용은 “개발자는 시스템 또는 시스템 일부분에 대해 논증된 획득자 요구사항을 정의해야 한다.”라고 정의되어 있다. 이를 “획득자 요구사항 논증”이라는 업무로 재 정의하였다. 같은 방식으로 요구사항들의 업무를 식별하여 정의할 수 있다. 이와 같이 식별되어 정의된 시스템설계 최상부 업무는, 획득자 요구사항 정의, 기타 이해당사자 요구사항 정의, 시스템 기술 요구사항 정의, 논리적 해결방안 정의, 물리적 해결방안 정의 그리고 요구사항 규

정으로 정의되며, Figure 3의 매트릭스(Matrix) 좌측에 위치하여 나타내고 각 업무의 번호를 A에서 F까지 순차적으로 정했다. 이는 Figure 1에서 설명했던 것과 같이 행과 열이 동일한 매트릭스이다.



<Figure 3> 시스템설계 프로세스 정황

다음은 이들 업무간의 관계를 매트릭스 내부에 '●'로 표시했다. Figure 1에서 설명한 업무 간의 정보흐름 표시의 방향은 시계방향의 흐름으로 설명되었으나, 본 연구에서는 반 시계방향으로 정의하여 작성하였다. 이유는 본 연구에서 사용된 DSM을 구현하는 도구(Tool)인 미국 problematics사의 PSM[®]도구가 반 시계방향으로 모델을 구성하고 해석하도록 되어 있기 때문이었다. PSM[®]도구는 Figure 1과 같은 매트릭스 한 개로 구성된 경우에 대해서만 표현하고 해석할 수 있다. Figure 3과 같은 입력과 출력관계가 표시된 매트릭스는 PSM[®]도구에서 구현할 수 없기 때문에, 본 연구에서는 Microsoft사의 Excel[®]을 사용하였다.

Figure 3은 일반적인 DSM 방법과는 달리, 중앙 매트릭스의 상부와 우측에 또 다른 매트릭스가 있다. 이 매트릭스들은 시스템설계 프로세스의 정의된 업무와 연관된

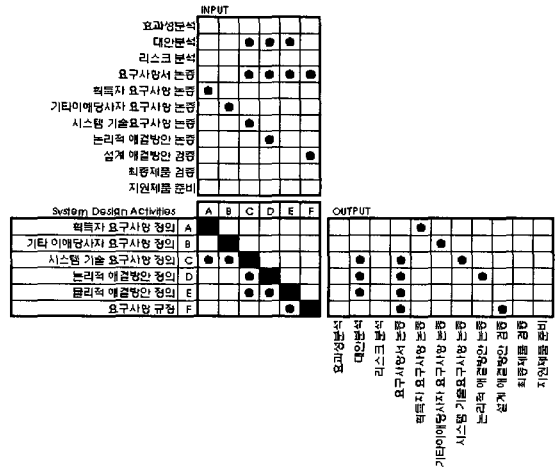
프로세스 또는 업무를 나타내고 있다. 상부 매트릭스에 표시된 '●'는 해당되는 중앙 매트릭스의 시스템설계 업무에 정보가 전달된다는 것을 나타내고 있으며, 우측에 있는 매트릭스의 표시된 '●'는 해당되는 시스템설계 업무로부터 정보가 전달되어 온다는 것을 나타내고 있다.

연구의 대상이 되는 시스템설계 프로세스인 중앙 매트릭스와 이와 연관된 프로세스 또는 업무와의 입력 및 출력 인터페이스 관계가 명확하게 나타남으로써, 최상부 수준에서의 정황분석이 완료된다. 정황분석 결과로, 시스템설계 업무와 인터페이스 되는 프로세스는 획득 및 공급, 기술관리, 제품 구현 및 배치와 그리고 기술평가이다. Figure 3에서 보듯이, 획득 및 공급, 기술관리와 기술적 평가는 시스템설계 업무의 입력을 제공하는 업무이며, 제품구현 및 배치와 기술적 평가는 시스템설계 업무의 결과를 받아 이루어지는 업무이다. 결과에서, 기술적 평가는 시스템설계 업무에 입력과 시스템설계 업무로부터의 결과를 받고 있음을 확인할 수 있다. 따라서 정황분석은 이루어졌으나, 시스템설계 업무의 입출력 관계가 명확하지 않음을 알 수 있다. 획득 및 공급은 시스템 설계 업무에 입력 정보를 제공하고 있으며, 시스템 설계 업무의 결과가 제품구현 및 배치에 영향을 미치고 있다는 것을 알 수 있다. 이는 시스템 설계 업무의 입력과 출력이 명확하게 정의되었다.

2.2 업무 분해 및 입출력 정의

Figure 3으로부터 분석된 결과에서 기술적 평가 프로세스의 입력 및 출력관계를 명확히 하기 위해, 기술적 평가 프로세스를

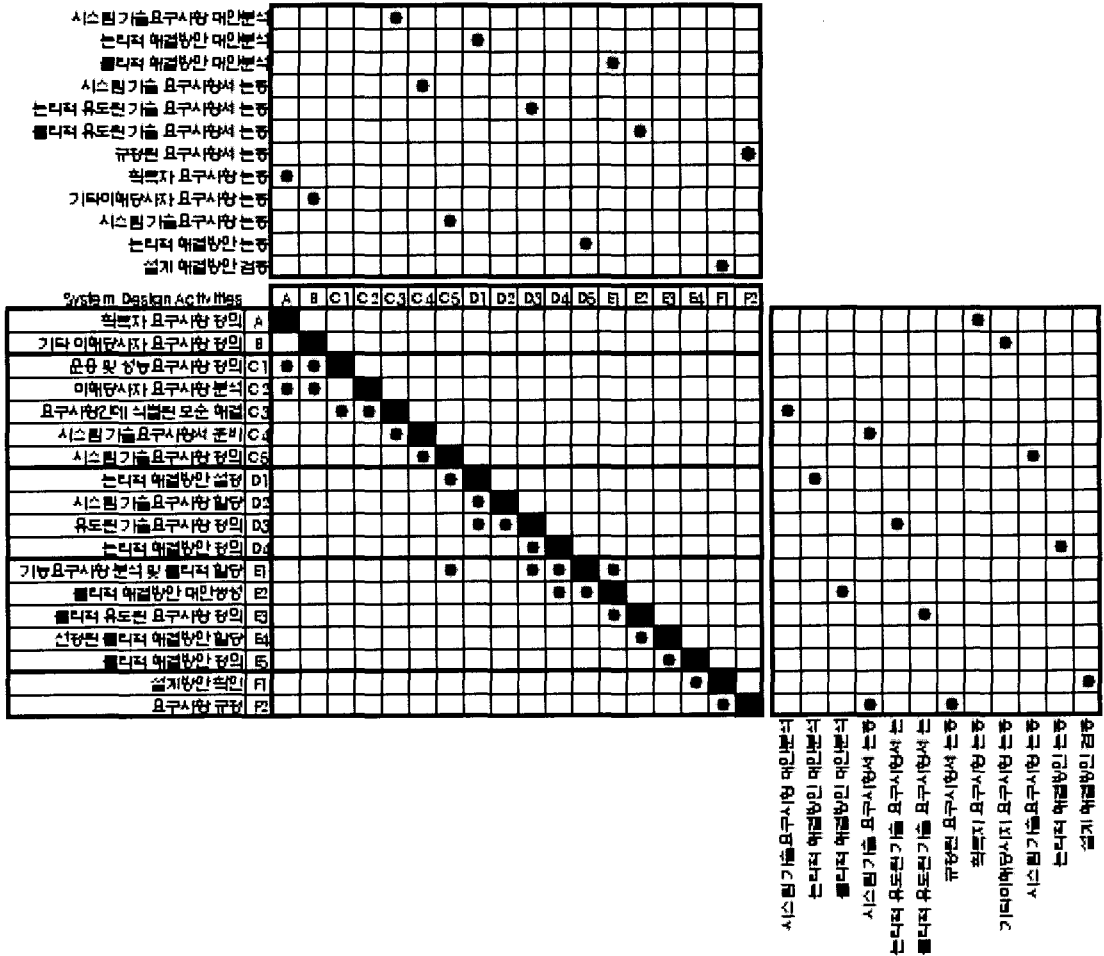
하부 업무로 분해하였다. 기술적 평가 프로세스의 하부 업무 분해 기준은 시스템 설계 프로세스로부터 업무를 식별하는 방법과 동일하게, EIA 632에서 제시하고 있는 ‘반드시 수행해야 할(Shall)’ 요구사항을 기준으로 식별하였다. 분해된 기술적 평가 프로세스의 하부업무는, 효과성분석, 대안분석, 리스크분석, 요구사항서 논증, 획득자 요구사항 논증, 기타이해당사자 요구사항 논증, 시스템 기술요구사항 논증, 논리적 해결방안 논증, 설계 해결방안 검증, 최종제품 검증, 지원제품 준비



<Figure 4> 시스템 설계 업무의 입출력

기술적 평가 프로세스의 나머지 업무는 시스템 설계 프로세스 업무와 정보흐름을 갖지만, 기술적 평가 프로세스에서 대안분석과 요구사항서 분석 업무가 시스템 설계 프로세스 업무의 여러 업무로부터 결과를 받고, 다시 시스템 설계 프로세스 업무로 입력되고 있다. 이를 해결하기 위해, 추가적인 업무 분해가 이루어졌다. 추가적인 업무 분해는 EIA 632에서 제시되는 ‘고려해야 할(should)’ 과업을 대상으로 하였다. 이 때, 업무 분해는 시스템설계 업무 및 기술적 평가 업무 모두를 하향 분해하였다. 이는 시스템 설계 프로세스 업무만 분해했을 때, 시스템 설계 업무에서의 입력 및 출력되는 정보 흐름 관계가 명확하게 개선되지 않았기 때문이다. 이와 같이 하부로 분해된 결과를 Figure 5에 나타내었다. 시스템설계 업무에서 나가는 정보와 시스템 설계 업무로 유입되는 정보가 입출력 중복 없이 명확하게 표현되고 있음을 확인할 수 있다.

Figure 4에서 입출력관계를 분석한 결과, 효과성 분석, 리스크 분석, 최종제품 검증, 지원제품 준비 업무는 시스템 설계 업무와 직접적인 정보흐름이 나타나지 않고 있다. 효과성 분석과 리스크 분석은 시스템 설계 업무에 있어서 반드시 필요한 과정이다. 그런데 시스템설계 업무와 인터페이스가 없는 이유는 대안분석 업무에서 효과성 분석과 리스크 분석 업무가 연동되기 때문이다. 즉, 대안분석 업무가 이루어진다는 것은 효과성 분석과 리스크 분석이 함께 이루어진다.



<Figure 5> 시스템 설계 업무의 분해

2.3 통합 및 동시수행 업무 정의

Figure 5에서와 같이 시스템설계 업무와 이와 연관된 업무간의 입력 및 출력 인터페이스가 명확하게 표현된 시스템 설계 업무는 하나의 매트릭스로 통합될 수 있는데, 먼저 단순히 시스템설계 업무와 입출력 업무를 하나의 매트릭스에 모으는 과정으로, 종합(aggregation)하는 결과를 Figure 6에 나타내었다. Figure 6은 figure 5에서 정의

된 시스템 설계 업무를 먼저 나열하고, figure 5의 상부 및 우측의 업무를 시스템 설계 업무에 이어서 나열하였다. 그리고 입출력 정보를 DSM의 시계반대방향 규칙에 맞춰 Figure 6에 나타내었다.

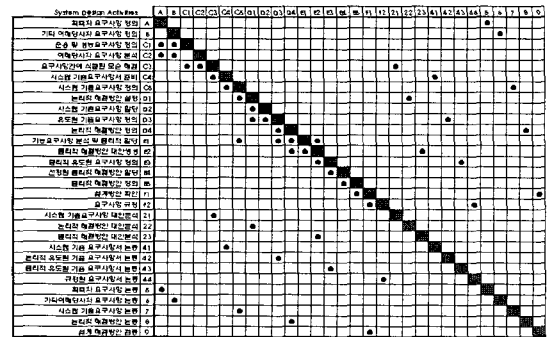
다음은 Figure 6의 결과로부터 업무 반복을 최소화하기 위한 DSM 방법이 적용된다. 이러한 과정은 Excel[®]과 같은 상용 프로그램을 통해 이루어질 수도 있지만, PSM[®]도

구는 DSM 방법을 구현하는 도구로써, 업무 반복을 최소화시키기 위해 정의된 업무 순서를 자동으로 최적의 상태로 변경시킬 수 있는 기능을 갖고 있다. 이 도구를 활용하여 Figure 7과 같은 결과를 얻었다. 즉, Figure 6에서 우측상단에 위치한 정보의 피드백이 대각선 방향의 업무에 인접할 수 있도록 업무의 순서가 바뀌면서, Figure 7과 같이 정보의 피드백이 최소화된 결과를 얻게 되었다.

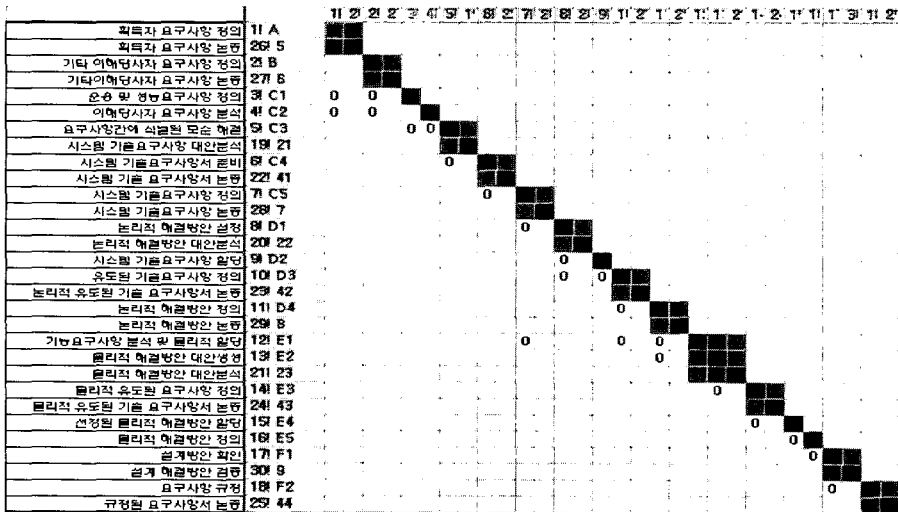
Figure 7의 결과를 분석해 보면, 기술적 평가 업무와 시스템 설계 업무가 통합된 결과는 업무 간에 상호의존적 관계가 많으며, 좌측 상단에서 우측 하단까지 대각선에 밀집된 업무 흐름을 나타내고 있다. 즉 업무적 낭비가 많이 발생하는 피드백이 큰 업무 흐름이 제거되었다. 상호의존적 관계가 많은 것은 Tyson(1998)과 Clausing(1994)에

의해 분류된 바 있는 의도적인 업무 반복에 해당된다. 표준 프로세스인 관계로 이러한 결과는 당연하다고 판단할 수 있을 것이다.

이러한 결과를 바탕으로 Figure 7의 결과는 순차적 업무 흐름과 일정 계획을 수립할 수 있는 기초 자료가 된다.



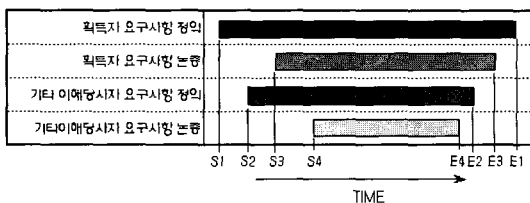
<Figure 6> 종합된 관련업무



<Figure 7> 통합된 시스템 설계 프로세스의 순차적 흐름

2.4 일정계획

Figure 7에서 정의된 동시적이며, 순차적 업무 순서를 통해 일정 계획을 수립하기 이전에 먼저, 업무가 동시적으로 수행될 수 있는 관계를 식별해야 한다. Figure 1에서 정의된 DSM의 활동 및 관계에서 독립적 관계와 상호의존적관계가 업무수행이 동시 또는 중첩(overlapping)으로 진행될 수 있는 관계라는 점을 착안하여, Figure 7에서 획득자 요구사항 정의업무와 획득자 요구사항 논증 업무가 상호의존적 관계임을 알 수 있다. 또한, 운용 및 성능 요구사항 정의 업무와 이해당사자 요구사항 분석업무는 독립적 관계이며, 획득자 요구사항 정의 업무 및 획득자 요구사항 논증 업무와 이해당사자 요구사항 정의 업무 및 이해당사자 요구사항 논증 업무는 다시 독립적 관계이다. 이러한 관계를 시계열적인 간트차트로 표현하는 경우에 Figure 8에서와 같이 표현할 수 있다.

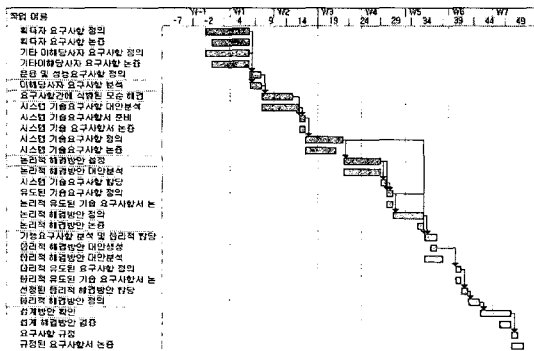


<Figure 8> 동시적 업무의 계획

Figure 8에서 표현된 업무들의 시작(Start)과 끝나는(End) 시간을 S와 E로 표현하였다. 상호의존적 관계에 있는 획득자 요구사항 정의와 획득자 요구사항 논증 업무의 경우, S1에 획득자요구사항 정의가 먼저 진행된다면, 이후 획득자요구사항 논증이

S3에서 진행되어야 한다. 종료되는 시점인 E1과 E3에서는 E3결과가 E1에 통보됨으로써 종료되기 때문에 E3시점은 항상 E1보다 늦거나 동일 할 것이다. 따라서 E1이 E3보다 뒤늦게 종료되도록 계획되어야 할 것이다. 이러한 상호의존적 관계는 기타 이해당사자 요구사항 정의와 기타이해당사자 요구사항 논증에서도 동일하다. 상호의존적 관계에 있는 업무는 그들 간의 관계에 따라 일정계획이 수립된다.

획득자 요구사항 정의 및 획득자 요구사항 논증 그룹과 기타 이해당사자 요구사항 정의 및 기타이해당사자 요구사항 논증 그룹과는 독립적 관계이다. 한 예로, S1에의 시작시점과 S2의 시작 시점은 상호 구속하지 않는다. 또한 종료되는 시점인 E1과 E2는 서로 구속하지 않는다. 그러나 독립적 관계의 그룹은 해당업무의 선행 또는 후행되는 업무로부터 구속을 받을 수 있다. 예를 들면, Figure 7에서 획득자 요구사항 정의와 기타이해당사자 요구사항 정의 둘 다의 결과물은 후행 업무인 운용 및 성능요구사항 정의 업무의 입력 정보가 된다. 따라서 획득자 요구사항 정의와 기타이해당사자 요구사항 정의 둘 다의 결과물이 입력되어야 운용 및 성능요구사항 정의 업무 시작될 수 있는 것이다. 이는 Figure 8의 E1 및 E2 시점이 운용 및 요구사항 정의 업무 시작 전까지 종료되도록 일정을 계획해야 한다. 이와 같이 상호의존적 관계나 독립적 관계를 포함한 Figure 7의 전체 업무는 동시적 또는 순차적인 일정으로 계획될 수 있다. 이를 Microsoft사의 Project®도구에서 간트차트로 나타낼 수 있는데, 그 결과를 Figure 9에 나타내었다.



<Figure 9> 일정계획 예제

Figure 7에서 정의된 업무를 Project®의 작업이름에 옮기고, 각 작업에서 필요로 하는 작업시간을 기입한 후, 작업 간의 순서에 따른 정보 흐름을 Project®에서 제공하는 작업간 링크 기능을 활용함으로써, 쉽게 일정계획이 수립된다. Figure 7에서 정의된 업무로부터 단위작업시간 및 일정을 정의하고 Figure 7에서 정의된 업무관계에 대한 정보는 Figure 9와 같은 결과를 얻게 한다.

Figure 9의 결과로부터 알 수 있는 것은, 시스템의 규격을 생성하는 과정에 시스템을 논증(Validation)하거나 검증(Verification)하는 업무가 밀접하게 시스템설계 업무에 포함되어 있다는 점이다. 통상 이러한 업무를 책임지고 있는 팀은 품질관리나 품질보증팀으로써, 제품이 통합되어 제작되는 단계에서, 시스템 규격의 타당성을 요구하는 경우가 종종 발생한다. 그러나 Figure 9에서 제시된 업무순서는 시스템 설계를 정의하는 단계에서부터, 품질관리나 품질보증팀과 밀접하게 동시에 업무를 수행되도록 계획됨을 볼 수 있다.

3. 결론

모든 기업은 시스템을 개발하기위해, 기업 정책과 절차에 적합한 구현 프로세스를 계획해야 한다. 본 연구는 시스템을 엔지니어링하는 EIA632 표준으로부터 시스템을 설계하는 프로세스를 구현 가능하도록 하부 업무와 이들의 순차적 흐름을 정의하였다. 본 연구의 상세 결과는 정황분석에서부터 하부 업무로 분해하면서 업무의 입출력을 정의하는 프로세스를 제시했다. 정의된 업무들은 DSM방법에 의해 간략한 형태로 표현되며, 정황분석과 업무분해 및 순차적/동시적 업무설계가 구현되었다. 이러한 결과는 통합되어, 통상적인 일정계획으로 수립되어 잘 정리된 순차적/동시적 업무 계획을 수립하였다.

본 연구의 절차와 방법은 EIA632에만 해당되는 것은 아니다. 특히, 기업이나 기관에서 제품이나 서비스의 품질인증이나 품질향상을 목적으로 ISO계열, APQP, 6시그마, CMMi 등을 기업 절차에 통합하고자 하는 경우에 본 연구 절차와 방법은 프로세스 업무를 정의하고 통합하여 효과적이고 효율적인 업무계획을 수립하는데 확장하여 적용할 수 있을 것이다.

4. 참고문헌

- [1] Clausing, Donald(1994), "Total Quality Development: A Step-by-Step Guide to World-Class Concurrent Engineering", ASME Press New York.
- [2] Dennis M. Buede(1999), The Engineering Design of Systems, John Wiley & Sons Inc., New York.
- [3] EIA(1999), EIA 632: Process for

Engineering a System, Electronic Industries Alliance, Arlington.

- [4] INCOSE(2000), *Systems Engineering Handbook*, INCOSE, Seattle.
 - [5] James N. Martin(2000), "Processes for Engineering a System : An Overview of the ANSI/EIA 632 Standard and Its Heritage", *Systems Engineering*, Vol 3, pp 1-37
 - [6] Tyson R. Browning(1998), *Modeling and Analyzing Cost, Schedule, and Performance in Complex System product Development*, Doctor of Philosophy in Technology, Management, and Policy, Massachusetts Institute of Technology.
 - [7] Tyson R. Browning(2002), "Process Integration Using the Design Structure Matrix", *Systems Engineering*, Vol. 5, No. 8, pp 180-193
 - [8] Tyson R. Browning and Steven D. Eppinger(2002), "Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development", *IEEE Transactions on Engineering Management*, Vol. 49, No. 4, pp 428-442
-