

# 데이터흐름도(DFD)의 SysML 다이어그램으로의 변환에 관한 연구

윤석인<sup>1\*</sup>, 왕지남<sup>1</sup>  
<sup>1</sup>아주대학교 시스템공학과

## Transformation from Data Flow Diagram to SysML Diagram

Seok-In Yoon<sup>1\*</sup> and Ji-Nam Wang<sup>1</sup>

<sup>1</sup>Division of Systems Engineering, Ajou University

**요 약** 과학기술의 발전으로 인해서 현대의 시스템은 과거에 비해 대형화되고 복잡화되었다. 복잡한 시스템 개발에서 복잡도를 줄이기 위한 방법으로 모델기반 시스템엔지니어링(MBSE)이 다양한 분야에서 적용되고 있다. 하지만 다양한 언어를 통하여 모델링이 이루어지고 있으므로, 개발 프로세스에 따른 산출물의 일관성 및 이해관계자간의 의사소통에 문제가 발생하고 각 MBSE 도구의 지원 다이어그램의 차이에 따라 산출물의 표현이 제한되기 때문에 그 효과성이 부족하다. 본 논문에서는 전통적 다이어그램의 하나인 데이터흐름도(DFD)를 SysML의 다이어그램으로 변환하는 법칙을 연구함으로써 이미 구축된 모델들을 재사용하여 SysML로 구축될 수 있도록 한다. 특히, 각 다이어그램의 메타모델(Meta Model)을 분석하고, 이분그래프를 통해 각 구성요소의 연결 관계를 확인하여 변환의 근거를 마련하고 무결성을 확인하며 보다 효과적인 변환 법칙을 제시하고 있다. 또한 위 내용을 함정전투체계설계에 적용해 봄으로써 그 효과성을 확인한다. 본 연구의 결과를 기반으로 향후 추가연구를 수행하면, 이미 구축된 시스템 개발과정에서 산출된 모델들을 SysML 을 이용하여 일관성 있게 표현할 수 있을 것 이다. 그리하여 산출물의 일관성을 높이고 이해관계자간의 의사소통을 원활하게 함으로써, SysML을 이용한 효과적인 모델기반시스템엔지니어링(MBSE)에 도움이 될 것으로 기대된다.

**Abstract** Due to science and technology evolutions, modern systems are becoming larger and more complex. In developing complex systems, Model-Based Systems Engineering (MBSE), which is approach to reduce complexity, is being introduced and applied to various system domains. However, because of the modeling being made through a variety of languages, there is a problem with communication within the stakeholders and a lack of consistency in the models. In this paper, by investigating the rule explaining the transformation of one of the only traditional diagrams, DFD, to SysML and reusing the formerly built models, we attempt to implement by SysML. Analyzing each diagram's Metamodel and validating the connection of each component through bipartite graph especially suggest an effective transformation rule. Also, by applying to naval-combat system, we confirm efficiency of this study. Establishing the results of this study as basis for conducting further study, we will be able to transform other previous models gained from formerly built system to SysML. In this way, the stakeholder's communication can be improved and we anticipate that the application of SysML will be beneficial to the much efficient MBSE.

**Key Words** : DFD(Data Flow Diagram), MBSE(Model-Based Systems Engineering), Systems Engineering, SysML(Systems Modeling Language), Transformation

### 1. 서 론

현대의 모든 시스템은 과학기술의 급속한 발전으로 인

해 과거의 것에 비해 대형화, 복잡화되는 추세이다. 이러한 복잡한 시스템의 설계를 위해서는 모델기반 시스템엔지니어링(MBSE, Model-Based Systems Engineering)의

\*Corresponding Author : Seok-In Yoon(Ajou Univ.)

Tel: +82-10-8921-8765 email: seokin25@naver.com

Received October 4, 2013

Revised (1st October 11, 2013, 2nd October 25, 2013)

Accepted November 7, 2013

적용이 불가피하다. 특히, MBSE는 MBSE 도구를 이용하여 각 개발 단계의 산출물을 데이터베이스화함으로써, 데이터에 대한 접근성을 높이고 설계에서부터 제작 및 설치 단계에 이르기 까지 일관성 및 추적성을 확보할 수 있다[1].

지금까지의 개발에서는 보편적으로 전통적 다이어그램으로 대표되는 DFD, FFBD, IDEF 등의 다이어그램이 사용되었다. 하지만, 산출되는 다이어그램들의 표준화가 되어있지 않았고, 타 분야의 전문엔지니어와의 의사소통에도 문제가 있었다. 이에 INCOSE(International Council On System Engineering) 와 OMG(Object Management Group)에서는 SysML(System Modeling Language)을 만들어 표준화 된 모델링 언어를 MBSE 프로세스에 적용이 되게 하였다. SysML은 객체기반의 접근법을 갖는 UML2를 기반으로 한 모델링언어로서, 시스템엔지니어와 소프트웨어 엔지니어 및 각 분야 엔지니어를 비롯한 많은 이해관계자들 간의 의사소통에 도움이 된다[4].

현재까지 개발된 시스템들의 산출물은 대부분이 전통적 다이어그램으로 기술되었고, 각 MBSE 도구에서 지원 하는 다이어그램도 각자 차이가 존재한다.

이에 각 MBSE 도구에 표준 언어인 SysML 적용이 확장됨에 따라 모델의 재사용성 증대 및 SysML을 이용한 효과적인 MBSE의 적용을 위해서는 다이어그램의 변환 방법에 대한 연구가 필요하다.

본 논문에서는 요구분석단계에서 구조적 분석을 통해 시스템 내에서의 프로세스들 간의 데이터 흐름을 기술하는 다이어그램인 데이터흐름도(DFD, Data Flow Diagram)를 SysML 다이어그램으로 변환하는 법칙을 제안함으로써 기존의 데이터베이스화 된 모델의 재사용성을 높이고 SysML을 통한 MBSE를 보다 효과적으로 할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 서론에서는 현재 동향과 필요성을 제시하였고, 2장에서는 관련 선행연구, 3장에서는 다이어그램을 비교하여 변환법칙의 근거를 확인하고, 4장에서 변환법칙을 이용하여 함정전투체계 상부 수준 설계에 적용해 보았고, 5장에서는 연구의 결과를 정리 및 요약 하였다.

## 2. 관련 연구

Fanchao Meng 은 그의 논문에서 DFD를 각node와 flow를 구분하여 1:1 변환을 방법으로 UML2.0의 Activity Diagram으로 변환하는 방법에 대한 연구를 하였다. 또한 이를 판매시스템에 적용하여 확인하였다[2].








하지만 Activity Diagram은 시간의 흐름을 표현하는데 반해서 DFD는 시간적 흐름의 순서가 명확하게 표현이 불가능하고, 이를 공학적 시스템에 적용할 시에 Activity Diagram 으로의 변환과정에서 누락되거나 새롭게 생성되는 노드들이 있기 때문에 이 변환은 논리적 오류를 범할 수 있다. 그래서 보다 나은 변환방법에 대한 연구가 필요하다.

### 2.1 DFD (Data Flow Diagram)

DFD는 소프트웨어 및 시스템 분석 및 설계에서 사용되는 보편적인 시스템 모형화 도구로써, 데이터에 비해 기능이 매우 복잡하고 중요할 경우에 유용하게 쓰인다. 이는 데이터가 소프트웨어, 혹은 시스템 내의 각 프로세스를 따라 흐르면서 변환되는 모습을 나타낸다. 하지만 시간에 따른 수행을 명확히 나타낼 수 없고, 반복적인 기능 수행을 쉽게 나타낼 수 없으며 기능수행의 성공 여부를 나타낼 수 없다는 단점을 갖고 있다.

DFD의 구성요소는 크게 네 가지가 있다. 첫째로 “Process”는 입력되는 데이터를 원하는 데이터로 출력시키기 위한 과정이다. 프로세스는 자체적으로 데이터를 생성할 수 없고 항상 입력되는 데이터가 있어야 한다. 둘째, “Flow”가 있다. 이는 구성요소들 간의 인터페이스를 나타낸다. 표기는 화살표를 이용하고 후속작업의 참조를 위해 명칭을 기록해두는 것이 바람직하다. 셋째, “Store” 는 저장되어있는 정보의 집합이다. 이는 단순한 데이터의 저장을 나타내는 것이지 데이터의 변동을 표시하는 것이 아니다. 넷째, “External Entity”는 프로세스 처리과정의 데이터 발생의 시작 및 종료를 나타낸다. 이는 DFD 상에서 프로세스와의 상호관련성을 표시하며, 일반적으로 DFD범위 밖에 사각형 형태로 나타낸다. 도시해 보면 Table 1과 같다[3].

[Table 1] DFD element and symbol

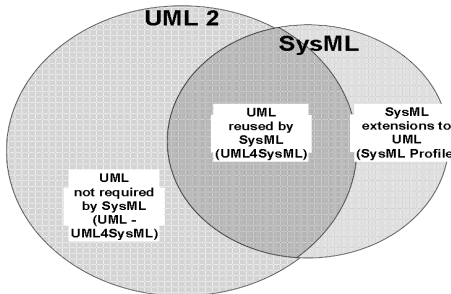
element		symbol
Process	Data Process	
	Control Process	
Store	Data Store	
	Control Store	
External Entity		
Flow	Data Flow	
	Control Flow	

### 2.2 SysML (System Modeling Language)

SysML은 하드웨어, 소프트웨어, 정보, 인원, 절차, 시

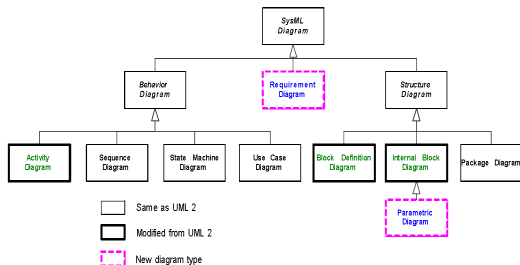
설 등을 포함한 복잡한 시스템의 사양화, 분석, 설계, 검증 등을 위한 그래픽 모델링 표준 언어로서, OMG (Object Management Group) 에 의해 그 사양이 책정되고 미국의 경우, 군사, 우주항공, 자동차 등의 다양한 분야에서 폭넓게 이용되고 있다[4].

SysML은 UML (Unified Modeling Language) 을 기반으로 하여 만들어졌고, 그 관계는 Fig. 1과 같다. 이로 인해 SysML은 몇 가지 특징을 갖는다. 첫째, 시각적인 모델링언어이다. 모델을 사용하여 사물을 시각적으로 표현하는 것은 다양한 효과를 준다. 복잡한 사물을 추상적으로 파악하는 방법이 쉬워지고, 여러 개의 다른 시점으로부터 모델링을 하게 되므로 일방적인 사고에 사로잡히지 않고 사물을 다양한 시각으로 분석할 수 있게 해준다. 둘째는 표준 언어로서의 효과이다. 표준 언어를 사용하면 정보의 공유가 쉽다. 문화적, 언어적 차이가 존재를 해도 표준 언어로 표현된 문서에 대한 이해는 공유할 수 있게 된다. 셋째로는 단순한 언어사양이 주는 효과이다. 이는 시스템개발 시 각 분야의 엔지니어들 사이에서의 이해를 공통화 할 수 있다. UML은 지금까지 주로 소프트웨어 개발 분야에서만 활용되어 왔으며, 그 사양 또한 단순하다고는 볼 수 없다. 하지만 SysML은 UML에 비해 단순하기 때문에 기억, 이해하기 쉽다[5].



[Fig. 1] Relationship between SysML and UML

SysML의 다이어그램은 거동다이어그램 4개, 구조다이어그램 4개, 요구사항 다이어그램 1개, 총 9개로 구성되며, 이들과 UML과의 관계는 Fig. 2를 보면 알 수 있다.



[Fig. 2] SysML Diagrams

### 2.2.1 UCD (Use Case Diagram)

UCD는 SysML의 거동다이어그램 중의 하나로서, UML2의 UCD와 동일하다.

UCD는 객체지향 모델링에서 시스템과 외부시스템 (actor)간의 인터페이스를 나타내기 위해 사용되며, use case는 시스템이 수행해야 할 최상위 서비스이다. 시스템 엔지니어는 use case를 기반으로 세부 설계를 수행해 나간다.

UCD의 구성요소는 5가지가 있다. 첫째로 “Actor”는 시스템을 촉진하거나 시스템으로부터 촉진되는 외부시스템의 일부를 나타낸다. “Use Case”는 하나이상의 Actor와 시스템간의 상호작용을 나타낸다. 셋째로 “Association”은 Actor와 Use Case간의 링크를 나타내고, “Relationship”은 Use Case간의 관계를 나타낸다. 마지막으로 “System Boundary”는 대상 시스템의 범위를 나타낸다. 도시화하면 Table 2와 같다[3].

[Table 2] UCD element and symbol

element	symbol
Actor	
Use Case	
Association	
Relationship	Include
	Extend
System Boundary	

### 2.2.2 BDD (Block Definition Diagram)

BDD는 SysML의 구조다이어그램 중의 하나로서, 가장 폭넓게 쓰인다. BDD는 UML2의 클래스 개념의 확장인 시스템과 그 구성요소를 나타내는 ‘Block’들 간의 관계를 정의하기 위한 다이어그램이다.

BDD의 구성요소는 크게 5가지가 있다. 시스템 및 구성요소를 나타내는 “Block”, Block 간의 관계를 나타내는 “Relationship”, Block 사이의 인터페이스를 나타내는 “Interface”, Block과 Relationship사이를 연결해주는 “Port”, Block 사이에서의 정보 및 신호의 흐름을 나타내는 “Item flow” 가 있다. 도시화하면 다음 Table.3과 같다 [5].

[Table 3] BDD element and symbol

element		symbol
Block		
Relationship		
Interface		
Port	Standard	
	Flow	
Item	flow	

### 3. 다이어그램 변환

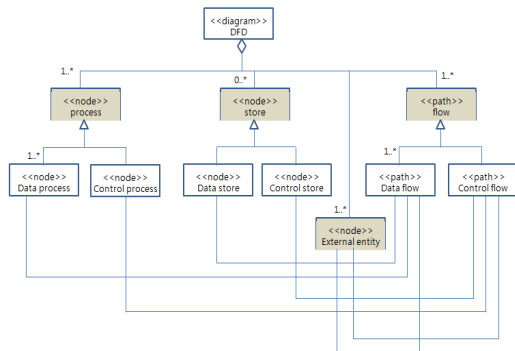
본 연구에서는 다이어그램 간의 변환과 변환 시 정보 손실에 관한 무결성을 확인하기 위하여 메타모델 (Meta Model)과 이분그래프를 이용하였다.

먼저, 각 다이어그램의 메타모델을 분석하였다. 그 다음, 분석된 메타모델을 기반으로 다이어그램의 핵심이 되는 구성요소를 도출, 비교하였고, 마지막으로 이분그래프를 이용하여 각 구성요소간의 연결 관계를 비교하여 다이어그램의 변환 법칙에 관한 결론을 얻어내었다. 이를 통해 변환의 근거와 절차를 정형화 할 수 있다.

#### 3.1 Diagram Meta Model 분석

앞서 언급한 바를 토대로 DFD, UCD와 BDD는 모두 최상위레벨에서의 시스템과 외부시스템의 관계를 설명하는 Context Diagram으로 사용이 된다는 것을 추론할 수 있다. 이는 많은 연구결과 및 자료들에서 확인할 수 있다.

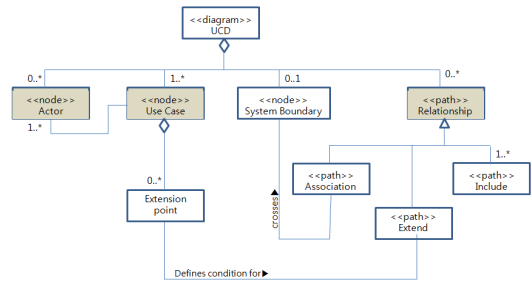
우선 DFD의 메타모델을 살펴보면 Fig. 3과 같다.



[Fig. 3] DFD meta model

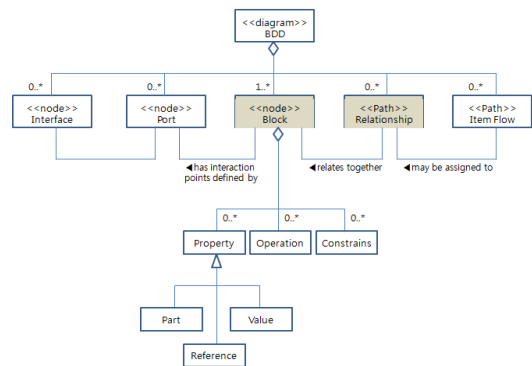
DFD의 메타모델에서는 DFD의 구성요소를 Node 와 Path 로 나누어 분석하였고, Node와 Path의 연결 관계 또한 볼 수 있다. 또한 반드시 1개 이상 포함되어야 하는 핵심 구성요소를 분석하여 음영으로 나타냄으로써 구성요소 비교 시 이용할 수 있다.

DFD와 동일한 방법으로 SysML의 UCD 메타모델을 분석하면 Fig. 4 와 같다[5].



[Fig. 4] UCD meta model

BDD의 Meta Model을 살펴보면 Fig. 5와 같다[5]. 메타모델을 분석하고 핵심 구성요소를 도출하여 음영처리를 하여 나타내었다.



[Fig. 5] BDD meta model

위의 세 다이어그램의 메타모델 분석을 통하여 구성요소의 속성 및 핵심 구성요소와 구성요소간의 연결 관계를 확인할 수 있게 되었다.

#### 3.2 Element 비교

메타모델 분석에서 도출 된 각 다이어그램의 구성요소 에서 Node를 비교해보면, DFD의 Process와 UCD의 Use Case, BDD의 Block은 서로 나타내는 정보가 같다. 또한

External Entity 와 Actor도 동일하다고 분석된다. BDD에서는 이 또한 Block 으로 나타낼 수 있다. Store는 Use Case/Block으로 나타낼 수 있다. Path를 비교해보면, DFD의 Flow 은 Relationship로 나타낼 수 있다. 이와 같은 비교를 통해 핵심 구성요소의 성질만으로는 일대일 변환이 가능한 것으로 나타난다. 이를 정리하면 아래의 Table 4와 같다.

[Table 4] comparison of diagram element

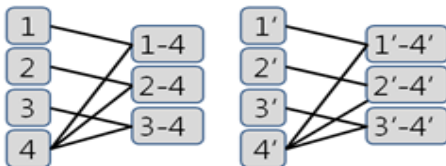
BBD	DFD	UCD
Block	Process	Use Case
Block	External Entity	Actor
Block	Store	Use Case
Relationship	Flow	Relationship

### 3.3 연결 관계 비교

구성요소간의 연결 관계의 비교를 위해 이분그래프(Bipartite Graph)분석방법을 사용한다.

도식화의 편의를 위해 위 Table 4의 DFD 구성요소를 상단에서부터 1,2,3,4, UCD의 구성요소를 1',2',3',4'로 표시한다. BDD는 모두 Block으로 도식되므로 이분그래프 분석을 할 필요가 없다.

우선 DFD와 UCD의 Element를 이분그래프로 나타내 보면 Fig. 6과 같다.



[Fig. 6] Bipartite graph of DFD and UCD element

Fig. 6을 보면, 구성요소가 각각 4와 4' 와 완전한 이분 그래프를 이룬다. Table 3에서 알 수 있듯이 1' 과 3' 은 동일하게 “Use Case”를 나타내기 때문에 “1’-3’은 불필요하지만, 완전한 이분그래프의 확인을 위해 삭제하지 않았다.

이분그래프 분석을 통한 연결 관계 비교에서 DFD의 구성요소가 갖고 있는 연결 관계가 UCD, BDD의 구성요소 연결 관계와 동일하다는 것을 알게 되었다.

### 3.4 다이어그램 비교

메타모델을 이용한 구성요소의 비교분석 및 이분그래프

프를 이용한 구성요소간의 연결 관계 비교를 통해서 DFD는 SysML의 UCD 와 BDD으로 변환할 경우 다이어그램의 정보가 누락되거나 새롭게 생성되지 않고 1:1 변환이 이루어진다는 것을 확인하였다.

하지만 DFD를 변환하여 데이터베이스를 구축할 경우, UCD로 변환할 것인가 BDD으로 변환할 것인가에 대해서는 아직 정의 내리지 않았다.

모델링은 기본적으로 효과적인 시스템의 시각적 표현을 하기 위함이므로 연구자의 선택에 따라 그 방법이 다르겠지만, SysML을 이용한 모델링에서는 시스템과 외부 시스템의 관계를 나타내는 Context Diagram의 경우와 시스템 상부수준 설계에서 각 서브시스템간의 기본관계를 나타낼 경우에 UCD를 이용하여 표현하는 것이 더 효과적인 방법이다[4].

하지만, 각 서브시스템간의 주고받는 정보 및 신호의 흐름이 순서의 의미가 내제되어있거나, 핵심정보일 경우에는 BDD을 이용하여 그 내용을 기록하는 것이 효과적이다.

## 4. 다이어그램 변환 사례 연구

앞에서 언급한 비교를 통하여 데이터흐름도(DFD)가 SysML의 UCD와 BDD으로 변환 될 수 있다는 근거를 마련하였다.

이에 함정전투체계 상부수준 설계에서의 산출물을 바탕으로 변환법칙을 확인해 보았다.

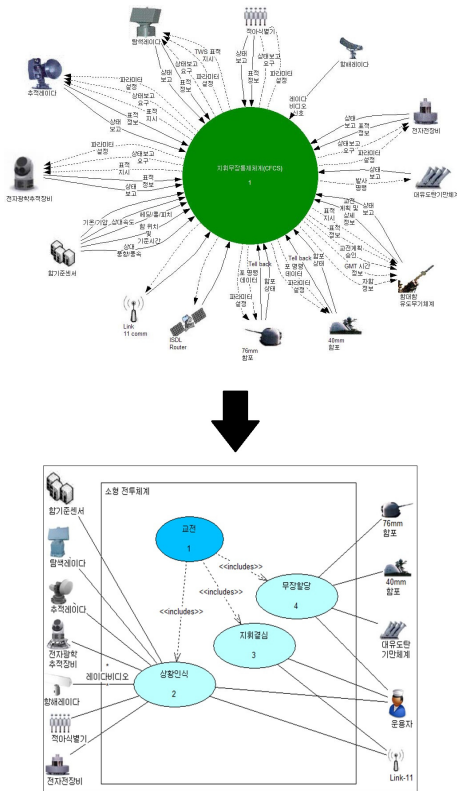
### 4.1 DFD의 SysML UCD로 변환

함정 전투체계 상부수준 설계에서의 정황도 모델링은 정의된 요구사항을 바탕으로 운용적 관점에서의 함정 전투체계와 구성요소의 관계를 나타내는데 그 의미가 있다. 전달되는 정보의 내용은 순서의 의미가 내제되어 있지 않고, 운용적 관점에서의 상부수준 모델링에는 생략되어도 무관하다.

함정전투체계는 기본적으로 지휘무장통제체계와 추적레이더, 탐색레이더, 항해레이더, 적아식별기, 전자전장비, 함포, 유도탄, 센서류, 전자광학추적장비 등의 요소로 구성되며, DFD를 이용한 정황도를 UCD로 변환하면 Fig. 8과 같다.

정황도의 주요 내용은 해당 시스템이 어떠한 외부시스템과 관계되어 있는지를 보여주는 것이다. Fig. 8에서 볼 수 있듯이, 전투체계 상부수준 설계에서의 DFD 정황도를 SysML의 UCD로 변환함으로써 운용적 관점의 정황

표현이 좀 더 간결하고 명확하게 표현 될 수 있다는 것을 보여준다.



[Fig. 8] Transformation from DFD to UCD

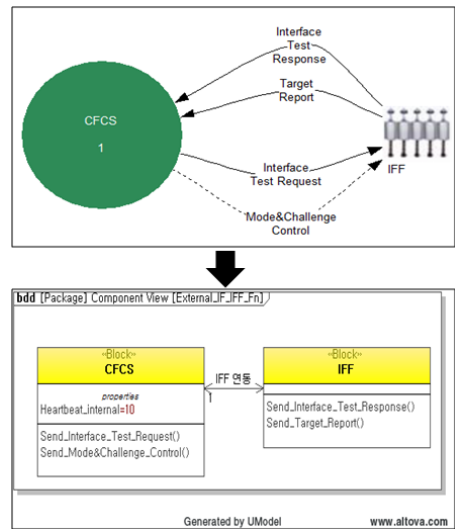
### 4.2 DFD의 BDD로 변환

함정 전투체계 상부수준 설계에서의 기능적 구성을 나타내는 모델은 정황도에 비해 각 구성요소들 간의 주고받는 정보 혹은 신호의 내용과 순서가 표현의 주 내용이 된다. 그러므로 SysML의 BDD으로의 변환이 효과적이다.

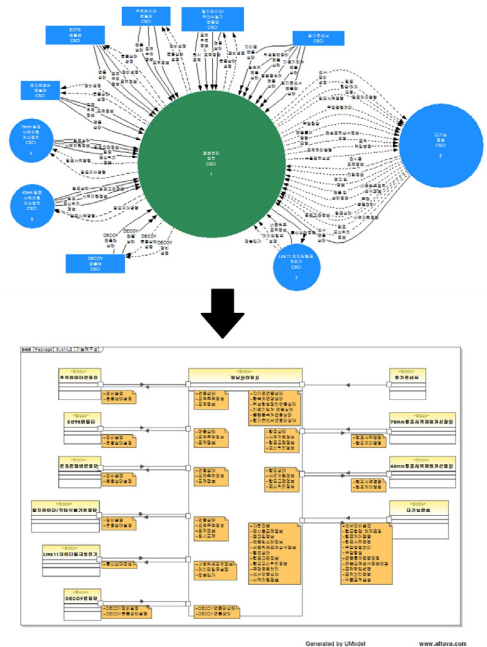
Fig. 9에서는 DFD로 나타내어진 함정전투체계 적아식별기의 기능적 구성을 SysML의 BDD으로의 변환을 보여준다. 이 모델은 상위 수준의 것이므로 BDD의 relationship에 note를 이용하여 주고받는 정보를 표기한다. 이 변환에서 정보의 손실 및 변질은 이루어지지 않았고, 보다 체계적이고 명확한 표현이 가능하다.

Fig. 10은 함정전투체계 상부수준 설계에서의 외부인터페이스의 논리적 인터페이스를 나타낸 DFD를 BDD으로 변환한 그림이다. 표현하는 대상이 하위수준의 인터페이스를 나타내기 때문에 Block 내부의 operation을 이용

하여 정보 및 신호의 손실 없이 주고받는 정보를 정확하게 표현한다. 앞의 두개의 예시와는 달리, Fig. 10은 DFD의 표현이 복잡하지만 구성요소 비교 시 정의된 요소로 변환을 시키고 연결 관계를 나타내어 어려움 없이 변환이 가능하다. 이와 같은 변환은 두 개의 Block간의 관계를 나타낼 때 유용하다.



[Fig. 9] Transformation from DFD to BDD ①



[Fig. 10] Transformation from DFD to BDD ②



## 5. 결론 및 요약

대형화와 복잡도가 증가되는 현재의 시스템 개발과정에서는 모델을 이용한 개발활동이 활발히 이루어지고 있다. 이러한 모델기반시스템공학(MBSE)에서 SysML이라는 공통의 표준 모델링 언어가 존재함에도 불구하고, 전통적 모델링 언어를 사용하여 시스템 개발을 함으로써 개발 프로세스의 효율적이지 못한 산출물과 비효율적인 문제들을 야기한다.

본 연구에서는 이러한 문제의 가능성을 줄이고, SysML을 통한 효과적인 MBSE를 위해서 이미 구축되어 있는 모델을 SysML로 변환하는 방법에 관해 연구하였다.

대표적인 전통적 다이어그램 중의 하나인 DFD를 SysML 다이어그램으로 변환하는 방법을 찾기 위해서 각 다이어그램의 메타모델(Meta model)을 분석하고, 이분그래프를 이용한 다이어그램 구성요소를 비교함으로써 변환 법칙에 관한 근거를 마련하였다.

본 연구에서는 매우 복잡한 구성요소와 인터페이스를 갖는 함정전투체계 상부수준 설계에서의 DFD를 SysML의 UCD와 BDD로의 변환을 통해서, 구축되어있는 전통적 모델을 사용하여 SysML을 사용한 모델기반시스템공학(MBSE)이 가능하도록 기여하였다고 생각한다.

추후 연구에서는 DFD외의 다른 전통적 다이어그램들의 SysML로의 변환 법칙을 연구함으로써, 시스템 개발 전 과정에 걸쳐 산출되는 모델들을 SysML로 표현하여, 그 활용성을 높이는 연구가 필요할 것이다.

## References

- [1] Yoon-Mi Kim, Kyoung-Jin Choi, Yeon-Ok Cho, Developing the Management System of the Constructing Railway Safety Test Facilities Using the Model-Based Systems Engineering Approach, 2009 Proceedings of Autumn Symposium of The Korean Society For Railway, p.1965-1967, 2009
- [2] Fanchao Meng, Transformation from Data Flow Diagram to UML2.0 Activity Diagram: Progress in Informatics and Computing (PIC), 2010 IEEE International Conference, p.1010-1014, 2010  
DOI: <http://dx.doi.org/10.1109/PIC.2010.5687483>
- [3] 3SL, Cradle 6.8 Tutorial, p.19-85, 2013
- [4] OMG Systems Modeling Language (OMG SysML™) v.1.2, p.45-149, 2010
- [5] Jon Holt, Simon Perry, SysML for Systems Engineering:

The Institution of Engineering and Technology, p.27-31,51-59 2008

- [6] G.Butler, P.Grogono, R.Shinghaln, Retrieving Information from Data Flow Diagrams. : Proceedings of 2nd Working Conference on Reverse Engineering, 1995.  
DOI: <http://dx.doi.org/10.1109/WCRE.1995.514690>
- [7] Yong-Cheol Oh, Database Modeling: Freelec, 2010
- [8] Myeong-Seob Cho, "A Study on Process Tailoring for MBSE": 4<sup>th</sup> APCOSE, 2010.

윤 석 인(Seok-In Yoon)

[정회원]



- 2013년 8월 : 아주대학교 시스템 공학과 (공학석사)

<관심분야>

모델기반시스템공학, 함정전투체계

왕 지 남(Ji-Nam Wang)

[정회원]



- 2013년 8월 : 아주대학교 산업공학과 교수, 시스템공학과 교수

<관심분야>

Modeling & Simulation, Neural Network, 제어 시뮬레이션