The Journal of Korean
Institute of CALS/EC
Vol. 2, No. 2, Dec. 1997

CALS 구현을 위한 모델링 방법론의 기능조건

김 철한*, 우 훈식**, 김 중인***, 임 동순****

Functional Requirements about Modeling Methodology for CALS

Cheol-Han Kim , Hoon-Shik Woo , Joong-In Kim , Dong-Soon Yim

Abstract

Modeling methodology has been widely used for analysis and design of a information system. Specially, under the CALS environments, modeling approach is more important because the enterprise functions are inter-related and information sharing speeds up the business . In this paper, we suggest functional requirements about modeling methodology for CALS by surveying the IDEF0 and ARIS. The former is FIPS 183 and the latter is basic methodology of SAP/R3 which is world-wide ERP system.

The proposed functional requirements include all semantics of IDEF0 and adds some features. The first is adding modeling components which are semantic representations. In addition to ICOMs, we add the time and cost component which is required to execute the function. The second is tracing mechanism. When we need some information, we drive the functions related with the information by reverse tracing of the function which produces the information as a output and input. Through the tracing, we find out the bottleneck process or high cost process. This approach guarantees the integrity of data by designating the data ownership. Finally, we suggest the final decomposition level. We call the final decomposed function into unit function which has only one output data. We can combine and reconstruct some of functions such as 'lego block' combination.

Keyword: functional modeling, methodology

---

[1] 이 연구는 시스템 공학연구소의"CALS 구현을 위한 분산 모델링 방법론개발"의 일환으로 진행된 것임
* 대전대학교 산업공학과(chkim@dragon.taejon.ac.kr),
** 시스템공학연구소 CALS연구실(hswoo@seri.re.kr),
*** 홍익대학교 경영정보학과 (jokim@wow.hongik.ac.kr)
**** 한남대학교 산업공학과 (dsyim@eve.hannam.ac.kr)

1. 서론

CALS 구현을 위한 방법론으로 기존의 BPR 방법론이나 시스템 분석 방법론이 사용되어 오고 있다. 그 중 대표적인 것이 IDEF[Bravoco et al. 1989, FIPS 183, FIPS 184]와 ARIS[Sheer 1992, Sheer 1994]라 할 수 있다. 전자는 미 공군에서 개발되어, 현재는 미국의 표준 규격인 FIPS(Federal Information Processing Standards)으로 정착되어 사용되고 있는 방법론이고, 후자는 현재, 전 세계에서 가장 널리 사용되고 있는 ERP 시스템인 SAP/R3의 표준 방법론으로 사용되고 있는 방법론이다. ERP시스템은 기업의 전사적인 자원관리를 목적으로 하는 시스템으로 최근에는 PDM, EDI등과 연계하고, WEB상에서 구현할 수 있도록 개발되고 있어 CALS가 추구하고자 하는 목적과 그 맥락을 같이 하고 있다.

본 연구에서는 이 두 가지 방법론을 분석하여 각각의 장단점을 분석하여 CALS를 구현하고자 할 때 요구되는 방법론에 대한 기능을 정의하고자 하고, 이를 바탕으로 새로운 방법론을 제안하고자 한다. 제안된 방법론은 기능모델링 방법론으로 향후 개발될 정보모델링 방법론, 동적 모델링 방법론과 함께 CALS구현을 위한 방법론의 일부이다.

2. IDEF 방법론

IDEF는 1970년대 미 공군의 ICAM project에 참여하고 있는 사람들이나 기관들간의 원활한 의사소통을 위하여 개발되어 오늘날까지 사용되고 있는 방법론으로 ICAM DEFinition의 약어이다. 미 공군의 ICAM 프로젝트는 군수업체, 연구소, 민간인, 정부 등 다양한 집단이 참여한 프로젝트로 항공산업 생산시스템의 공통기능인 정보관리, 물류관리, Shop Flow Control,등을 위한 기본시스템(Generic System)을 개발하고 이를 발전시켜, 생산성을 향상시키고자 추진되었다.

분석이나 설계하고자 하는 대상시스템을 시스템의 수행에 요구되는 기능과 정보, 그리고 기능들간의 순서등에 의해서 정의하기 위하여 IDEF0 [Bravoco et al. 1989] IDEF1[Wisdom 1985], IDEF2 [U.S. Air Force 1981] 등 3가지로 구성되어 있다. IDEF1은 E-R모델을 기반으로 개발되었지만, IDEF1x로 확장되어, IDEF1은 초기의 요구분석에 IDEF1x는 이를 바탕으로 하는 Database 설계단계의 데이터모델링에 활용하고 있다. 1990년대 들어서, 국방성과 군수산업을 중심으로 하는 CALS라는 개념이 민간산업으로 전파되면서, 미 연방정부에서는 IDEF를 Integration DEFinition으로 정의하고, IDEF0, IDEF1x를 표준방법론(FIPS 183, FIPS 184)으로 정하여 활용하고 있다.

본 연구에서는 세계적으로 널리 사용되

고 있는 IDEF방법론 중에서 모델링의 단순함과 모델의 공유 및 해독의 용이성으로 인하여 보편적으로 사용되고 있는 IDEF0의 방법론에 대하여, 그 특징을 살펴보고, 이 방법론이 가지고 있는 단점을 분석하여 이를 바탕으로 IDEF0를 보완하는 방법론을 제안하고자 한다.

2.1 IDEF0의 배경

ICAM 프로젝트는 다양한 집단의 의사소통의 정형화를 위하여 다음과 같은 조건에 맞는 언어를 개발하고자 하였다.

1. 항공산업의 작업을 표현하고 문서화 할 수 있는 언어로
2. 생산시스템을 표현할 수 있어야 하며,
3. 간결하고, 쉽게 원하는 정도까지 바로 정보를 얻을 수 있어야 한다.
4. 올바른 결과를 보장하기 위하여 충분히 견고하고, 정확하여야 한다.
5. 항공산업에 종사하는 다양한 사람들의 의사소통이 가능하여야 한다.
6. 이 언어의 사용을 위한 방법론을 포함하여야 한다.
7. 조직의 측면에서 가질 수 있는 편견을 제거하기 위하여 조직과 기능을 분리하여 모델링 하여야 한다.

이러한 조건을 만족하는 방법론을 D.T ROSS의 SADT(Structured Analysis and Design Technique)[Ross et al. 1977]가 채택되어 IDEF0라는 명칭으로 정의되어 현재까지 사용되어 오고 있다. 이 방법론의 특징은 기능이 어떻게 실행되는가에 관심을 가지는 것이 아니라, 기능 그 자체가 무엇인지를 규명하는데 관심을 가지고 있다. 따라서, 기능들간의 연계에 의한 프로세스의 설계보다는 단위기능의 구현에 필요한 요소가 무엇인지를 규명하는데 요구되는 의미와 방법을 가지고 있다.

2.2 IDEF0의 기본개념

IDEF0는 셀 모델링 기법으로 출발하였기 때문에 IDEF0 모델링의 특징도 주로 여기에 초점이 맞추어져 있다. 이 방법의 접근방법은 크게 다음의 세가지로 정의될 수 있다.

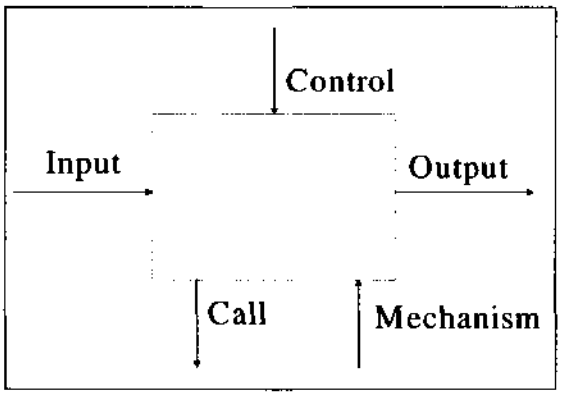
1. 시스템이 왜 필요한지에 대한 정의부분 : 배경분석(CONTEXT ANALYSIS)
2. 시스템이 어떠한 기능을 하는가 : 기능정의(FUNCTIONAL SPECIFICATION)
3. 시스템이 어떻게 구성되어 있는가 : 설계조건 (DESIGN CONSTRAINTS)

IDEF0는 이러한 접근방법에 따라서 IDEF0는 다음과 같은 개념을 내포하고 있다.

- 셀 모델링 방식의 도식적 표현(CELL MODLEING GRAPHIC REPRESENTATION) : IDEF0의 기본적인 표현 수단은 [그림 1]에 나타난 바와 같이, 기능을 나타내는 박스와 기능들 간의 인터페이스(입력/출력/제어 데이터)를 표현하는 화살표로 정의된다. 기능과 인터페이스는 다이어그램 상에 표현되어, 언제, 어떻게 기능이 트리거되고 제어되는 가를 설명하게 된다. 화살표 중에서 박스 아래의 화살표는 기능의 동작이나 작업을 수행하는 사람이나 기계 또는 도구 등을 의미한다.

- 컨택스트 다이어그램(CONTEXT DIAGRAM): 기능을 정의하는 최상위레벨을 의미한다. 이 다이어그램은 정의하고자 하는 기능의 전반적인 설명을 위한 것으로, 외부환경과의 인터페이스를 포함하여, 기능을 수행하는데 요구되는 전반적인 상황을 설명한다. 이 다이어그램을 정의하기 위해서는 두 가지 작업이 요구된다. 하나는 관점을 정의하는 일이고, 다른 하나는 목적을 정하는 일이다. 목적을 정의하기 위해서는 기능을 정의하는 사람들이 모여서 이 기능에서 얻고자 하는 바를 기술하여, 이를 토대로 전체의 의견을 취합하여 모델의 전반적인 의미를 정의하게 된다. 다음 단계로는 기능에 관련된 액티비티와 데이터를 수집하여 정리하는 것이다. 이때 데이터들은 구체적인 데이터이기 보다는 상위레벨의 추상적인 데이터 중심으로 수집되며, 액티비티는 이 기능이 무엇을 수행하여야 하는 지에 대한 조사이다.

- 분할의 원칙 : 하나의 상위기능은 3-6개의 하위기능으로 분할할 수 있는데, 최소한의 단위를 3개로 정의한 이유는 원하는 정보의 표현을 구체화하기 위함이다. 기능의 분할은 원하는 정보가 표현될 때까지 계속된다. 각 단계에서 정의된 기능은 하위 기능이라 정의하며, 하위기능은 상위기능에서 정의한 인터페이스(boundary arrow)를 포함하여야 한다. 이는 기능의 분할 시 발생될 수 있는 정보의 유실을 방지하기 위한 것이다. 하나의 기능이 2개로 분할될 경우라면, 분할하지 않아도 그 정보의 의미를 파악할 수 있다는 생각하에서 최소한의 분할 수를 3개로 정의하였다. 상한선을 6개로 정의한 이유는 초기의 작업이 A4에 직접 작성하는 수작업이었기 때문에 해독성을 좋게 하기 위하여 제한하였으며, G. Miller의 'Magic number 7'의 영향을 받은 면도 있다.



[그림 1] IDEF0 모델의 표기

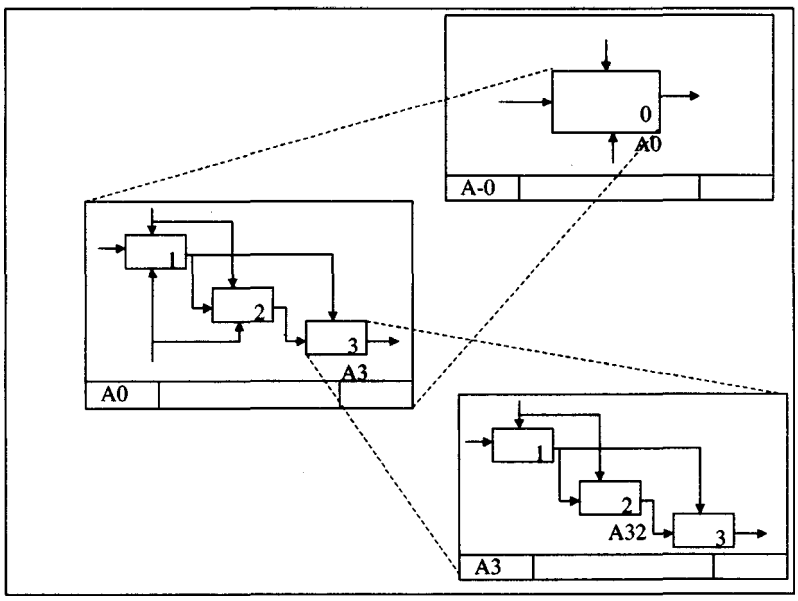
- 명명법 : 기능은 영어의 동사를, 데이터는 영어의 단수 명사를 사용하여 명명하도록 정의함으로써, 의사소통시의 오류를 예방하였으며, 하나의 기능이나 인터페이스에서 정의한 언어를 타 기능에서 다른 의미로 사용하는 것을 금지하여, 표기에서 오는 불일치성을 방지하며 무결성을 보장하였다.
- 조직과 기능의 분리: 기능을 정의하는데 있어서 조직의 관점에서 보면 여러 가지 관점(View)으로 정의될 수 있다. 따라서, 정의한 모델에 따라 다를 수 있으므로, 이를 바탕으로 모델링 하는 것은 바람직하지 않다. 조직과 기능을 분리함으로써, 기능을 하나의 통일된 관점에서 조망할 수 있다. 이를 위하여 IDEF0에서는 컨텍스트 다이어그램의 정의 시에 분석자 관점(viewpoint)를 정의하도록 하고 있다. 분석자 관점의 정의는 기능에 관계된 모든 사람들의 관점을 정리한 것이 아니라, 그 중에서 모델의 중심이 되는 관점을 정의하고, 이 관점에서 모델을 정의하고 기술하도록 함으로써, 모든 사람들이 그 관점에서 모델을 이해하도록 하여, 의사소통의 장애를 제거하는 것이다. 분할도중 분석자 관점이 바뀌게 되면, 새로운 컨텍스트 다이어그램이 정의된다.

2.3 IDEF0 구성요소

2.3.1 다이어그램 (DIAGRAMS)

IDEF0의 모델은 복잡한 주제를 하부요소로 분할하여 이를 다이어그램으로 표현한 것이다. IDEF0모델의 최상위 레벨의 다이어

그램을 컨텍스트 다이어그램이라 정의하고, 이를 분석자의 관점에 따라서 3-6개의 하위구성요소로 분할하는 기법을 사용하여 모델링하게 된다. 상위 다이어그램의 외부 화살표(boundary arrows)는 반드시 하위 다이어그램에 표현되어야 한다. 이에 대한 예외는 최상위 다이어그램인 컨텍스트 다이어그램에서 분할될 경우와 뒤에서 언급할 괄호표시 화살표(TUNNELLED ARROW)인 경우이다. 전자의 경우, 최상위레벨의 다이어그램이 추상적인 의미의 상위레벨의 데이터를 포함하고 있으므로, 다음단계로의 분할에서 반드시 일치할 필요가 없다. 후자의 경우는 정의된 인터페이스가 상위나 하위레벨에서 필요하지 않고, 그 레벨에서만 정의될 경우에 허용한다. 각 다이어그램은 레벨을 정의하는 번호를 갖는데, 컨텍스트 다이어그램은 A-N (N은 정수)으로 표현되며, 바로 다음 단계의 다이어그램에는 A0의 번호를 부여한다. [그림 2]에 나타난 바와 같이 다음 단계의 다이어그램은 A0 다이어그램 내의 대각선 위치에 따라 부여 받은 일련번호와 자신의 다이어그램의 고유번호로 사용한다.



[그림2] 다이어그램과 기능분기의 예

2.3.2 기능상자 (FUNCTIONAL BOXES)

IDEF0에서 정의된 기능상자는 기능이 수행하는 어떤 행위, 행동, 프로세스 또는 동작 등을 의미하는 것으로 이를 통칭하여 기능이라 정의한다. 기능은 무엇을 수행하여야 하는 지에 대한 정의로, 기능상자 안에 동사형의 단어로 표현한다. [그림2]에 표현한 바와 같아, 하나의 다이어그램 안에 있는 기능상자에 기능상자의 왼쪽 모서리에 1에서 6까지의 일련번호를 대각선의 위치에 따라 부여하여 이를 인식기호(box number)로 사용한다. 이 번호와 다이어그램의 번호가 결합하여 기능상자 자신을 설명하는 다이어그램의 번호가 되며, 이 번호를 노드번호(NODE NUMBER) 또는 C-NUMBER(연대번호: chronological number)라 한다. 다이어그램 내에서 기능 상자의 위치는 수행되는 순서와는 무관하지만, 일반적으로 왼쪽 모서리에 위치하는 1번 기능상자는 다이어그램 내에서 다른 기능상자에 대하여 주도적인 (dominant)역할을 한다

2.3.3 화살표(ARROWS)

다이어그램 내의 화살표는 [그림 1]에 정의한 바와 같이 두 가지로 구분된다. 한가지는 기능의 수행에 관련된 데이터(입력/제어/출력데이터: Input/Control/Output Data))를 의미하며, 다른 한가지는 기능을 수행하는 주체에 대한 것으로 메커니즘(Mechanism)이라 정의된 화살표를 의미한다. 메커니즘은 일반적으로 기능을 수행하는 사람이나 도구 또는 시스템을 의미한다. 이 화살표는 기능상자의 아래에서 위로 향하도록 정의되어 있으나, [그림 1]과 같이 반대로 기능상자의 아래에서 아래로 정의된 화살표가 존재한다. 이 화살표는 참조(CALL) 화살표라 명명된 것으로 기능을 수행하는데 참조가 되는 다이어그램을 정의한다.

다이어그램 내에서 정의된 데이터는 기능들간의 순차적인 순서를 의미하는 것이 아니라, 기능상자를 정의하는 제약조건을 의미한다. 즉 기능상자는 입력데이터를 받아서, 제어데이터의 조건에 따라 출력데이터를 생성하는 것을 의미하는 것으로, 준비된 데이터가 없으면, 수행할 수 없음을 의미한다. 따라서 화살표는 기능을 수행하는데 요구되거나 (입력/제어 데이터), 기능의 결과로서 산출되는 정보나 개체 또는 객체(출력 데이터)를 의미한다. 이 데이터는 명사형으로 명명한다. 한편 다이어그램 내의 화살표는 인터페이스 화살표(BOUNDARY ARROWS)와 내부화살표(INTERNAL ARROWS)로 나뉘어지는데, 전자의 경우는 상위레벨의 다이어그램에서 정의된 화살표이며, 후자의 경우는 그 레벨에서 분할 시에 발생하는 화살표로서 다음 단계에 인터페이스 화살표로 정의된다. 한 단계에서 정의한 화살표가 다음단계에서는 필요하지 않거나, 특정한 단계에서만 필요한 데이터의 경우에는 처음부터 이를 언급할 필요는 없다.

이 경우, 상위레벨에서 정의한 데이터는 반드시 하위레벨에 정의되어야 한다는 일관성을 유지하기 위하여 특정의 표시를 하는 경우가 있는데 이러한 화살표를 괄호표시 화살표라 한다.

IDEF0의 데이터 중에서 가장 중요한 것은 입력데이터와 제어데이터에 관한 정의이다. 제어데이터는 기능의 수행에 요구되는 환경이나 조건 등을 나타내는 데이터로서 특정 데이터가 입력과 제어 데이터로서 역할을 수행하거나, 그 역할이 모호할 경우 제어데이터로 정의하도록 권고하였다. 제어데이터로 정의된 데이터도 기능에 의해서 그 값이 변화할 수 있으므로, 기능은 적어도 하나 이상의 제어데이터를 요구하지만 입력데이터는 정의하지 않아도 된다.

한편, 다이어그램 내의 데이터 화살표는 경우에 따라서, 다른 화살표와 결합하여 상위레벨의 데이터 또는 일반화된 데이터로 표현되거나, 나뉘어져서 세분화된 데이터로 정의가 가능하다.

2.4 IDEF0의 제약사항

IDEF0는 기본적으로 기능들간의 관계에 초점을 맞추기보다는 기능이 무엇인가를 정의하는데 중점을 두었으므로, 업무영역을 분석하여 이를 바탕으로 하는 시스템의 구현에는 다음과 같은 제한이 따른다.[Kim et al. 1993]

- 프로세스의 모델링 시에 프로세스 간의 선후관계에 따른 여러 가지 사항들을 표현할 수 없다는 것이다. 즉, 대각선 상으로 기능들을 일렬로 모델링함으로써, 동시에 수행되는 기능이나, 선후관계가 있는 기능들간의 수행순서를 정확히 표현할 수 없다. 이를 위하여 IDEF3[Mayer et al. 1995]가 개발되었다. IDEF3에서 표현할 수 있는 것은 기능과 객체로, 프로세스는 UOB(Unit of Behavior)들간의 관계에 의해서 정의되며, 객체는 객체의 상태변화에 따라 정의된다. 그러나, IDEF3의 프로세스는 IOCM이 없는 일련의 행위(activity)나 기능(function) 들만을 연결한 것이다. IDEF0에서 정의하는 프로세스와 IDEF3에서 정의하는 프로세스 의 의미가 다른 점도 혼돈을 야기시키는 요인이다.
- 기능을 ICOM로 정의하는 과정에서 기능을 수행하는데 요구되는 입력데이터, 제어데이터에 대한 정보의 출처를 정의하지 않는 것이다. 이것은 단순히 하나의 업무영역에서 분석하여 모델링할 경우에는 문제가 되지 않을 수 있으나, 통합환경하에서의 시스템 구현 시에는 혼란을 야기시킨다. 즉, 종래의 단위 업무중심의 기능분석이나 시스템구현에서 벗어나, 업무영역 간의 횡적인 연결을 전제

로 하는 통합 환경하에서는 기능의 수행에 요구되는 정보의 출처를 파악하는 것은 중요하다. 이 정보의 출처는 궁극적으로 정보의 소유권 (ownership) 과 관계가 있다. 즉, 정보공학측면에서 보면, 정보는 CRUD(Create/Read /Update/Delete)라는 수명주기를 갖게 되는데, 통합 환경하에서 생성의 기능을 갖는 프로세스나 기능은 유일한 것이어야 데이터의 무결성 (data integrity)을 보장할 수 있다.

- IDEF0모델링 기법에서 모델링 결과에 대한 품질(quality)를 평가하는 기준 중의 하나가 기능의 분할 방법이다. [Marca et al. 1993] 기능을 구체적으로 파악하기 위한 기능의 분할은 일정한 원칙하에 이루어져야 한다. 모든 분석을 분석가나 설계자들이 모여서 할 수 없는 경우에, 각각의 분석이 일관성을 갖기 위해서는 분할하는 기준이 일정하여야 하지만, IDEF0는 이러한 원칙을 제공하지 못하고 있다. 따라서, IDEF0로 분석할 경우, 결과에 대한 신뢰성이 상대적으로 취약할 수 있다. 또한 , 기능의 분할이 어디까지 이루어져야 한다는 원칙이 없으므로, 분석자가 임의로 충분한 정보를 얻을 수 있을 때까지 분할하게 되어, 업무영역에 따라 분할정도의 차이와, 분할 내용의 세세함이 다를 수 있다.
- 기능의 표현 시 ICOM으로만 표현하도록 함으로써, 정성적인 분석은 가능하지만, BPR시에 요구되는 정량적인 값을 정의할 수 없다는 것이다. 단순히 무엇을 하는 기능이라고 정의하는 것은 BPR을 위한 분석에는 도움이 되지 않는다. 전체의 프로세스 상에서 어느 기능이 병목기능인지, 가장 코스트가 많이 드는 기능이 무엇인지를 찾아내어, 이들의 리드타임과 코스트를 줄이는 방법을 강구할 수 있어야 BPR을 위한 분석도구로서의 가치를 가질 수 있다
- 통합 환경하에서는 기존에 단위 업무영역에서 주어지는 정보보다 더 많은 정보가 요구되며, 이러한 정보들은 여러 가지 관점에서 통합되어 의사결정을 지원하여야 한다. 의사결정을 위하여 사용자가 새로운 정보요구사항을 제시하였을 경우, 분석가나 설계자의 입장에서 이 정보를 얻는 데 요구되는 기능들의 조합이나, 그 결과를 얻기 위한 비용과 시간 등을 산정하는 방법이 없다는 것이다.
- 기능들간의 관계에서는 각 기능의 입력데이터가 어느 시점에 도달하는 지를 알 수 없으므로, 주어진 프로세스가 수행되는데 걸리는 시간에 대한 통계치를 파악할 수 없다. 이러한 이유는 각 기능의 입력데이터가 이미 주어져 있다는 가정하에서 기능에 필요한 입력데이터 자체만을 정의하였기 때문에 발생한다. 그러나, 하나의 기능이 수행될 때, 그 기능의 수행에 요구되는 시간은 일정할 수도 가변

적 일 수도 있다. 또한 입력데이터 역시 늘 주어지기 보다는 산출기능의 수행시간에 따라, 또 정보가 전파되는 시간에 따라 기능에 도달하는 시간이 달라질 수 있다. 따라서, 기능들간의 흐름을 파악하는데 기능의 수행시간과 함께, 데이터의 도달시간도 고려되어야 한다.

3. ARIS (Architecture of Information System)

3.1 ARIS방법론의 개요

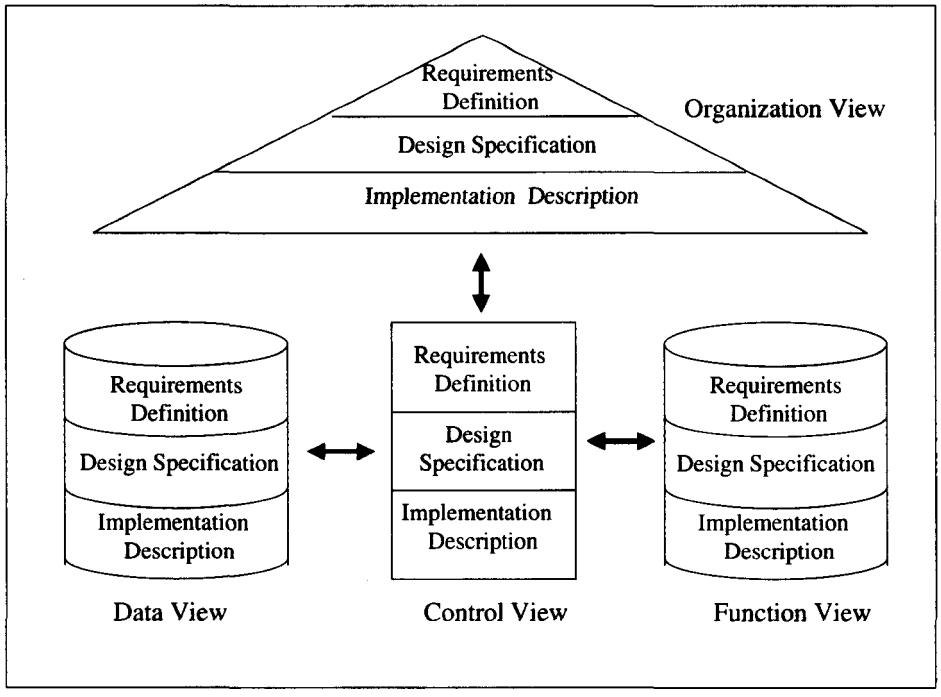
ARIS는 A.W.Scheer[Sheer 1992, Sheer 1994]가 개발한 방법론으로 현재 세계적으로 가장 많이 사용되고 있는 ERP(Enterprise Resource Planning) System인 SAP/R3의 기본 방법론으로 사용되고 있는 방법론이다. 이 방법론은 S/W개발 방법론에 입각한 방법론으로 [그림 3]과 같이 정보시스템의 구조를 데이터, 기능, 조직과 이들간의 연계(데이터제어, 기능제어, 조직제어 : DATA, FUNCTION, ORGANIZATION, CONTROL)이라는 네 가지 축을 중심으로 하여 각 각에 대하여 기술하고, 이들간의 관계를 정의하여 정보시스템의 전체 구조를 정의한 것이다.

기능 뷰 (Function View)는 조직이 수행하여야 할 업무에 대한 뷰로서, 기능 트리(FUNCTION TREE)와 이를 바탕으로 하는 프로세스로 구성되어 있다. 기능은 업무에 대한 정적인 면을 정의하는 것이고, 프로세스는 업무의 동적인 정의하는 것으로 이 두 가지에 의해서 기업의 업무가 정의된다. 조직 뷰(Organization View)는 프로젝트의 오너와 스폰서가 누구인지, 그 업무에 대한 전문가는 누구인지를 나타내는 것으로, 실제의 업무 프로세스를 표현하기 위하여 조직을 고려하여 작성하는 것이다. 이는 BPR 추진 시에 프로세스의 오너를 결정하기 위해서는 조직문제를 다루어야 하기 때문이다. 조직은 조직단위로 정의되는데, 조직단위는 기업의 목표달성을 위하여 업무를 수행할 책임이 있는 단위조직을 의미하는 것으로 동일한 특성을 갖는 조직을 하나의 타입으로 정의한다. 조직은 주어진 과제(JOB)에 따라, 사람에 따라 또는 위치에 따라 다른 조직도를 가질 수 있으므로, ARIS에서는 이 세가지 조직도를 제공한다.

데이터 뷰(Data View)는 EER(Extended ER MODEL)을 이용하여, 기업의 데이터를 정의하는 것으로, 데이터 모델의 작성시에 가이드라인을 제공하여, 작성자에 관계없는 일관성을 제공하고 있다. 즉 데이터의 특성(추상화, 일반화, 상세화)에 따라 위치를 결정하여 모델의 해독성을 높이고 있다. 이를 위해서 ARIS 방법론에서는 클러스터 데이터 모델(CLUSTER DATA MODEL)을 제공한다. 클러스터는 몇 개의 개체와 관계 타입에 대한 논리적 관점을 포함하는 것으로 ER 모델을 단순화 시켜주는 효과가 있다. 또 한가지 특징은 모

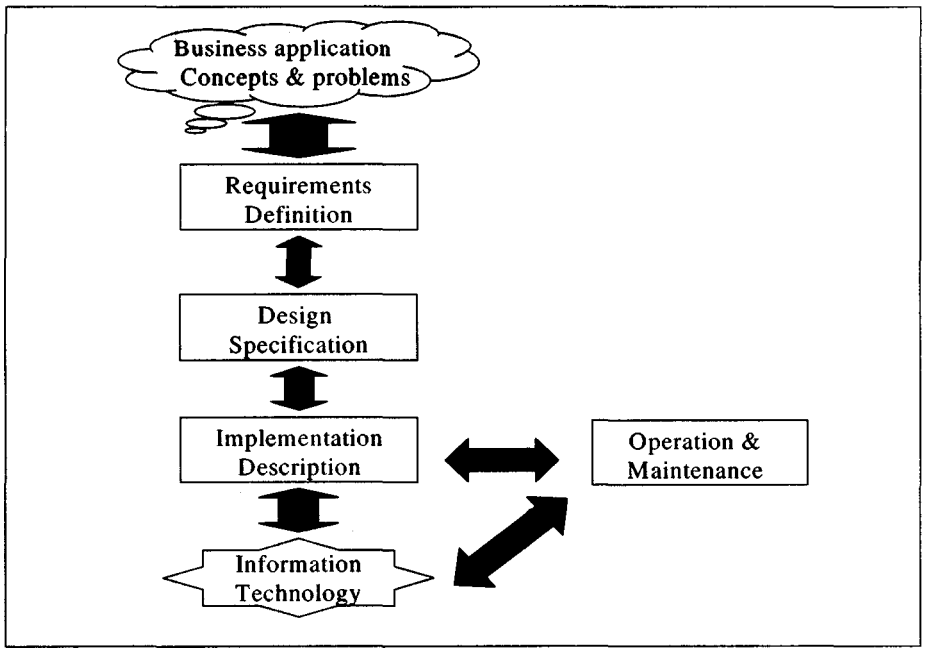
형의 복잡성을 줄이기 위하여 ER 모델에서는 속성을 표현하지 않고, 별도의 다이어그램 (eERM attribute allocation diagram)으로 처리하여 표현한다.

제어 뷰(Control View)는 데이터, 기능, 조직 뷰에 있는 오브젝트 사이의 연결관계를 표현한 것으로, 프로세스 중심으로 기능들간의 관계를 표현한다. 따라서, 제어 뷰는 서로 다른 뷰들간의 조합(데이터-기능, 데이터-조직, 기능-조직, 데이터-기능-조직)으로 구성되어 있다. 데이터-기능간의 뷰에는 기능을 수행하는데 요구되는 데이터와 기능 수행의 결과로 얻어지는 데이터가 무엇인지를 정의하게 되며, 기능들 사이에 상호 교환하는 데이터가 무엇인지, 동일한 데이터를 필요로 하는 기능이 무엇인지를 규명하게 된다. 데이터-조직간의 뷰는 어떤 조직단위에서 어떠한 데이터에 접근할 수 있는지, 어떤 조직 단위에서 어떤 데이터를 요구하는지, 누가 어떤 데이터에 대한 책임을 가지고 있는 지를 규명한다. 기능-조직 간의 뷰에는 어떤 조직이 어떤 기능을 실행할 책임이 있는 지, 기능의 수행결과가 누구에게 전달되는 지, 기능의 수행 결정의 책임은 누구에게 있는가를 규명하게 된다.

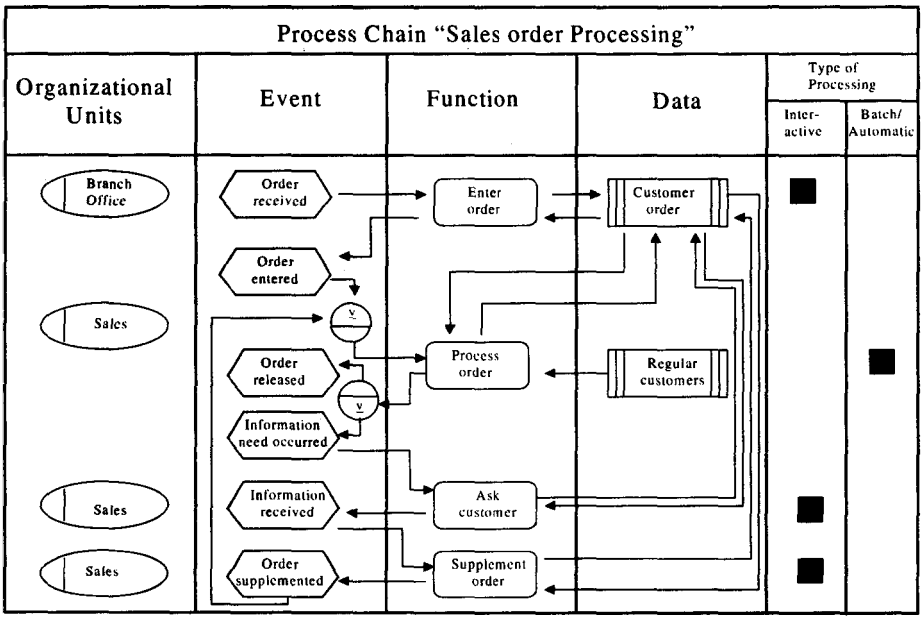


[그림3] ARIS의 구조

따라서 각 각은 별도의 독립적인 구조를 갖고 있지만, 서로 연계되어 하나의 틀(FRAMEWORK)을 구축하게 된다. [그림 4]는 이러한 단계를 도식적으로 나타낸 것이다. 이 방법론의 특징은 그림에 언급한 바와 같이, 정보기술 (INFORMATIN TECHNOLOGY)을 방법론의 단계에서 정의하고 있다는 것이다. 이는 다른 방법론이 시스템의 분석 및 설계에 중점을 둔 반면 ARIS 방법론은 구현까지도 방법론에서 정의하고 있다는 점이 차이가 있다. 이 방법론은 비즈니스에 정보기술이 영향을 미치고, 또한 정보기술이 비즈니스의 영향을 받아 발전한다는 상호 보완적인 관점에서 구현을 고려한 방법론이다. IDEF0 방법론이 기능의 정의(WHAT)에 중점을 둔 모델링 방법론인 반면에, ARIS는 기능과 다른 정보시스템 구성요소 들간의 연계방법(HOW)에 중점을 둔 방법론이다. 따라서 이 방법론의 출발점은 정보시스템을 구성하는 구성요소 들간의 연계를 규명하는 것으로, 출발점은 프로세스 체인(EPC: Event Process Chain)의 정의이다. [그림 5]는 프로세스 체인의 예를 나타낸 것이다.



[그림 4] ARIS 방법론의 절차



[그림 5] 프로세스 체인의 예

즉, ARIS방법론에서는 모델링 결과를 기능이 아닌 프로세스 체인으로 설명한다. IDEF방법론에서는 기능모델과 정보모델이 별개로 정의되지만 (CASE TOOL에 의한 연계는 별도의 작업임), ARIS방법론에서는 언급한 바와 같이 기능모델과 정보모델이 동시에 정의되어야 한다는 점이 다르다.

ARIS방법론에서 정의하는 기능은 기업의 다양한 목적을 이루기 위하여 정보 또는 객체와 관련되어 수행하는 과업(task)으로 조직이 수행하여야 할 업무를 규명하여 이를 바탕으로 정보시스템(업무시스템)의 구조를 설계하기 위한 기본을 제공하는 것이다. 따라서, 기능은 해야 할 것을 나타내는 정적인 모습이며, 프로세스는 일하는 방법을 보여주는 동적인 모습으로 시간에 따라 달라진다.

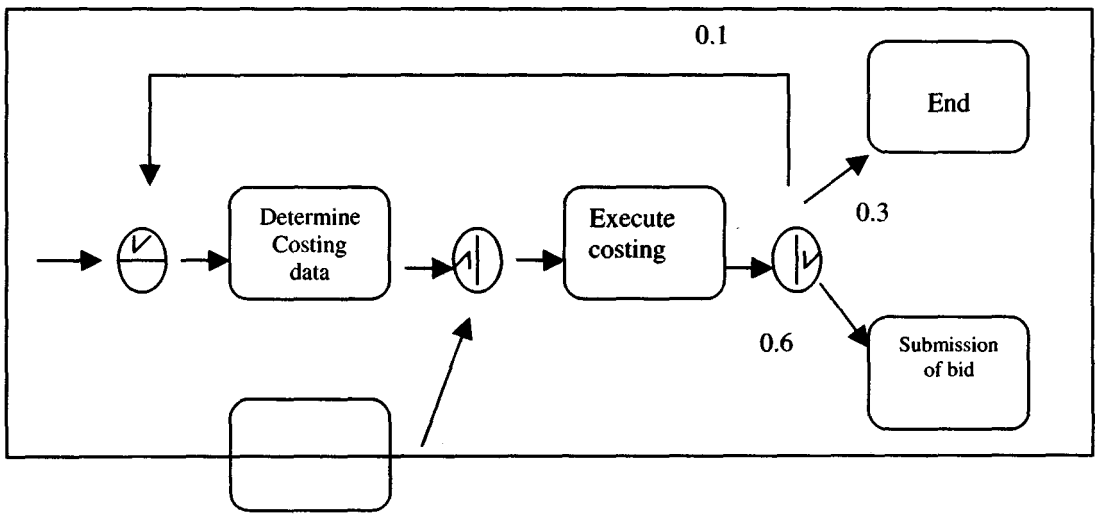
3.2 기능모델링 방법론

3.2.1 요구정의단계 : 기능

ARIS방법론 중에서 기능모델링 방법론은 [그림 4]에서 정의한 바와 같이 먼저 요구정의에서 출발한다. 요구정의를 통하여 기능뷰를 정의하게 되는데, 기능 뷰에는 기능의 구

조와 프로세스 순서와 프로세스의 처리형태 등이 기술된다. IDEF0 방법론과 마찬가지로 복잡한 기능은 그 기능을 분할함으로써, 단순화시켜 기능의 구조를 파악하게 되는데 이때 사용되는 차트가 기능 트리이다. 기능 트리는 기능의 계층구조를 나타낸 것으로, 분할된 기능은 모서리가 둥근 박스(round box)로 표현된다. 기능 트리는 지향하는 바에 따라 객체중심의 기능 트리, 실행중심의 기능 트리, 프로세스 중심의 기능 트리로 나누어 정의한다

ARIS 방법론에서는 IDEF0와 달리 기능의 분할에 대한 정의가 있는데, 기능은 기능이 수행하는 작업의 사이클에 따라 분할하며, 비즈니스 관점에서 보아 더 이상의 분할이 의미가 없다면, 분할하지 않으며, 이 때 분할된 최소한의 단위를 요소기능(elementary function)라 정의하며, 모든 기능들은 이러한 요소기능 들의 조합으로 정의될 수 있다. 기능구조도 역시 기능의 정적인 구조로 기능들 간의 순서가 명확하지 않다. 이러한 부분을 해결하기 위하여 기능들간의 위치관계(positional relationships)를 [그림 6]과 같이 화살표로 정의하게 된다.



[그림 6] 기능들간의 위치관계의 예

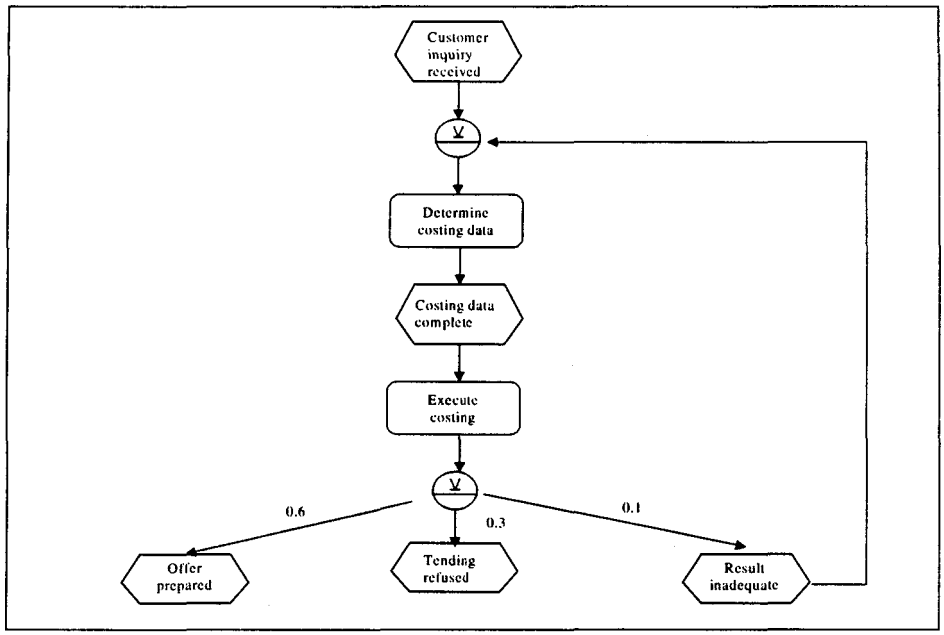
ARIS에서 정의하는 프로세스는 시간과 이벤트의 개념으로, 프로세스는 항상 시작 이벤트에 의해서 시작되며, 종료 이벤트에 의해서 종료된다. 이벤트 프로세스의 분기는 [그림 6]에 표시한 바와 같이 확률적 분기를 한다. ARIS 방법론에서는 이벤트를 업무에 관련된 정보 객체의 상태를 의미하며, 이 객체의 상태가 업무 프로세스의 흐름을 통제하거나 영향을 준다.

프로세스의 처리형태는 두 가지 경우로 나뉘어진다. 하나는 사용자의 간섭이 배제된 자동적인 처리형태로 배치처리의 형태를 의미하며, 다른 하나는 사용자와의 상호작용에 의해서 처리되는 경우로, 의사결정이 요구되는 어플리케이션의 경우이다. [그림 5]의 오른쪽에 정의된 "type of process"가 이에 대한 예를 나타낸 것이다.

3.2.2 요구정의단계 : 제어

이미 언급한 바와 같이, 기능에 대한 요구정의는 독립적으로 하나의 뷰를 형성하지만, 정보시스템을 구성하는 다른 뷰들과 연관을 갖게 된다. 이러한 연관성을 정의하는 뷰가 제어 뷰(control view)이다.

- 기능과 조직과의 연계: 기능과 조직과의 연계는 먼저 조직 차트상에 프로세스 체인 기능과 조작을 연계하는 것이다. 이러한 연계는 기능의 통합정도를 결정할 수 있으며, 기능에 대한 조직의 의사결정 책임을 정의할 수 있다.
- 기능과 데이터와의 연계 : 데이터의 속성값의 변화나 객체의 발생으로 정의되는 이벤트는 기능의 수행에 관련된 것이다. 따라서, 기능의 관점에서 보면 데이터는 이벤트의 결과인 동시에 이벤트의 입력으로서 정의된다. 따라서, 데이터와의 연계는 이벤트와의 연계로 정의될 수 있다. 따라서 ARIS 방법론에서는 데이터 흐름을 이벤트 제어에 의해서 표현한다. ARIS 방법론에서는 이벤트를 [그림 5]에 나타낸 바와 같이 마름모 형태로 표현된다. 그림에서 보는 바와 같이 하나의 이벤트에 의해서 여러 개의 기능이 수행되기도 하며, 하나의 기능에 의해서 여러 개의 이벤트가 발생하기도 한다. 이를 위해서는 이벤트들간의 논리적 조합이 요구되는데 ARIS방법론에서 제공하는 논리자는 AND, OR, XOR이다. [그림 7]과 같이 이벤트와 기능들간의 조합에 의해서 정의되는 다이어그램을 EPC(event-driven process chain)이라 한다.



[그림7] EPC의 예

3.2.3 설계 정의 단계 (DESIGN SPECIFICATION) : 기능부분

이 단계에서는 모듈과 거래처리(TRANSACTION)에 관한 설계를 하는 단계로 구현될 시스템의 제어구조와 입출력 화면을 설계한다. 전 단계에서 정의된 기능구조는 STRUCTOGRAM으로 표현된다. 이 다이어그램은 프로세스의 상세한 레벨에서의 관계를 설명할 수 있을 뿐 아니라, 업무의 의사결정의 규칙도 잘 표현할 수 있다는 점에서 사용된다. 모듈은 그 내부에 계층구조를 가지며, 이에 따라 태스크를 처리하는 단계마다 요구되는 처리방법(자동/배치/사람의 입력)을 정의하게 된다. ARIS 방법론에서 사용되는 모듈이라는 용어는 일반적인 의미로, 거래처리, 프로그램, 처리단계, 프로그램 요소 등을 총칭한다. 요구정의단계에서 정의된 프로세스 체인은 내부에 여러 개의 트랜잭션을 포함하고 있으므로, 각 단계마다 요구되는 화면의 형태를 정의하여야 한다.

3. ARIS 방법론의 제약사항

ARIS 방법론은 언급한 바와 같이 SAP/R3의 방법론으로 이미 그 성과를 파악할 수 있다. 그러나, 이 방법론을 일반 분야에 적용시키기에는 다음과 같은 단점이 있다.

- 방법론의 복잡성 : ARIS 방법론은 전체를 일관되게 정의할 수 는 있지만, IDEF0에 비하여 상대적으로 복잡하여, 모델링 작업이 전문가가 아니면 구현하기 쉽지 않다. 방법론의 복잡성은 자연적으로 분석가와 사용자 또는 현업과의 의사소통의 문제를 야기할 소지가 있다. 이러한 문제는 전 세계적으로 널리 사용되고 있는 SAP/R3의 전문가를 고용하는데 다른 방법론 전문가 보다 많은 비용을 지불하여야 하고, 교육에 많은 시간을 투자하여야 하는 문제를 갖고 있다.
- 모델의 연계성 : IDEF방법론은 모든 모델을 독립적으로 정의할 수 있으므로, 필요한 부분만 정의함으로써, 그 결과를 알 수 있지만, ARIS방법론은 각 각의 뷰들이 서로 연계되어 있어 이들을 종합적으로 정의하지 않으면, 파악하기 어렵다. 모델의 연계는 IDEF0보다 용이하지만, 많은 사람들이 쉽게 사용할 수 있도록 하기위한 IDEF0가 추구해온 단순성과는 배치된다.
- 기능정의 :ARIS 방법론에서의 기능은 IDEF0 모델과 달리, 기능 자체보다는 기능들 또는 데이터와 연계를 바탕으로 정의하게 되므로, 기능 자체를 정의하는데 독립적이지 못하다. 이는 모델간의 독립성보다는 모델들간의 연계를 통하여 전체를 파악하고자 하는 의도에서 출발하였기 때문이다. 따라서, 모델을 통한 의사소통에는 IDEF0에 비해 상대적으로 불리한 점이 있따.
- 모델의 정량화 : IDEF0모델과 마찬가지로, ARIS 의 모델 역시 그 결과를 정량화하는 문제에 있어 명백하지 않다.

4. 제안 방법론

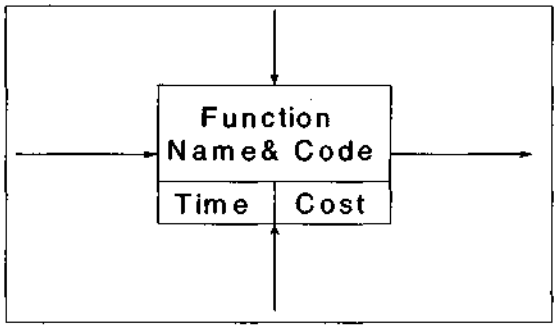
본 연구에서는 기능 모델링 방법론 중에서 현재 미 FIPS의 표준으로 정의되고, 향후 CALS 표준으로 정착될 가능성이 있는 IDEF0을 바탕으로 이 방법론의 단점을 보완하고, ARIS 방법론의 장점을 일부 채택한 새로운 방법론을 제안하고자 한다. IDEF0를 기본 방법론으로 정의한 이유는 다음과 같다.

- 미 연방정부의 표준(FIPS) 및 CALS 및 BPR 분야에서의 수용
- 모델링의 단순함으로 인하여, 습득의 용이함과 실무자와의 의사소통의 원활
- 모델링의 엄격한 규칙성을 인한 모델의 공유 및 해독의 효율성

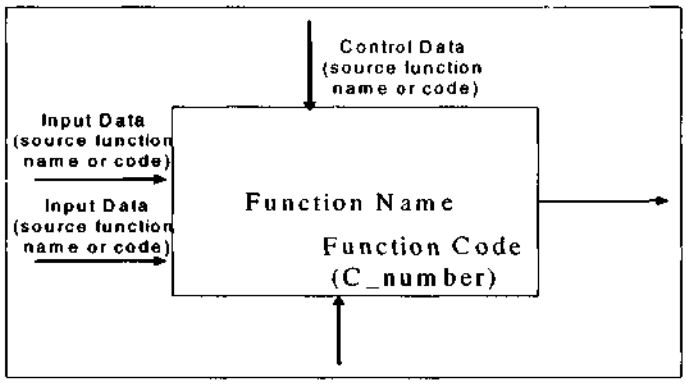
- 기존 사용자들의 사용시의 정확성 및 완전성 [Yadav et al 1988]

IDEF0를 바탕으로 본 연구에서 제안한 방법론은 다음과 같은 기능적 특성을 갖는다.

- 시간과 비용항목 추가: 각 기능은 그 기능을 수행하는데 요구되는 시간과 비용이 있는데 이를 [그림 8]과 같이 표현하는 것이다. 또한, 입력데이터에도, 이 데이터가 도달하는 시간분포가 요구된다. 이러한 시간에 대한 정의는 주어진 프로세스에 대한 시뮬레이션에 요구되는 정보이다. 이러한 항목의 추가는 앞에서 언급하 바와 같이 BPR사에 프로세스 상에서 병목이 되는 기능과 비용이 많이 드는 기능을 찾아내는데 도움을 주게 된다. 즉, 기능에 대한 모델링 결과를 정량화 하는데 요구되는 정보인 시간과 비용을 정의함으로써, IDEF0가 가지고 있는 제약 사항인 정량화 문제를 해결할 수 있다. '시간' 요소의 추가는 후에 이 모델을 이용하여 시뮬레이션 시에 필요한 요소중의 하나이다.
- 정보의 산출기능에 대한 표기 : 그림9와 같은 정보의 근원에 대한 표기는 기업 통합 시, 이 정보의 근거가 기업내부인지 기업 외부인지를 파악하는 기준이 되며, 어느 기능이 이 정보에 대한 소유권을 가지고 있는 지를 파악할 수 있게 한다. 이는 통합환경이나 기업 통합 시에 전제조건으로 작용하는 정보의 무결성을 보장하는 한 방법이 된다. 즉, 하나의 기능이 특정정보를 산출하게 되면, 그 기능은 그 정보에 대한 생성/삭제 권한을 갖는 것이며, 입력데이터로 사용하게 되면, 수정/갱신의 권한을 갖게 되는 것이며, 제어데이터로 사용하게 되면, 입력의 권한을 갖게 되는 것이다. 정보에 대한 권한이 표기됨으로써, 기능을 수행하는 사용자는 정보에 대한 접근권한을 명확히 알 수 있으며, 정보관리자는 이를 관리함으로써, 시스템의 무결성을 보장할 수 있게 된다. 이는 다음 단계에 개발될 정보모델에 중요한 영향을 미치게 된다. 데이터의 소유권(Ownership)을 정의하여, 데이터의 무결성을 보장하는 데이터 모델의 설계는 정보 모델링 방법론의 가장 중요한 원칙중의 하나이다.



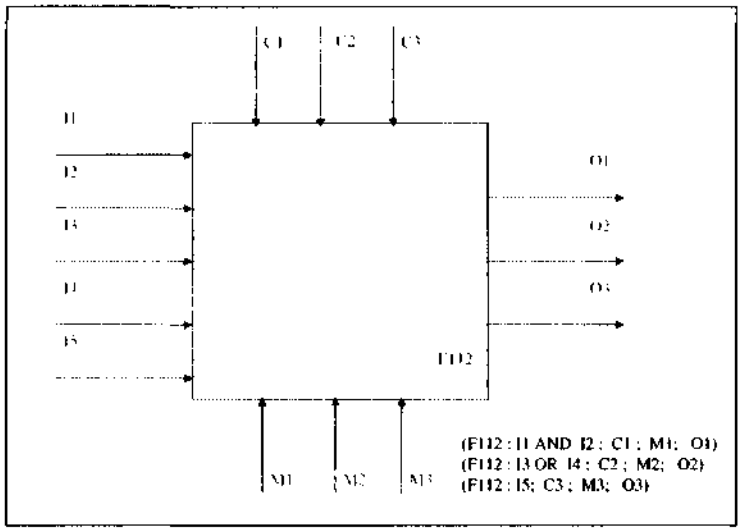
[그림 8] 시간과 비용의 표기



[그림 9] 입력/제어데이터에 대한 근원 표기

- 입력데이터의 조합에 의한 기능표현: 기능을 수행하는데 요구되는 입력데이터는 몇 가지 조합에 의해서, 기능을 수행하게 된다. 이들 조합은 기능의 분할에도 영향을 미치게 된다. 즉, 하위기능으로의 분할은 기본적으로 분할된 기능을 수행하는데 요구되는 데이터의 조합에 의해서 수행되기 때문이다. 따라서, [그림10]과 같이 이들 데이터의 조합을 그림에 표현하는 것은 기능의 수행에 중요한 역할을 하게 된다. 입력 데이터의 조합을 연산자 'AND', 'OR', 그리고 'XOR'로 정의하여, 각 기능의 기동조건 (Firing condition)을 정의하게 한다. 'AND'는 입력데이터가 모두 들어와야 수행되는 경우이며, 'OR'는 입력데이터의 조합 중 하나만 들어와도 기능이 기동되는 경우를, 'XOR'는 어느 데이터가 입력되면, 다른

데이터는 입력되어서는 안되는 경우를 정의한 것이다. 이 관계는 데이터의 배타적 관계를 정의할 수 있으므로, 기능들간의 순서에도 영향을 미칠 수 있다. [그림 10]의 (F112: I1 AND I2 : C1: O1)이라는 표현은 출력 O1이 생성되기 위해서는 I1, I2라는 입력데이터가 동시에 요구되며 C1이라는 제어데이터가 필요하다는 의미이다. 이렇게 연산자를 이용하여 입력데이터의 조합을 정의하면, 데이터간의 관계를 명확히 할 수 있을 뿐 아니라, 기능의 기동조건도 명확해지는 장점이 있다. 하나의 기능이 수행되기 위해서 요구되는 모든 데이터가 입력되었을 때, 비로소 기능은 그 역할을 제대로 할 수 있기 때문이다. 이 작업은 추후에 데이터 모델링 시에 데이터들간의 관계를 정의하는 근간이 된다.



[그림 10] 입력과 출력데이터의 조합표기

- 요구정보에 대한 추적기능: 사용자가 원하는 정보를 제시하였을 경우, 이 정보를 생성하는 기능을 찾을 수 있으며, 이 기능의 입력 또는 제어데이터로 작용하는 데이터를 찾을 수 있으므로, 또 다시 이들 데이터를 산출하는 기능을 찾을 수 있다. 이러한 방법을 반복하면, [그림 11]과 같이 원하는 정보에 관련된 모든 기능들을 추적할 수 있으며, 이 결과로 이 정보를 얻는데 요구되는 시간과 비용을 파악

할 수 있으며, 각 기능의 메커니즘 데이터를 이용하여, 이를 정보를 산출하는 데 요구되는 요소들을 파악할 수 있다. 이러한 추적기능은 다음에 언급할 기능의 블록화에 의한 단위기능 정의와 입력정보의 근원에 대한 정의표기에 의해서 가능하다. 즉, 모든 기능은 단 하나의 출력만을 갖는 단위기능으로 정의되므로, 이들 기능의 입력이 되는 기능들을 정의된 근원에 따라 추적하여 원하는 정보에 관련된 기능들의 조합을 정의할 수 있다.

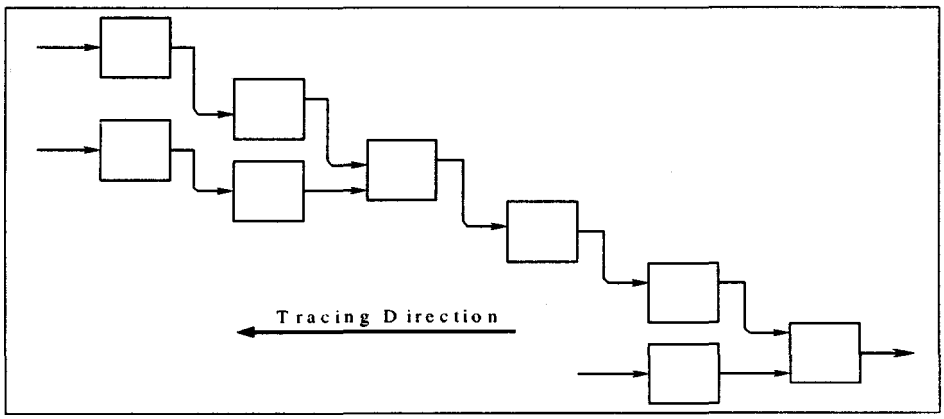
- 입력과 제어의 정의: IDEF0 모델에서는 입력과 제어데이터의 역할이 모호할 경우, 이를 제어데이터로 정의하였으나, 이 경우, 모호성이 증대하게 되므로, 본 연구에서는 입력과 제어를 다음과 같이 정의한다.
  1. 입력데이터 : 기능에 의해서 그 값이 변화되는 데이터로서, 그 데이터가 입력되는 경우에 한해서 기능이 기동될 때(Firing Condition), 이 데이터를 입력 데이터라 정의
  2. 제어데이터 : 가능에 의해서 그 값이 변화하지 않는 데이터로서, 이미 주어진 데이터를 의미한다. 따라서 이 데이터는 기능의 수행 시에 이미 존재하므로, 이 데이터에 의해서 기능의 기동여부가 결정되지 않는다.

- 분할중지의 방법정의 (Stopping Rule) : 기능의 분할은 그 기능이 하나의 출력을 산출할 때까지 분할을 계속한다. 최종적으로 분할된 기능을 단위 기능(unit function)이라 정의하면, 단위기능은 레고 블럭처럼 활용할 수 있다. 모든 기능을 분할하여 레고 블럭처럼 활용할 수 있다면, 새로운 프로세스의 구현이나 이를 바탕으로 하는 시스템 구현 시에 이들 블록들의 조합으로 구현할 수 있으므로, 기능의 모듈화가 가능해진다. 본 연구에서 제안한 분할의 방법은 다음과 같다.

  1. 계획-실행-통제('Plan-Do-See')의 개념에 따라 분할: 기능은 수명주기를 갖게 되므로, 이에 따라 기능을 분할한다.
  2. '정보수집-정보처리-정보분배/제공(Information Gathering - Information Processing -Information Stewardship')의 기능을 구분하여 분할: 정보를 처리하는 기능은 정보를 수집하고, 이를 처리하고 처리된 정보를 공급하는 기능을 갖게 되므로, 이에 따라 분할한다
  3. 메카니즘에 따라 분할: 하부기능이 각기 다른 메커니즘을 요구한다면, 이에 따라 분할한다. 이 경우는 크게 두 가지로 나뉘어지는데, 첫째는 메커니즘의 공유가 불가능한 자원(resource)라면, 하부기능은 메커니즘에 따라서 당연히 분할되어야 한다. 둘째는, 메커니즘이 기능을

수행하는 사람인 경우, 사람의 역할에 따라서 기능이 분할된다.

1. 출력데이터에 따라 분할: 여러 개의 출력데이터가 존재할 경우, 이 데이터가 입력데이터의 조합에 의해서 이루어진다면, 이 조합에 따라 기능을 분할한다
2. 알고 있는 결과대로 분할 : 제어시스템의 경우처럼, 분기되는 경우를 미리 명백히 알고 있으면, 이 경우에 따라서 분할한다.
3. 분할된 기능이 출력을 생성하지 않으면, 사용자가 원하는 정보를 생성하는 기능이 아니므로, 분할의 의미가 없다.



[그림 11] 입력데이터에 의한 기능의 역 추적

5. 결론

IDEF0는 그 방법의 단순성과 이해하기 쉬운 장점 때문에 널리 사용되어 왔다. 그러나, CALS가 추구하는 기업간 또는 기업내부의 통합을 전제로하는 업무의 분석에는 표현요소의 부족으로 많은 제한이 따른다. 본 연구는 이러한 점에서 IDEF0가 가지고 있는 고유의 장점을 바탕으로 새로운 환경에 맞도록 확장한 것이다.

현재 이 방법론은 위에서 언급한 내용이외에 방법론 진행에 관한 절차와 지침(Procedure and Guideline)을 개발 중에 있으며, 방법론을 지원하는 CASE TOOL을 현재 개발하고 있다. 또한, 이 모델을 바탕으로 기존의 SALM II와 같은 시뮬레이션 언어로의 변환을 통하여, 모델의 정의 시, 이를 바로 시뮬레이션하여 결과를 판단하거나, 모델의 대안을 선정하게 하는 연구도 진행 중에 있다.

이 연구는 정보모델링, 동적모델링, 프로세스 모델링 등의 일련의 작업을 위한 첫 단계로 진행되는 것으로, CALS 구현을 위한 방법론은 이 모든 것을 포함하여야 한다. 이러한 관제들은 이 작업 후에 연차적으로 이루어질 계획이다.

References:

[Bravoco et al, 1989] R.R.Bravoco and S.B.yadav, " A Methodology to Model the Functional Structure of an Organization", Computers in Industry, No.6, 1989 (345-361)

[FIPS 183] Draft Federal Information processing Standards Publication 183,

[FIPS 184] Draft Federal Information processing Standards Publication 184,

[Kim et al, 1993] C. Kim, K. Kim, and I. Choi, " An Object-Oriented Information Modeling Methodology for manufacturing Information Systems", Computers and Industrial Engineering, Vol.24, No.3, 1993 (337-353)

[Marca et al, 1993] D.A.Marca and C.L.McGowan, IDEF0/SADT Business Process and Enterprise Modeling, , Eclectic Solutions Corporation, 1988

[Mayer et al, 1995] R.J. Mayer, C.P.Menzel, et al, Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, KBSI, 1995

[Ross et al, 1977] D.T. Ross and K.E.Schoman, " Structured Analysis for Requirements Definition", IEEE Transactions on Spoftware Engineering, Vol. SE-3, No.1, Jan. 1977, (6-15)

[Sheer, 1994] A.W.Scheer, Business Process Engineering, Springer-Verlag, 1994

[Sheer, 1992] A.W.Sheer, Architecture of Integrated Information Systems, Springer-Veralg, 1992

[U.S. Air Force, 1981] U.S. Air Force, "ICAM Architecture Part II, Vol. VI - Dynamics Modeling manual(IDEF2), Wright-Patterson, 1981

[Wisdom, 1985] Information Modeling manual IDEF 1x, Wisdom Systems, Inc., 1985

[Yadav et al 1988] Yadav, S.B. Bravoco, R.R., Chatfield, A.T., and Rajkumar, T.M. "Comparison of analysis techniques for information requirements determination, Communications of ACM, Vol. 31, No.9, 1988 (1090-1097)

저자소개

김철한

현재 대전대학교 산업공학과 전임강사로 재직중이다. 한양대학교 정밀기계공학과에서 학사, 석사를 취득하고, 대우중공업 중앙연구소, LG Software의 공장자동화실에 근무, 포항공과대학교 산업공학과에서 박사학위를 취득하고, 삼성 SDS/SM 전략실에서 삼성그룹의 정보인프라를 기획

관심분야는 정보전략, CIM과 ERP를 포함하는 생산정보시스템(Data/Process Modeling), 개발방법론, CALS(IDB) 및 EI(Engineering Database and Process Design)등이다.

우훈식

한양대학교 산업공학과 (학사)

Iowa State University, Dept. of Industrial Manufacturing and Systems Engineering (석사 및 박사)

현재 시스템공학연구소 시스템통합연구부 CALS연구실 선임연구원

관심 분야: CALS IDB, CITIS, IDEF, EI (Enterprise Integration), SCM (Supply Chain Management)등

현재, 정보통신부 "CALS/EC 기술 및 모델 개발"사업의 "CITIS/CALS 통합 DB 기술 개발"과제 중 CALS 통합 DB 기술 개발 과제를 담당하고 있음.

김 중인

1982-1987: 한양대학교 산업공학과 (학사)

1987-1989: 한양대학교 산업공학과 (석사)

1992-1995: Arizona State Univ. 산업공학과 (박사)

1995-1996: 한국전자통신연구원(ETRI) 선임연구원

1996-현재: 홍익대학교 경영정보학과 전임강사

관심분야: CALS/EC, 시스템개발방법론, ERP, Database

임 동 순

현재 한남대학교 산업공학과 부교수로 재직 중이다. 한양대학교 산업공학과에서 학사, 한국과학기술원 산업공학과에서 석사, 아이오와주립대학 산업공학과에서 박사학위를 수여하였다. 주요 관심 분야는 생산시스템 및 프로세스의 설계, 분석을 위한 시뮬레이션 도구 개발이다.