

# Document Trace Diagram과 IDEF 모델을 이용한 CALS 시스템 개발 방법론

김 성 희\*, 조 성 식\*, 이 재 광\*, 한 창 희\*, 윤 영 석\*

## CALS System Development Methodology Using Document Trace Diagram and IDEF Model

Kim, Soung Hie, Cho, Sung Sik, Lee, Jae Kwang, Han, Chang Hee, Yoon, Young Suk

The basic goal of CALS is to improve transactions and relationships among organizations through information sharing and integration. CALS is an information strategy which needs strong cooperation between organizations or between users and developers in design step. However, current design methodologies using IDEF models, that are considered to be standard for CALS system development, has some limitations. For example, it is difficult for system developers to communicate with counterparts by IDEF model since IDEF models are difficult for counterparts to understand.

In this paper, we suggest a development methodology for CALS systems by complementing IDEF model with Document Trace Diagram, which we developed as a communication tool. The concept of Document Trace Diagram stems from the fact that most information exchanged within or between organizations is in the form of documents and most standard operating procedures of organizations are about processing the documents. It helps system developers identify functions and their ICOMs (Input, Control, Output, Mechanism) with ease and little communication cost. With this methodology, we have constructed the CALS prototype system for construction industry.

---

\* 한국과학기술원 테크노경영대학원

## I. 서 론

최근 CALS는 기업 경영의 효율성을 극대화하기 위한 정보 전략으로 인식되고 있다. CALS는 비즈니스 프로세스 리엔지니어링 (Business Process Reengineering), 동시공학 (Concurrent Engineering) 등을 바탕으로 업무 기능 및 업무 간 연계를 효율적으로 체계화시키고 업무 프로세스 과정에서 생성, 저장, 분배되는 문서 및 자료를 디지털화 하여 네트워크 시스템을 통한 상호 공유 및 교환이 되도록 자동화되고 통합된 정보교환시스템을 구축하고자 하는 것이다 [서효원 & 김진영, 1995; 김성희, 1995]. 기업의 불필요한 업무 및 프로세스들간의 비효율적인 연계, 비체계적인 문서 및 데이터베이스 등의 문제를 CALS를 도입함으로써 해결할 수 있다. [김성희, 김문호, 조운호, 한창희 & 장기진, 1997].

CALS를 성공적으로 도입하기 위해서는 관련 업무 프로세스의 개선 및 정리, 효과적인 정보 공유를 위한 통합 데이터베이스와 네트워크의 설계, 그리고 기업간의 데이터 표준화 등의 문제를 해결하여야 한다. 그러나 기업들에서 CALS 도입을 추진할 때 체계적인 방법론이 전무한 상태이다. 따라서 효율적이고 체계적인 방법론이 요구되고 있다 [김효석 & 김창수, 1997].

CALS 시스템 구축 방법론으로서 업무프로세스와 데이터베이스와의 연계를 위한 IDEF0 모델과 IDEF1X 모델의 연계 방법이 김성희 등에 의해 제시되었다 [김성희, 이재광, 한창희 & 조운호, 1997]. 이 방법론은 CALS 도입 목적의 설정, 핵심프로세스의 선정, 도입시나리오의 작성, IDEF 모델링, 시스템 구현의 순서로 CALS 구축의 일반적 방법론을 정리, 제안하고 있으며, 이에 덧붙여 IDEF 모델링 단계에서는 IDEF0 모델에서 IDEF1X 모델을 도출함으로써 IDEF1X 모델링에 소요되는 시간과 노력을 줄일 수 있음을 제안하고 있다. 이 방법론은 CALS 구축에

필요한 IDEF 모델링 단계의 문제점을 줄이기 위한 노력으로 IDEF 모델링을 최소화하고 있다.

각 기능간의 정보 흐름을 제시하는 IDEF0 모델과 각 업무 활동에서 파악된 정보를 통합하여 데이터베이스화 하는 IDEF1X의 작성은 CALS의 구축에 있어 매우 중요한 과정이라 할 수 있으나 실제 기업의 활동을 모델링 하는데 있어서는 많은 문제점이 제기되고 있다 [Colquhoun, Baines, & Crossley, 1993; Busby & Williams, 1993; Shunk, Sullivan, & Cahill, 1986]. 특히 IDEF 모델은 아직 현업의 실무자들이 쉽게 이해하지 못하여 분석가와의 의사소통 수단으로 부적합하며 하향식 접근법을 사용하고 있음으로 해서 오류가 발생하면 상위 모델을 모두 수정해야 하므로 많은 시간이 소요된다.

본 연구에서는 기존의 CALS 시스템 구축 방법론이 갖고 있는 IDEF 모델링의 문제점을 해결하기 위한 도구로서 Document Trace Diagram (DTD)을 개발하고 이를 김성희 등의 방법론의 IDEF 모델링 단계 앞에 추가하여 기존 방법론을 보완하였다. DTD는 조직 내의 문서교환을 업무 흐름에 따라 도식화한 것으로 업무 분석 모델링 단계에서 분석가의 IDEF 모델링 지식과 현업 실무자의 실무 지식을 연결하는 의사소통 도구 역할을 하며 일정한 규칙에 따라 IDEF0 모델로 변환되므로 기존 방법론의 IDEF 모델의 장점을 유지하면서 의사소통의 문제점을 줄일 수 있는 도구이다. 그리고, 제시된 구축 절차 및 방법론을 다양하고 방대한 양의 문서 교환이 필수적인 건설 산업에 적용하여 CALS 프로토타입을 설계한 예를 제시한다.

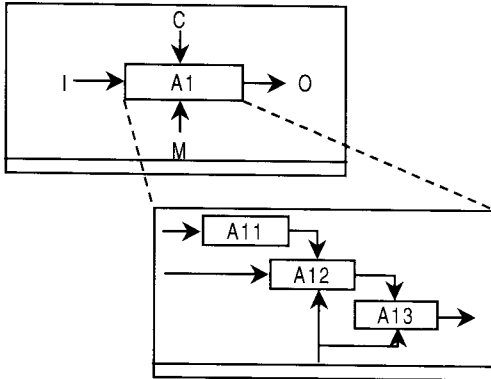
## II. CALS시스템 구축 방법

### 2.1 IDEF 모델

IDEF (Integration Definition) 모델은 조직과 시스템이 복잡하게 연관을 가진 환경을 분석하

기에 매우 유용한 기법으로 체계적이고 구조적인 절차를 제공한다. IDEF모델의 활동 단위의 분석은 불필요한 복잡성을 제거하고 시스템과 조직을 수준에 따라 표현하므로 기업의 통합과 재설계를 지원하기 위한 분석에 적합하다. 또한 IDEF 모델은 기능 모델에 제한되지 않고 정보, 프로세스 등을 분석할 수 있는 모델을 제공한다[Mayer, 1994].

IDEF0는 시스템과 환경에 기능 (function)을 표현하고 각 기능간의 관계와 연관된 정보와 객체를 표현하는 모델이다[Mayer, 1994; Sarkis & Lin, 1994]. 각 활동은 <그림 1>과 같이 ICOM 즉, 입력 (Input), 출력 (Output), 제어 (Control), 그리고 메카니즘 (Mechanism)으로 표현되며 상위 수준의 추상화된 활동을 상세하게 세분화하여 하위 수준에서 모델링한다.

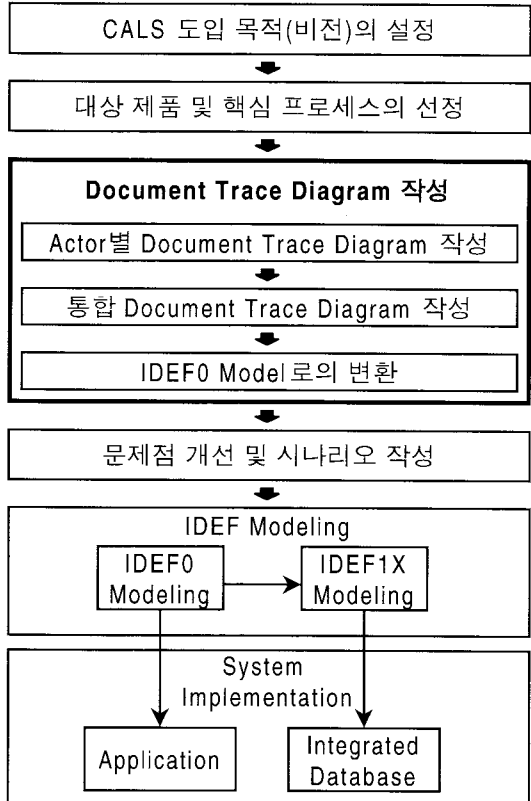


<그림 1> IDEF0 모델

IDEF1은 시스템과 환경의 기능을 지원하기 위한 정보의 구조를 생성하고 표현하는 모델이다[Mayer, 1991]. IDEF1X는 관계형 데이터베이스를 구축하기 위해 개념적 데이터 모델을 작성하며 이 모델을 기반으로 논리적 데이터 모델을 작성할 수 있다[Mayer, 1991]. 이 모델은 시스템과 조직의 목적을 수행하기 위해 필요한 정보를 표현하는데 사용된다

## 2.2 CALS 시스템 구축 절차

<그림 2>는 김성희 등의 방법론에 DTD를 추가한 CALS시스템 구축 절차를 설명하고 있다.



<그림 2> DTD와 IDEF를 이용한 CALS 시스템 구축 절차

제일 먼저 CALS 도입 목적을 설정한다. 기업이 추구하는 목적과 비용절감, 제품 납기의 단축, 품질 및 신뢰성의 향상, paperless한 사업 환경의 구축 등의 CALS 목적이 잘 부합될 수 있는 도입 목적을 설정 한다.

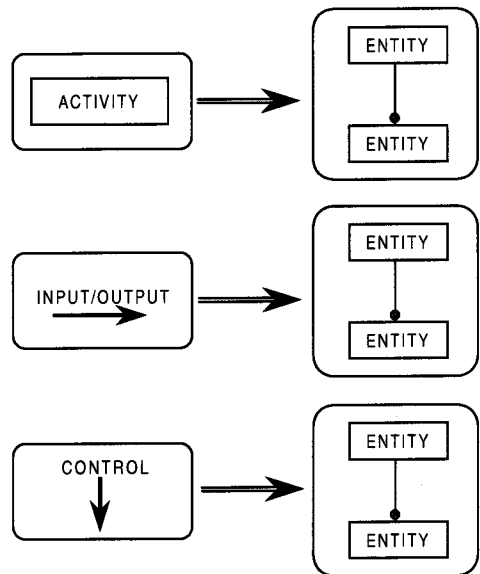
CALS 시스템을 효과적으로 구현하기 위한 대상 제품(군)을 선정된 후, 선정된 대상 제품(군)에 대한 개별적인 프로세스의 역할을 이해하기 위하여 프로세스의 분류를 수행한다. 기업

의 주요 프로세스를 제품이나 서비스에 가치를 부가하는 운용 프로세스, 업무를 수행하기 위한 사람 및 장비의 의사결정이나 평가에 관련된 관리 프로세스, 그리고 운용 프로세스의 흐름을 후방에서 지원하는 지원 프로세스로 분류하여 개별적인 프로세스의 역할을 분석한다. 대상 제품(군)에 관련된 모든 프로세스에 CALS를 도입하는 것이 CALS의 효과를 극대화하지만 CALS 도입의 타당성을 미리 파악하기 위한 파일럿 테스트가 목적인 경우 대상 제품 관련 프로세스 중 핵심 프로세스를 선정하여 CALS를 도입한다. CALS 도입시의 기업 프로세스의 구조 분석과 프로세스의 분류를 위해 다양한 업무를 수행한 경력이 있거나 기능간 업무의 조정을 수행하는 내부 구성원들과의 면담을 통하여 필요한 자료를 수집한다.

CALS 도입 목적(비전)이 설정되고 핵심 프로세스가 선정되면 시스템 분석과 설계를 위해 IDEF 모델링을 시작해야 하는데 이 단계 전에 DTD를 사용하여 IDEF 모델링에 소요되는 시간과 오류를 줄인다. 우선 업무를 수행하는 주체인 actor별로 DTD를 작성하게 하고 이를 취합하여 통합 DTD를 완성한다. 각 actor들은 자신을 중심으로 들어오는 문서와 작성하여 내보내는 문서를 DTD에 표현한다. 이를 통합하여 핵심 프로세스 전체를 문서들의 흐름으로 표현하면 이 프로세스에 대한 전체적 분석이 가능해진다. 이 핵심 프로세스에 관여하는 actor들이 누구인지 그들이 처리하는 문서는 어떤 것들이 있는지, 어떤 절차를 거치는지, 처리하는 과정에 필요한 정보는 어떤 것들이 있는지가 파악된다. 이 과정에서 actor들이 직접 자신을 중심으로 DTD를 작성하게 되므로 실무자에게서 분석가로 정보가 전달되는 과정에서의 정보 왜곡 가능성을 막을 수 있으며 동시에 많은 actor들이 각자 DTD를 작성할 수 있으므로 분석 시간을 단축할 수 있고 상향식 방식이라 상위 모델의 수정 필요가 없어진다. 그러나 통합 DTD만

으로는 CALS 시스템을 구축할 수 없다. IDEF 모델의 활용을 위해서는 DTD를 IDEF0 모델로 변환해야 한다. 통합 DTD를 변환시켜 얻은 IDEF0를 바탕으로 문제점을 발견하고 이를 개선하여 새로운 업무 프로세스를 시나리오의 형태로 제시한다.

하나의 IDEF 모델을 이용하여 시스템의 기능, 정보, 그리고 동적 관점을 모두 표현할 수는 없다. 따라서 기능 모델인 IDEF0와 데이터 모델인 IDEF1X의 연계가 필요하다[Colquhoun, Baines & Crossley, 1993]. 업무 프로세스 분석을 통하여 IDEF0모델을 완성한 후 각 활동과 입력, 출력, 그리고 제어(Control)에 해당하는 데이터를 분석하여 IDEF1X 모델을 각 부분별로 그린다<그림 3>. 이렇게 하여 만들어진 부분 데이터 모델을 같은 개체들을 통합하여 하나의 데이터 모델로 만든다. 그리고 최종적으로 데이터 모델에서 관계를 표현하여 통합 데이터 모델을 완성한다. 이렇게 완성된 IDEF1X모델을 바탕으로 통합 데이터베이스를 구축하게 된다 [김성희, 이재광, 한창희, 조운호, 1997].



<그림 3> IDEF0모델로부터 데이터 모델의 도출

각각의 모델을 별도로 작성하지 않고 기능 모델을 기반으로 데이터 모델을 작성함으로써 시스템의 통합 관점 (View)를 제공하며 이렇게 하여 개발된 시스템은 업무와 데이터와의 유기적 관계를 가지는 시스템이 될 수 있다. 또한 모델링에 소요되는 시간과 노력을 절감할 수 있다.

### Ⅲ. Document Trace Diagram을 이용한 IDEF0 모델링

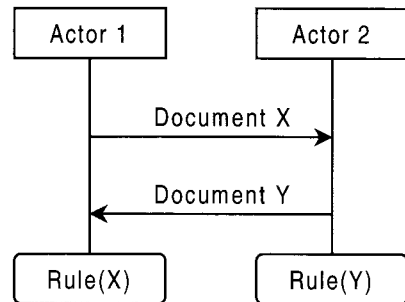
#### 3.1 Document Trace Diagram

DTD란 각 조직과 조직 사이에서 교환되는 문서들의 흐름을 업무 처리 주체인 actor를 중심으로 도식화한 것이다. 현재 조직간의 공식적인 정보 교환은 대부분 문서의 교환을 통해 이루어지며 CALS 시스템이 갖추어야 할 핵심 요인이 조직간의 정보의 원활한 공유 및 흐름을 지원하는 것이므로 CALS 시스템의 분석 및 설계의 출발점은 각 조직의 업무 프로세스별로 교환되는 문서의 흐름을 추적하는 것이다.

DTD의 장점은 실무자들이 IDEF 모델보다 이해하기 쉬운 형태의 업무 프로세스 모델링 도구라는 것이다. 실무자는 자신의 업무를 자신이 처리하는 문서의 순서대로 표현하면 된다. 어디서 어떤 문서가 전달되어 오고 어떤 규정이나 절차에 따라 그 문서를 처리하며 새로 작성한 문서는 누구에게 전달하는지를 기술하면 된다. 실제로 실무자와의 인터뷰를 통한 업무 분석 시 IDEF와 같은 도구들은 의사소통에 상당한 애로점을 갖고 있다. IDEF를 처음 접하는 실무자들은 이를 이해하는데 많은 시간을 소요하며 시스템 개발자는 인터뷰 내용을 IDEF 모델로 정리하여 실무자에게 확인 받는 과정에서 실무자의 이해 부족으로 인해 정확한 모델링이 이루어지지 않는 경우가 발생한다.

DTD를 사용하게 되면 actor들은 다른 actor들의 업무 처리 과정은 알 필요 없이 오직 자신의 업무와 다른 actor들과 주고 받는 문서만 알고 있으면 되므로 업무 프로세스 전체를 알고 있지 않아도 DTD를 작성할 수 있다. 이러한 장점은 업무 프로세스 분석 시 분업화 및 병렬화를 가능하게 하여 분석에 소요되는 시간을 줄일 수 있다. 특히 분석가가 실무자들과 일일이 면담할 필요 없이 실무자들이 작성한 DTD를 취합함으로써 업무 프로세스 분석이 가능하므로 분석 시간을 상당히 줄일 수 있으며 실무자들이 직접 DTD를 작성하므로 개발자와 실무자의 오해의 소지를 줄일 수 있다.

<그림 4>는 간단한 형태의 문서의 흐름을 보여 준다. DTD는 집합 Document, Actor, Rule, Sequence로 표현이 가능하다.



<그림 4> Document Trace Diagram

##### 3.1.1 Document

Document는 actor와 actor사이에 교환되는 문서로 정의된다. 이름이 같으면 동일한 document로 본다. 따라서 하나의 document가 수정, 결재 등의 처리를 거치게 되면 다른 document로 변환되는 것이며 이 처리 전후의 두 document는 다른 이름을 가져야 한다. Document의 형태는 전자적 형태, 물리적 형태 등 기업 내에서 주고 받을 수 있는 모든 형태를 포함한다. 예를 들면, 입찰서, 청사진, 시방서, 설계 도면 CAD File,

전자 결재 문서 등의 모든 형태가 document가 될 수 있다. Document는 <그림 4>에서 보듯이 그 이름과 방향을 나타내는 화살표로 표현되며 이는 다음과 같이 기술된다.

*Document Name (Source Actor→Destination Actor)<sup>1)</sup>*

*Document Name (Source Actor→Destination Actor1, Destination Actor2, ...)*

Destination Actor가 여러 개인 경우는 동시에 여러 actor로 document를 전달하는 경우로 어느 actor에게 먼저 주는가에 대한 구분은 없다. 여기에 Destination Actor가 명확하게 명시되지 않은 Document는 다음과 같이 Source만으로 기술한다.

*Document Name (Source)*

이러한 document는 특정 destination actor를 위해서 만들어진 것이 아니라 불특정 actor들이 참조할 수 있도록 만들어 놓은 document를 표현하기 위해 만들어진 것으로 document의 출처만 밝히고 있는 형태며 주로 Rule에서 사용되는 document이다. 건설업법시행령(건설교통부), 업무지침서(관리부) 등과 같은 document들이 대표적인 예라 할 수 있겠다.

### 3.1.2 Actor

Actor는 기업에서 업무를 수행하는 개념적 단위로 정의되며 직사각형으로 표현된다. 예를 들면, 경리부, 설계1부, 주관사업처 등이 있다. Actor는 계층구조를 가질 수 있다. 설계1부 actor는 그 하부에 조사설계팀, 실시설계팀, 상하수설계팀 등의 하위 actor를 가질 수 있다. Actor의 명확한 구분을 위해 하위 actor와 그 상위 actor를 함께 표현하고자 할 경우 다음과 같이 '/' 기호를 사용한다.

*Super-Actor / Sub-Actor*

하나의 DTD에 표현되는 actor들은 서로 배타적인 관계를 가져야 한다. 즉, 하나의 DTD에 상위 actor와 하위 actor가 동시에 표현될 수 없다.

### 3.1.3 Rule

Rule은 actor가 document를 입력 받아서 이를 처리하여 새로운 document를 생성하는데 참조하는 규칙으로 정의되며 등근 사각형 안에 다음과 같이 기술된다.

Rule (X) : Ref. Y, Z

이는 document X를 생성하기 위해 document Y, Z를 참조하라는 의미다. Document Y, Z에 해당하는 것을 예로 들면 건설업법 시행령(건설교통부)와 같은 관련 법규, 계약서작성지침서(관리부)와 같은 사내 업무 절차 등이 있다.

### 3.1.4 Sequence

Sequence는 actor마다 발생하는 document 교환의 시간적 선후 관계를 표현하는 수직선으로 상위가 하위에 대해 선행한다. 그러나 한 document와 다음 document까지의 sequence line의 간격이 시간의 절대적 또는 상대적 양을 나타내지는 않는다. Document가 source actor에서 destination actor로 전달되는데 소요되는 시간은 무시하며 document는 항상 DTD에서 수평으로 교환된다. Actor들의 sequence line들의 같은 수평 위치는 같은 시각을 나타낸다. 즉, document가 시간적으로 거슬러 전달될 수는 없다.

하나의 DTD내에서 상위에서 하위로 sequence line을 따라 document의 교환 순서를 time tag를 붙여 나타낸다. <그림 4>의 예를 들면 다음과 같다.

t<sub>0</sub>:X(Actor1→Actor2)

t<sub>1</sub>:Y(Actor2→Actor1)

1) 이하 이탤릭체로 표현된 것은 모델링 시 직접 입력할 내용임.

### 3.2 Actor별 Document Trace Diagram의 작성

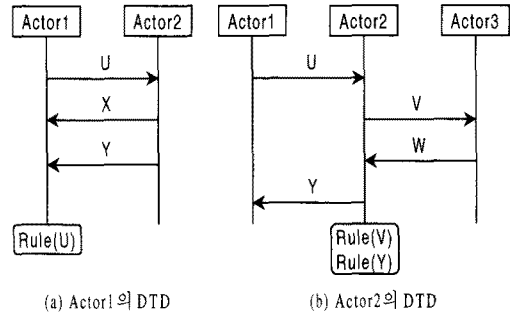
시스템 분석자는 전체 업무 프로세스를 개략적으로 파악하고 CALS 시스템 도입 대상 프로세스에 대하여 actor list를 파악한다. 그리고 actor list에 포함된 각 actor들에게 DTD를 작성하게 한다. Actor들은 자신을 중심으로 주고 받는 document와 그에 관련된 rule을 작성한다. 이 때 시스템 분석자는 자신이 파악한 actor list를 actor들에게 명시하여 이 actor list에 포함된 actor들만으로 DTD를 작성하게 한다. Actor들이 DTD를 작성하는 과정에서 더욱 세분화된 actor를 요구할 경우 시스템 분석자는 이를 검토하여 actor를 더욱 세분화하여 actor list에 포함시킨다. 이는 각 actor들이 DTD를 작성함에 있어서 DTD내에 중복된 actor가 발생하지 않도록 방지하기 위함이다.

### 3.3 통합 Document Trace Diagram의 작성

#### 3.3.1 Actor별 Document Trace Diagram의 오류 점검

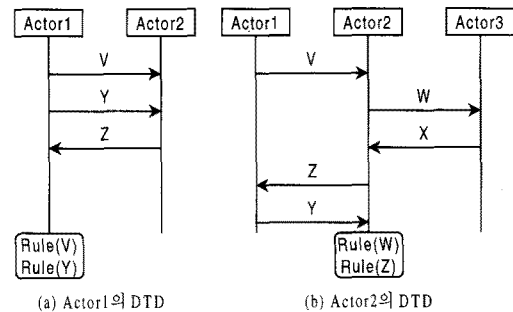
분석자는 actor별로 작성된 DTD를 통합하여 하나의 DTD를 완성하여야 하는데 먼저 actor별 DTD 사이에 발생할 수 있는 오류들을 점검한다. 점검해야 할 오류 중 대부분은 document의 불일치와 sequence의 불일치이다. Document의 불일치는 Actor1의 DTD에 나타나 있는 document가 Actor2의 DTD에는 나타나 있지 않을 때를 말하며 sequence의 불일치는 같은 document에 대한 교환 순서가 일치하지 않을 때를 말한다.

<그림 5>를 보면 Actor1의 DTD에 있는 X(Actor2→Actor1)가 Actor2의 DTD에는 없는 것을 볼 수 있다. 이 때 document의 정의에 따라 방향이 다른 경우도 불일치로 처리한다.



<그림 5> Document의 불일치

<그림 6>을 보면 Actor1의 DTD의 sequence는  $t_1:Y(\text{Actor1} \rightarrow \text{Actor2})$ ,  $t_2:Z(\text{Actor2} \rightarrow \text{Actor1})$ 인데 Actor2의 DTD의 sequence는  $t_3:Z(\text{Actor2} \rightarrow \text{Actor1})$ ,  $t_4:Y(\text{Actor1} \rightarrow \text{Actor2})$ 이므로 sequence가 일치하지 않는 것을 알 수 있다.



<그림 6> Sequence의 불일치

이러한 document의 불일치와 sequence의 불일치를 발견하면 두 actor들과의 협의를 거쳐 해결한다. Document의 불일치의 경우 document의 이름이 일치 하는지, 발생 빈도가 낮아 생략된 것인지 확인하여야 한다. Sequence의 불일치의 경우 큰 문제가 발생하지 않는다면 하나의 sequence로 통일한다.

#### 3.3.2 DTD들의 통합

Actor별로 작성된 DTD가 k개 있다고 가정하고  $A^i, D^i, R^i, S^i$ 를 각각 actor i의 DTD에

나타나는 actor, document, rule, sequence의 집합이라고 하면 DTD의 통합은 다음 알고리즘을 따른다. 통합 DTD의 document, actor, rule은 actor별 DTD들이 갖고 있는 document, rule, actor들을 통합한 것이 될 것이며 sequence는 actor별 DTD들의 sequence를 공통의 document를 기준으로 일관성 있게 document 교환의 전후 순서를 정리한 것이 될 것이다. 통합 알고리즘의 초기화 부분이 actor별 DTD의 document, rule, actor를 각각 통합하며 두개의 루프가 sequence를 통합한다. Sequence 통합의 기본 개념은 actor별 DTD에 공통으로 존재하는 document가 있으면 이를 기준으로 두 DTD의 document 교환 선후 관계를 결정하는 것이다. Set\_SEQ는 처음에는 비어 있으나 actor별 DTD를 하나씩 처리해 나가면서 통합 DTD의 sequence를 갖게 된다. 만약 공통의 document가 존재하지 않는다면 두 DTD는 특별한 선후 관계가 없으므로 병렬로 처리되거나 처리 순서가 중요하지 않다는 것을 알 수 있다.

$$\text{Set\_ACT} = \bigcup_{i=1}^k A^i$$

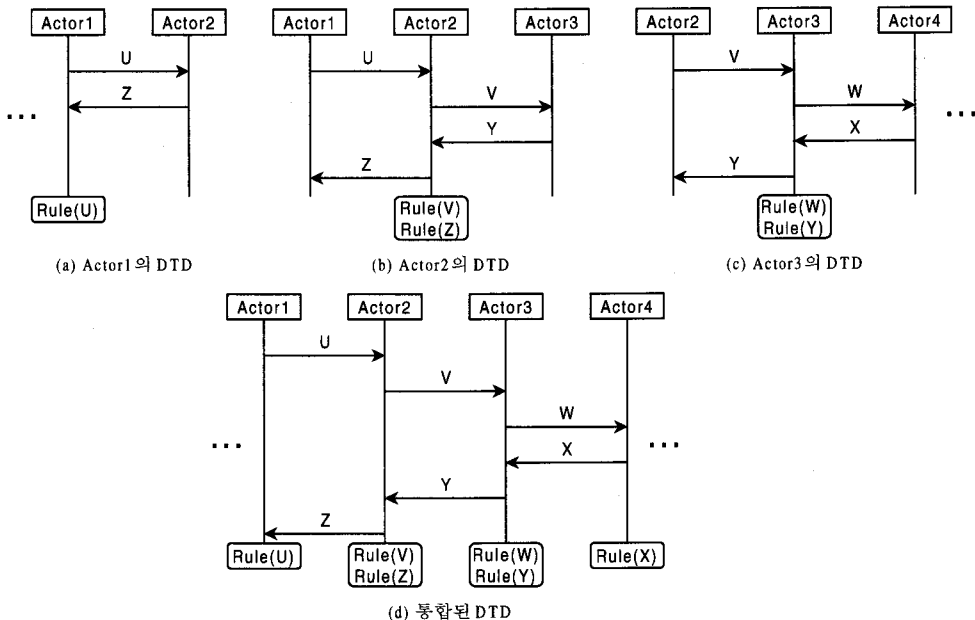
$$\text{Set\_DOC} = \bigcup_{i=1}^k D^i$$

$$\text{Set\_RULE} = \bigcup_{i=1}^k R^i$$

$$\text{Set\_SEQ} = \text{Null}$$

Loop(1)  $i = 1$  to  $k$   
 Loop(2)  $j = 1$  to Number of documents in  $S^i$   
 If  $j$ th document,  $X$ , in  $S^i$  also exists in Set\_SEQ  
 Then move all documents in  $S^i$  that precede  $X$  before  $X$  in Set\_SEQ  
 End of Loop(2)  
 Move all the rest documents in  $S^i$  at the end of Set\_SEQ  
 End of Loop(1)

이 알고리즘을 거쳐 Set\_ACT, Set\_DOC, Set\_RULE, Set\_SEQ를 구하여 이것을 그림으로 옮기면 하나의 통합된 DTD를 얻을 수 있다. <그림 7>은 통합 DTD 작성의 과정을 보여주고 있다.

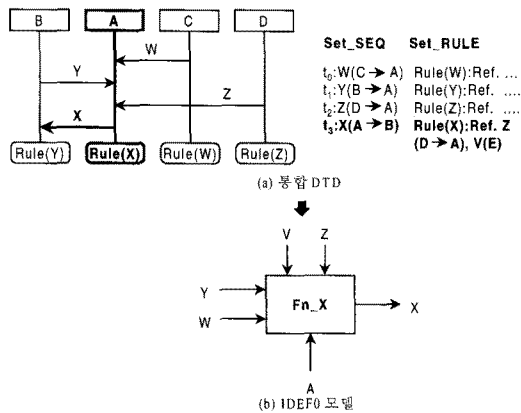


<그림 7> Actor별 Document Trace Diagram들의 통합



### 3.4 IDEF0 모델로의 변환

DTD를 이용하여 업무 프로세스의 흐름을 파악한 다음 실제 CALS 시스템을 개발하기 위해서는 이를 IDEF0모델로 변환하여야 한다. 통합 DTD가 완성되면 각 actor를 중심으로 입력되는 문서는 IDEF0의 Input으로, 출력되는 문서는 Output으로, actor는 Mechanism으로, 하나의 문서를 입력으로 받아 다음 문서를 출력으로 내보내기까지의 actor의 행위를 하나의 Function으로 정의하며, Rule은 Control로 변환된다. 이를 도식화하면 다음과 같다. Function의 이름은 임의로 정의할 수 있으며 주로 실무자와의 상의를 통해 적당한 이름을 붙인다.

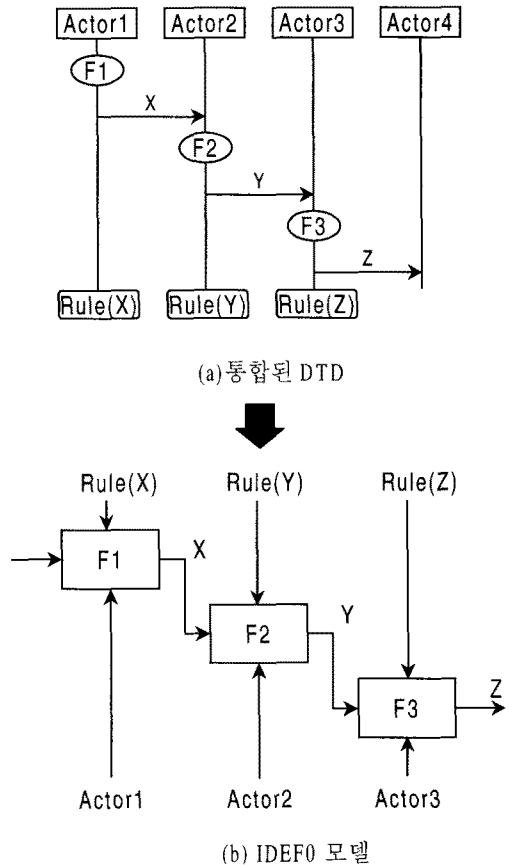


<그림 8> DTD로부터 Function 찾기

통합 DTD의 sequence 집합 Set\_SEQ에 들어 있는 document의 순서로 변환이 이루어진다. Set\_SEQ에  $X(A \rightarrow B)$ 가 있다면  $X(A \rightarrow B)$ 를 만들기 위한 actor A의 function을  $Fn\_X$ 라고 정의한다. 이  $Fn\_X$ 의 Mechanism은 actor A가 되며, Control은 통합 DTD의 Rule 집합 Set\_RULE에서 Rule(X)를 찾아 이 Rule(X)가 참조하고 있는 document가 되며, Output은 X가 된다. Sequence 집합에서  $X(A \rightarrow B)$  이전에 임의의  $T(A \rightarrow \text{any actor})$ 가 있다면 이  $T(A \rightarrow \text{any actor})$

와  $X(A \rightarrow B)$ 사이의 모든 document  $Z(\text{any actor} \rightarrow A)$ 가  $Fn\_X$ 의 Input이 된다. 즉,  $Fn\_X$  바로 이전의 function 이후에 입력되는 모든 문서를  $Fn\_X$ 의 input으로 본다. 단, Rule(X)가 참조하는 document와 Input document가 중복된다면 이 document는 Control로 본다.

Set\_SEQ에 나타나 있는 document들의 순서를 따라 function과 ICOM을 찾아 각 function들을 정의하고 function들의 ICOM을 연결시키게 되면 하나의 IDEF0모델이 완성된다. 통합 DTD에서 IDEF0 모델을 이끌어 내는 과정은 결국 다음 그림과 같이 간단하게 볼 수 있다.



<그림 9> 통합 DTD에서 IDEF0모델로의 변환

IDEF0모델이 완성되면 IDEF0 모델로부터 IDEFIX

모델을 도출하여 각각 응용 프로그램 및 통합 데이터베이스를 개발한다. IDEF0 모델에서 IDEF1X 모델을 도출하는 방법과 각각에서 응용 프로그램과 통합 데이터베이스를 개발하는 방법은 위의 '2.2 CALS 시스템 구축 절차'에서 설명하였다.

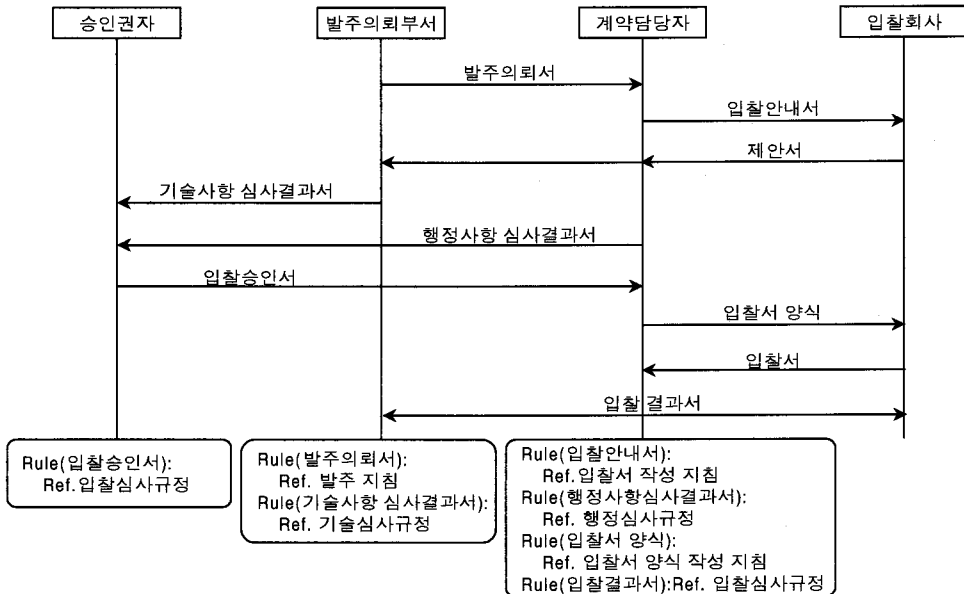
## IV. 건설업의 CALS 프로토타입의 설계

### 4.1 입찰 프로세스의 통합 DTD 작성

건설업계의 K-공사에서 가장 빈번하게 발생하는 업무 중 하나인 입찰 프로세스를 DTD로 표현하였다. Actor별로 작성된 DTD를 입찰 프로세스 관련 내용으로 통합하여 <그림 10>의 통합 DTD를 얻었다. 업무지침서를 통해 입찰 관련 actor들을 선정하고 actor를 대표하는 실무자에게 입찰 프로세스에 관한 DTD를 작성하게 하고 이를 통합한 것이다.

공사에서는 시공, 설계, 감리 등의 직접 처리

하지 않는 업무에 대해서 외부 기업에게 하청을 준다. 하청 발주를 원하는 부서는 발주 내용을 기술한 발주의뢰서를 계약담당자에게 제출하고 계약담당자는 이를 검토하여 입찰에 응할 것을 알리는 입찰 안내서를 외부 관련 업체들에게 전달한다. 입찰에 응하고자 하는 업체들은 제안서를 공사의 계약담당자에게 제출하고 계약담당자는 사본을 발주의뢰부서에 전달하여 기술 관련 사항을 검토하게 하고 행정 관련 사항을 검토한다. 입찰 승인권자는 각 입찰회사별 기술사항 검토결과와 행정사항 검토결과를 토대로 입찰 기업들 중 입찰 자격을 만족하는 기업들을 선정하고 이를 명시한 입찰승인서를 계약담당자에게 전달한다. 계약담당자는 입찰승인을 받은 기업들에 한해서 입찰서 양식을 전달하고 입찰서 양식을 받은 기업들은 입찰서를 작성하여 계약담당자에게 제출함으로써 입찰이 이루어진다. 계약담당자는 입찰 심사 기준을 참고하여 하나의 기업을 선택하여 그 결과를 발주의뢰부서와 낙찰된 회사에게 통보한다.



<그림 10> 입찰 프로세스의 통합 DTD

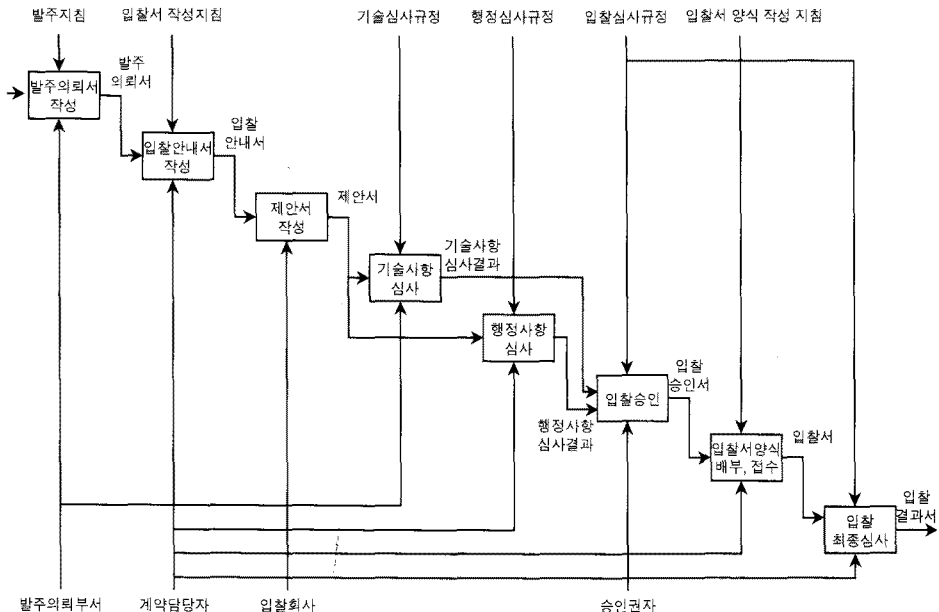
## 4.2 IDEF0 모델로의 변환

위의 DTD를 IDEF0로 변환하여 <그림 11>과 같은 IDEF0 모델을 도출하였다. 통합 DTD의 actor들이 mechanism으로, rule들이 control로, 전달되는 document들이 input과 output으로 변환되었다. 제안서 작성 function의 경우 외부 업체이기 때문에 어떤 rule을 거쳐 제안서가 작성되는지 알 수 없으므로 control이 없다.

시스템 분석 및 설계자는 이렇게 도출된 IDEF0 모델에서 부족한 부분이 있다고 판단되면 추후 actor와의 면담을 통해 보충하고 한 화면에 function의 수가 너무 많으면 유사한 function들을 묶어 상위 수준의 function을 만들어 보기 편하게 한다. 실제로 IDEF0 모델링 도구인 KBSI (Knowledge Based Systems, Inc.)의 AIO 프로그램은 한 화면에 function의 수를 6개 이하로 할 것을 권장하고 있으므로 4~5개의 function을 묶어 상위 function을 작성한다.

## V. 결 론

본 연구에서는 IDEF 모델링을 통한 CALS 시스템 개발 방법론인 김성희 등의 방법론을 확장하고 IDEF 모델링의 단점을 보완하였다. 기존 방법론의 IDEF 모델링은 여러 가지 장점에도 불구하고 실무자들이 쉽게 이해하지 못한다는 단점과 이로 인해 업무 프로세스 분석 단계에서 많은 시간을 소비하며 정확한 모델링이 어렵다는 문제점이 있었다. 그러나 본 논문에서 제시하는 방법은 Document Trace Diagram을 이용하여 실무자들이 쉽게 이해할 수 있도록 하였으며 그에 따라 분석 단계에서 실무자와 분석가의 의사소통을 원활하게 하였다. 실무자가 동시에 직접 업무 활동을 표현할 수 있으므로 분석 단계에서의 시간과 비용을 줄일 수 있도록 하였다. 또한 본 논문에서는 Document Trace Diagram에서 IDEF0 모델을 도출하는 과정을 제시하여 실무자와 시스템 분석 및 설계자를 연결시키는 방법을 제시하였다.



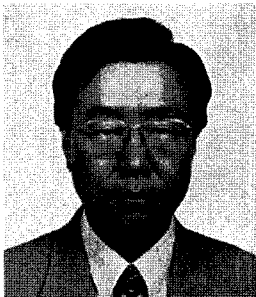
<그림 11> DTD로부터 도출된 IDEF0 모델

## 〈참 고 문 헌〉

- [1] 김성희, 김문호, 조윤희, 한창희, 장기진, “전 자산업의 CALS 파일럿 시스템 구축,” *정보처리학회지*, 제4권, 제 1호, 1997, pp. 114-130.
- [2] 김성희, 이재광, 한창희, 조윤희, “IDEF0모델과 IDEF1X모델의 연계를 통한 제조업 CALS 시스템 구축 방법 및 프로토타입 구현” *산업공학*, 제10권, 제3호, 1997, pp. 95-107.
- [3] 김성희, “CALS의 개념과 배경,” *컴퓨터월드*, 11월, 1995, p. 196.
- [4] 김효석, 김창수, “CALS 구현 프레임워크의 개발 및 적용,” *경영정보학연구*, 제7권, 2호, 1997, pp. 1-33.
- [5] 서효원, 김진영, “CALS체계에서의 CAD/CAM 및 제조하부구조,” *CALS Korea 95 Proceedings*, 1995, pp. 204-215.
- [6] Busby, J. S. and G. M. Williams, “The Value and Limitations of Using Process Models to Describe the Manufacturing Organization,” *International Journal of Production Research*, Vol. 31, No. 9, 1993, pp. 2179-2194.
- [7] Colquhoun, G. J., R. W. Baines, and Roger Crossley, “A State of the Art Review of IDEF0,” *International Journal of Computer Integrated Manufacturing*, Vol. 6, No. 4, 1993, pp. 252-264.
- [8] Mayer, R. J., *An Overview of the IDEF1 Method*, Knowledge Based Systems, Inc., College Station, Texas, 1991.
- [9] Mayer, R. J., *IDEF0 Functional Modeling*, Knowledge Based Systems, Inc., College Station, Texas, 1994.
- [10] Sarkis, J. and L. Lin, “An IDEF0 Functional Planning Model for the Strategic Implementation of CIM Systems,” *International Journal of Computer Integrated Manufacturing*, Vol. 7, No. 2, 1994, pp. 100-115.
- [11] Shunk, D., B. Sullivan, and J. Cahill, “Making the Most of IDEF Modeling-The Triple-Diagonal Concept,” *CIM Review*, Vol. 3, No. 1, 1986, pp. 12-17.

◆ 이 논문은 1998년 4월 27일 접수하여 1차 수정을 거쳐 1998년 9월 22일 게재확정되었습니다.

## ◆ 저자소개 ◆



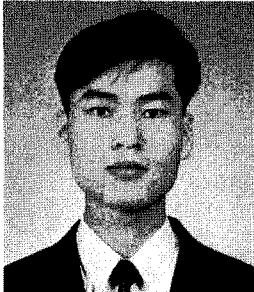
김성희(Kim, Soung Hie)

현재 KAIST 테크노경영대학원 교수로 재직 중이다. 서울대학교 공과대학을 졸업하고 University of Missouri-Columbia에서 석사학위를, Stanford University에서 경영과학 박사학위를 취득하였다. 미국 Strategic Decision Group에서 컨설팅 연구원, University of Michigan 객원교수를 역임하였으며, 1983년부터 KAIST 산업공학과, 경영정책학과 교수를 역임하였다. 주요 관심분야는 경영혁신/BPR, CALS/EC, GDSS, DA 등이다.



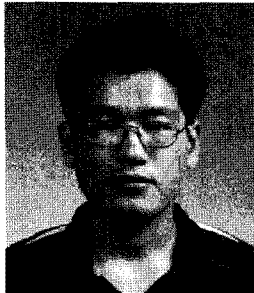
조성식(Cho, Sung Sik)

현재 한국과학기술원 테크노경영대학원 박사과정 재학 중이다. 고려대학교 전산과학과를 졸업(학사)하고, 한국과학기술원 경영정보공학과에서 석사학위를 취득하였다. 관심분야는 CALS/EC, 객체지향 모델링, GDSS, IDEF Modeling 등이다.



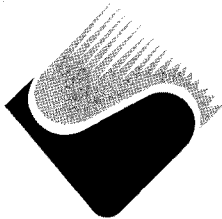
이재광(Lee, Jae Kwang)

현재 한국과학기술원 테크노경영대학원 박사과정 재학 중이다. 한국과학기술원 산업공학과를 졸업(학사)하고, 한국과학기술원 경영정보공학과에서 석사학위를 취득하였다. 관심분야는 CALS/EC, System Modeling, SCM, DSS, 인공지능 등이다.



한창희(Han, Chang Hee)

현재 한국과학기술원 테크노경영대학원 박사과정 재학 중이다. 한양대학교 산업공학과를 졸업(학사)하고, 한국과학기술원 산업공학과에서 석사학위를 취득하였다. 관심분야는 DSS/ES, CALS/EC, Strategic Decision Making, System Modeling and Simulation 등이다.



윤영석(Yoon, Young Suk)

현재 한국과학기술원 테크노경영대학원 박사과정 재학 중이다. University of Utah를 졸업(학사)하고 George Washington University에서 석사학위를 취득하였다. 관심분야는 CALS/EC, SCM, BPR, IDEF Modeling 등이다.