

## 4. IDEF3 프로세스 모델링 방법론(Process Description Capture Method)

### 4.1 IDEF3 개요

#### 4.1.1 프로세스 모델이란 ?

제임스 스와츠(James B. Swartz)는 “.....가치전달 시스템에서의 여러가지 일하는 방법들 .....”이라고 정의하고 있는데 이는 다시 말하자면 다음과 같은 것이다

- ① 프로세스의 표현 그리고 시간의 관점에 따라 표현된 요소들
- ② 프로세스 안에서 존재하는 의사결정 논리

상황(Situation)이나 프로세스(Process)를 설명하는 가장 일반적인 커뮤니케이션 메커니즘 가운데 하나는 사건이나 활동을 시간의 연속적 순서로 표현하는 이야기(Story)이다. IDEF3는 이러한 형태의 설명적 활동(Story)을 표현하기 위하여 특별히 개발된 시나리오 지향적 프로세스 흐름의 모델화 방법(Process-driven process flow modeling method)으로서 주어진 환경에서 어떤 상황이나 사건의 원인과 결과에 대한 전문가의 설명을 바로 포착, 표현할 수 있도록 구축되었다. IDEF3은 미 공군의 주도하에 Knowledge Based Systems Inc.가 수행한 ICE(Information Integration for Concurrent Engineering) Program에서 개발되었으며 1992년 5월 ICE Process Description Capture Method(IDEF3)로 발표되었다. IDEF3의 목표는 특정 시스템이나 조직이 작동되는 방식에 관하여 그 부분에 대한 전문가의 지식을 표현하는 구조적 방법을 제공하는 것이다.

#### ■ 왜 프로세스 모델을 작성하는가 ?

프로세스 모델은 프로젝트의 사실 발견을 위한 시스템 분석 인터뷰에서 나오는 원시 데이터를 기록, 분석하는 시스템적 방법을 제공하는데 다음과 같은 것이 가능토록 지원한다.

- ① 기업에서 중요 운영 시나리오에 따라 정보 자원이 조직에 미치는 영향을 결정한다.
- ② 중요한 공유 데이터(특히 생산.엔지니어링.유지 보수.제품 정의 데이터)의 상태와 라이프 사이클에 영향을 주는 의사결정 절차의 문서화 메커니즘을 제공한다.
- ③ 데이터 구조 관리와 통제 정책 결정의 변화에 관한 정의를 내린다.
- ④ 시스템 디자인과 디자인 트레이드 오프(Trade-off) 분석을 지원한다.
- ⑤ 시뮬레이션 모델의 생산을 지원하는 강력한 메커니즘을 제공한다.
- ⑥ IDEF3 기능 모델의 개발을 위한 유용한 정보를 제공한다.

- ⑦ 사실.의사결정 시점.업무 분류 등을 명확하게 정의하는 메커니즘을 제공함으로써, 실시간 통제를 이룩하는 소프트웨어 디자인을 위한 프로세스 매핑(mapping)을 촉진한다.
- ⑧ 사용자 관점에서의 요구 사항과 개발하는데 필요한 데이터를 명확하게 정의하는 방법론을 분석가에게 제공한다.
- ⑨ 전문가 시스템 개발에 필요한 전문가의 관점을 포착하고 표현한다.

프로세스 모델링의 기본적인 전제는 개선의 노력이 제품 그 자체가 아닌 프로세스에 집중되어야 한다는 것이다. 그 것은 제품이 아니라, 제품을 만드는 프로세스만이 회사의 장기적인 성공을 보장할 수 있기 때문이다.

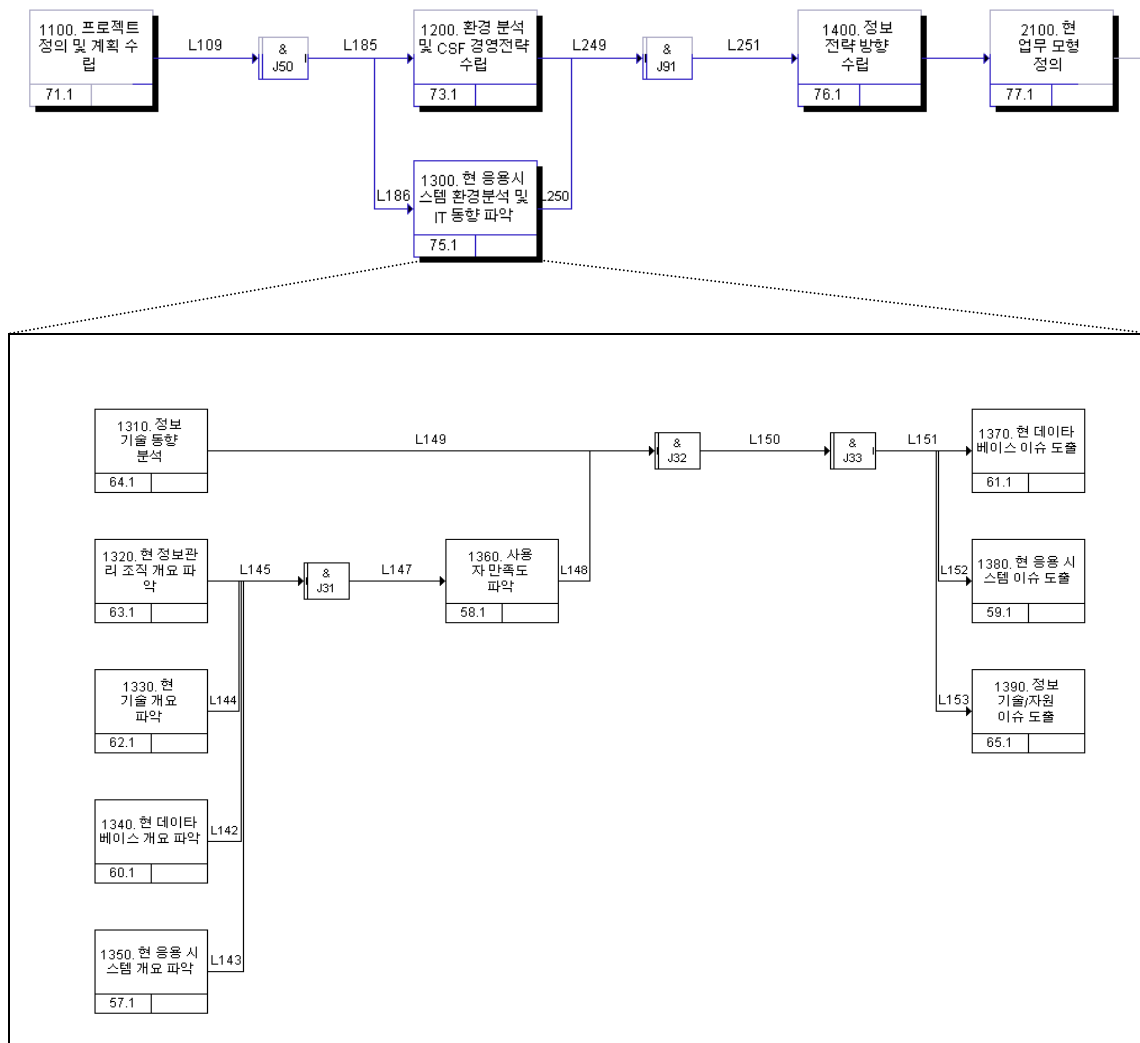


그림 4-1 : IDEF3 프로세스 모델의 예

■ IDEF3는 무엇인가 ?

IDEF3 모델링 방법론은 크게 두 가지의 분리된 방법으로 이루어져 있다.

- ① 프로세스 설명 포착방법
- ② 객체상태 전환 설명방법

IDEF3 은 시스템이 ‘어떤 방법으로 어떻게 되어야 한다’는 설명을 표현하고자 하는 필요에 의해 개발 되었으며 시스템에 관한 여러 사용자의 관점을 표현하고 조합하기 위한 언어로서, 시스템이 ‘주어진 조건에서 무엇을 얼마나 할지를 예측’하는 수치적 이상치를 알아내기 위한 시뮬레이션 언어군(즉 QGERT, SLAM 등) 과는 다른 목적에서 개발 되었다. 시뮬레이션 언어는 시간별로 달라지는 시스템 자원의 행동을 표현하고, 수량-모델-기반적(math-model-based) 시뮬레이션의 특성에 대한 틀을 제공한다. IDEF3 는 이에 반하여 시스템에 관한 구조적, 논리적 모델을 사용자에게 지원 함으로서 프로젝트 구성원간의 커뮤니케이션을 촉진하고 구축 되어야 할 시스템의 표현을 위한 구조적 틀을 제공한다. IDEF3 에는 두 가지 모델화 양식이 존재한다. 프로세스 흐름 묘사(Process Flow Description), 그리고 객체 상태의 전이 묘사(Object State Transition Description)이다. IDEF3 프로세스 모델링 방법은 기존의 혹은 제안된 시스템의 ‘행동적’ 측면에 관한 중요 전문가의 지식을 파악하려는 시스템 개발자들이 사용한다. IDEF3 를 이용하여 파악된 프로세스 지식은 시나리오의 전후관계 속에서 구조화 되며, IDEF3 는 시스템의 묘사를 위한 직관적 지식 습득 도구의 역할을 수행한다. 일반성과 단순성을 촉진하기 위하여, 시스템의 단일 관점을 채택하여 모든 일시적 논리를 확실히 제거하는 IDEF0모델과 달리, IDEF3 는 기업 프로세스에서 발생하는 일시적 원인과 결과에 관한 여러 사용자의 설명을 수용, 지원한다. 결과적으로 IDEF3 설명은 분석이나 설계를 위한 모델이 구축될 수 있는 구조적 지식 기반을 제공한다.

■ IDEF3는 무엇을 표현하는가 ?

프로세스 흐름 묘사는 조직 내에서 ‘일하는 방식’에 관한 지식을 파악하려는 것이다. 객체 상태 전환 묘사는, 객체가 특정 프로세스를 통과할 경우 발생 가능한 전환 상태를 표현한다. 프로세스 흐름 묘사와 객체 상태 전환 묘사는 모두 묘사를 구성하는 정보 단위를 내포한다. 이들 모델 실체(entities)는 IDEF3 설명의 기본적 단위를 구성한다.

이 결과 다이어그램과 텍스트로 이루어지는 ‘묘사(Description)’는 다른 IDEF 방법에서 만들어지는 ‘모델(Model)’과는 상이한 개념인데 이 차이는 중요하다.

앞서 검토했듯이 모델은 주어진 현실 세계 시스템의 중요한 측면을 모방하도록 디자인 된,

객체, 소속, 관계에 대한 하나의 개념적 시스템이다. 아주 현실적 의미에서 모델은, 현실적 시스템의 가정화. 단순화를 중심으로 구축된 시스템 그 자체이며 그것은 모델이 현실적으로 적용되는 예측된 상황의 범위 내에서 확실히 유효하다고 가정하고 있다. 따라서 모델은 그 유용성을 보증하기 위하여 완전해야 하며 내부적으로 일관되어야 한다.

그러나 묘사는 일반적으로 불완전하다. 예를 들어 시스템 분석 활동에 있어서 분석자에게 제공되는 각 부문 전문가의 설명은 설명자 자신의 관점에서 인식된 사실에 대한 표현일 뿐이며 이는 한정된 범위 내에서만 일반적인 모델로서 전환되어질 수 있다. 즉 이러한 묘사에서는 전문가는 전문 영역 밖에 있는 프로세스나 사건에 대하여 일부를 알 수는 있겠지만 그 전체적 사실을 알 수 없는 경우가 많으며, 전문가와 관계가 없거나 그 순간 단순히 잊혀졌기 때문에 주어진 설명에서 생략될 가능성이 있다. 묘사는 단지 불완전하지만 진실된다고 가정된 우리의 주변 세계에 관한 사실과 믿음에 대한 기록일 따름이다. 모델이 일정한 척도와 규칙에 따라 작성되어 공식적으로 인정된 지도라면 묘사는 여러분이 알고있는 한 지점에서 주변의 다른 지점으로 이동하기 위한 경로를 순차적(시간) 서술적으로 기술하는 시나리오(약도)와 같다. 시스템 분석 활동의 목적이 현 시스템의 기능, 정보, 프로세스에 관련된 사실과 관계를 정확하게 표현하는 모델의 구축에 있다면 이에 앞서서, 유력한 전문가에 의해 제공되는 묘사의 수집은 필수적인 것이다.

4.1.2 FLOW CHART와 IDEF3 프로세스 모델의 비교

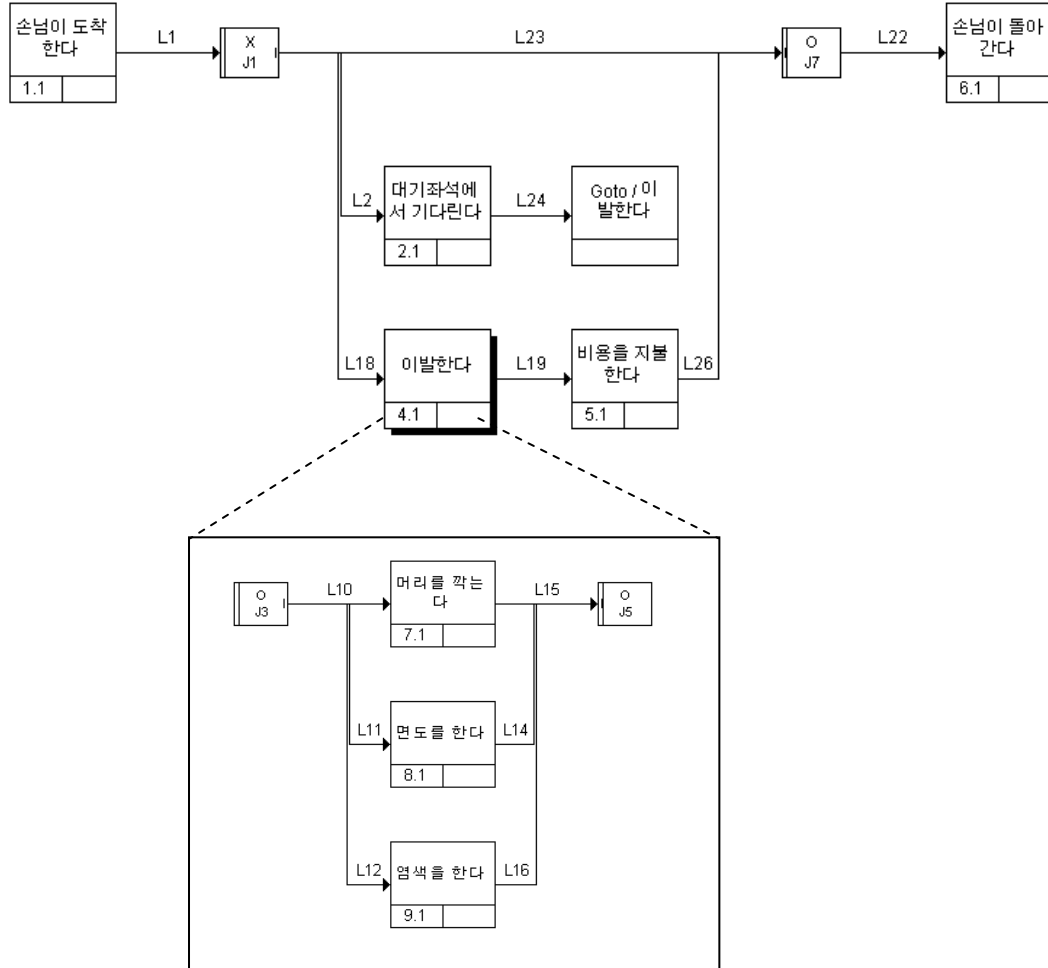


그림 4-2 : 이발소업무 프로세스

우리는 플로우차트(Flow Chart) 와의 비교를 통해서 더욱 쉽게 IDEF3 프로세스 모델을 이해할 수 있다. 그림 4-2 의 IDEF3 프로세스 다이어그램이 나타내고 있는 내용은 우리가 일반적으로 접할 수 있는 이발소의 프로세스를 이발소 주인의 관점에서 나타내고 있다. 구현된 시나리오의 내용을 이발소 주인과의 인터뷰를 바탕으로 서술적으로 나타내면 다음과 같다.

- ① 이발소에 손님이 도착한다.
- ② 도착한 손님은 대기하는 손님이 많은 경우 그냥 돌아가기도 하고
- ③ 대기좌석에 앉아 기다리는 경우 도 있는데 기다리다가 차례가 되면 이발을 한다.
- ④ 대기하는 손님이 없는 경우 도착한 손님은 바로 이발대에 앉아 이발을 시작한다.
- ⑤ 이발소에서 제공하는 이발 서비스는 세가지로 나눌 수 있는데 특별한 순서 없이 손님

의 요청에 따라 머리를 깎거나, 면도를 하거나, 염색을 하는 것이다. 물론 손님에 따라 이중 두가 지나 세가지를 모두 하기도 한다.

⑥ 이발을 마친 손님은 비용을 지불한 후 이발소를 떠난다.

우리는 위의 IDEF3 프로세스 다이어그램을 통하여 이발소에서 발생하는 업무프로세스를 파악할 수 있다.

여기서 우리는 IDEF3의 특성중 하나인 두 개의 논리적 분기를 찾아 볼 수 있는데 그 하나는 도착한 손님이 그냥 돌아가던지, 대기좌석에 앉아 기다리던지, 아니면 바로 이발대에 앉아 이발을 시작하던지 중에서 한가지만 선택할 수 있다는 배타적 선택논리(J1으로 표시된 작은 박스)가 그 하나이고 다른 하나는 이발을 시작한 손님이 제공되는 서비스(커트, 면도, 염색) 중에서 한가지 이상을 원하는바에 따라 선택이 가능하고 이들 서비스가 동시에 수행되어야 하는가를 나타내는 시간적 동시성의 논리이다(J3로 나타난 작은 박스). IDEF3 프로세스 다이어그램은 프로세스의 흐름에서 우리가 익히 알고있는 논리게이트(AND/OR/XOR)와 같은 프로세스의 분기의 구조와 함께 이러한 분기 이벤트의 동시성 여부(Synchronous / Asynchronous)를 표현 할 수 있는 방법을 지원 함으로써 프로세스 흐름의 확실한 시간적 논리적 표현이 가능도록 한다.

IDEF3의 또 다른 하나의 특성은 프로세스의 분해를 통한 레벨구조의 지원인데 그림 4-2는 ‘이발을 한다’는 프로세스를 분해하여 레벨화 함으로써 특정 프로세스의 계층화 구조를 보여주고 있다. 이러한 계층구조의 지원은 모델 작업자로 하여금 추상화된 다양한 레벨을 조작할 수 있도록 지원하여 프로세스 표현의 융통성을 확장한다. 또한 IDEF3 프로세스 다이어그램은 이러한 시나리오적 다이어그램을 바탕으로 각 프로세스에 관여되는 객체들에 대한 다음과 같은 정보를 수집하는데 이는 이러한 프로세스를 기반으로 구축되어질 전문가의 지식을 수집하고 객체중심의 프로세스 표현방법인 시뮬레이션 모델생성의 기초를 제공한다.

- ① 어디에서 프로세스가 수행되는가?
- ② 누가 혹은 어떤 장치가 이러한 프로세스의 수행에 관여하는가?
- ③ 프로세스의 수행으로 생성되거나 변형되어질 부품이나 정보는 어떤것인가?
- ④ 프로세스의 수행으로 인하여 가공되는 부품이나 정보의 상태는 어떻게 변하는가?

이와 같은 프로세스와 관련된 객체의 정보를 표현하는 구체적인 방법은 다음 절에서 논하도록 한다.

IDEF3 프로세스 다이어그램과 Flow Chart의 비교를 돕기위해 위에 그림 4-2를 다시 그림 4-3과 같이 Flow Chart로 표현했다.

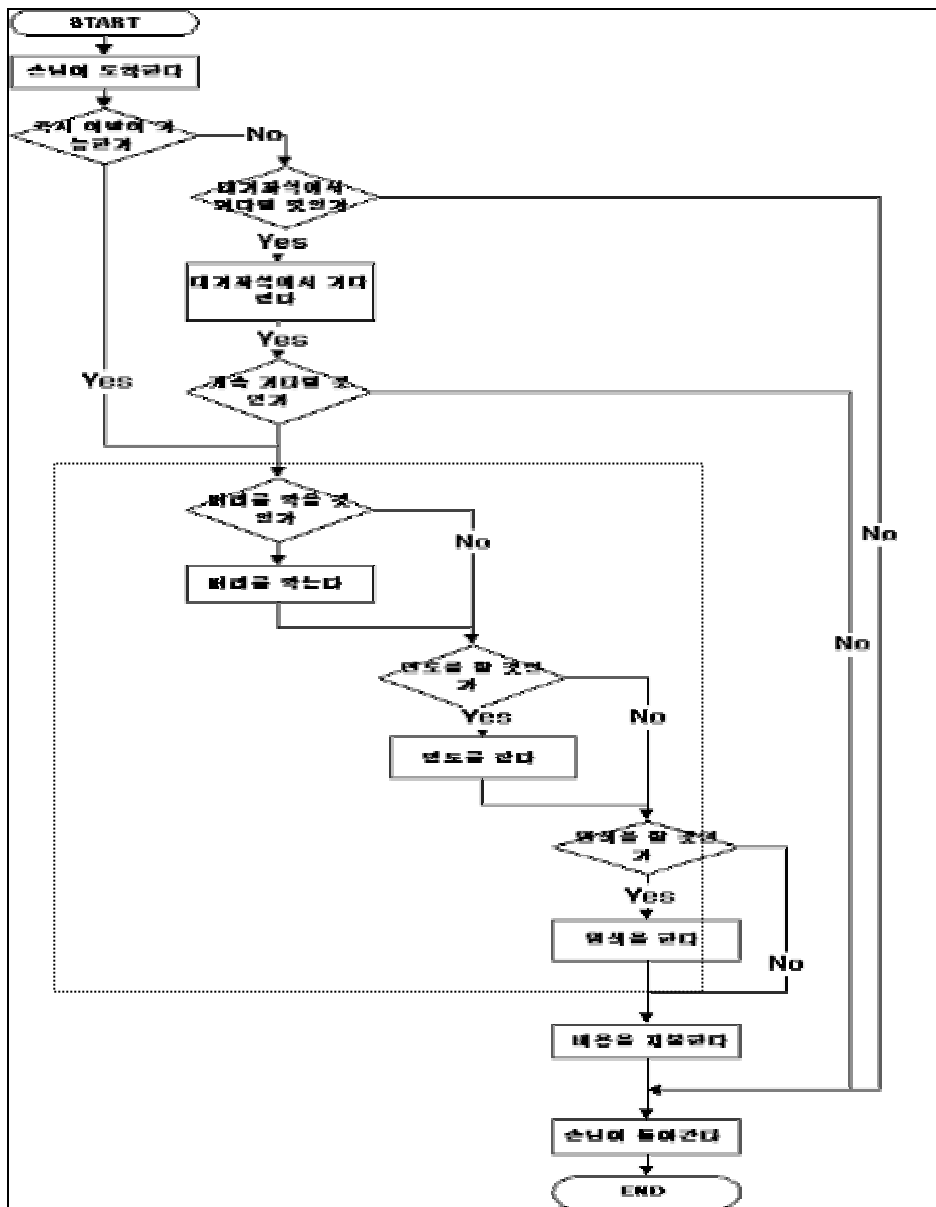


그림 4-3 : FlowChart

■ Flow Charting 과 Process Modeling 의 차이점

Flow Charting	Process Modeling
<ul style="list-style-type: none"> <li>• 모호한 방법의 프로세스 논리전개</li> <li>• 추상화 된 다양한 레벨을 포착할 수 없다.</li> <li>• 프로세스 내부에서 개체의 정보를 제공하지 않는다.</li> </ul>	<ul style="list-style-type: none"> <li>• 모호하지 않은 문법의 프로세스 논리전개</li> <li>• 추상화 된 다양한 레벨의 포착</li> <li>• 프로세스에 개체와 시뮬레이션 자료를 등록할 수 있다</li> </ul>

## 4.2 IDEF3 모델링 표현 방법

IDEF3 프로세스 흐름 묘사는 특정 시나리오의 전후관계 속에서 행동간에 관계되어진 네트워크를 파악한다. 이 묘사의 목적은 특정한 문제 해결(혹은 발생) 상황의 전후 관계 속에서 특정 조직 내의 일 처리 방식을 보여 주는 것이다. IDEF3는 프로세스 묘사를 위한 경계 조건과 초점을 확립하는 기본적 조직 구조로서 '시나리오'를 활용한다. 이러한 특성은 인간이 어떤 상황 속에서 경험하거나 관찰하여 알게 된 것이나, 예정하고 있는 계획에 관한 내용을 활동의 시간적 순서에 의해 묘사하려는 인간적 경향에 기반을 두고 있다. 프로세스 묘사의 전후관계 속에서 생각과 표현을 정리하려는 이러한 자연적 경향은 어떤 설정된 조건에서 행동의 구성을 표현하는 역할로서, '외부적 관점'에서 볼 때 구현 가능한 시스템의 디자인을 제안하기 위한 비공식적 틀(Framework)에서 시나리오의 광범위한 사용을 촉진시켰다. 이러한 개발 방식을 '외부 조건 지향적 디자인' 방식이라고 하며 실행 결과 새로운 시스템 디자인을 위한 효과적 메커니즘이란 점이 계속적으로 입증되었다.

### 4.2.1 다이어그램 문법

IDEF3 프로세스 흐름 다이어그램은 다음과 같은 네 가지의 요소로 구성되며 그림 4-4 는 IDEF3 프로세스 흐름 다이어그램의 예를 보여 주고 있다.

- ① 행동 단위(UOB, Units of Behavior) : 박스 형태로 표현되는 행동 단위
- ② 연결(Links) : 화살표로 표현되는 UOB 의 선후 연관 관계
- ③ 접속(Junctions) : UOB 의 분기 및 결합 시 사용되는 시간, 논리적 관계
- ④ 참조 사항(Referents) : UOB 와 관련된 부가적 내용을 도형적으로 표현하는 박스

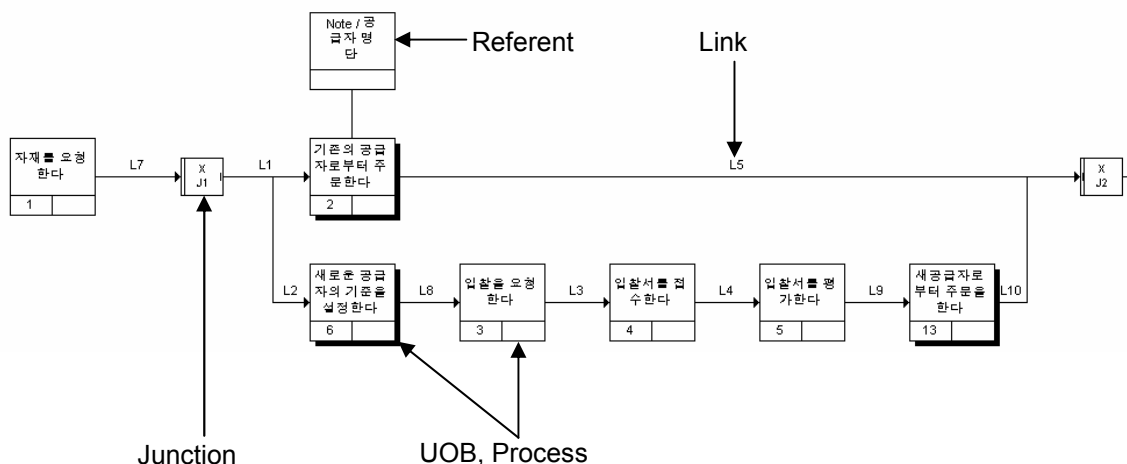


그림 4-4 : IDEF3 Process Flow Diagram 의 예



■ 4.2.1.1 UOB(Unit of Behavior)란 무엇인가?

프로세스는 현실세계 시스템에서 발생하는 기능이나 절차의 일반적인 표현으로서 IDEF3 에서 프로세스는 다음과 같은 것들을 표현한다.

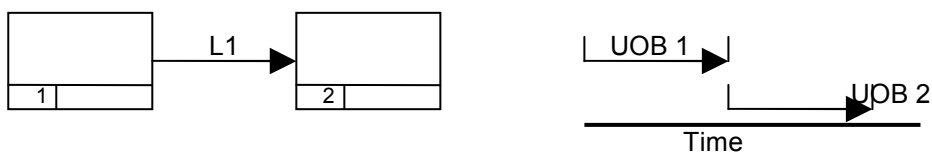
기능(Functions)	행동(Actions)	프로세스(Processes)
활동(Activities)	행위(Acts)	운영(Operations)
사건(Events)	시나리오(Scenarios)	의사결정(Decisions)
절차(Procedures)		

IDEF3 프로세스 흐름 다이어그램은 그 주변 구조에 따라 기능, 활동, 행위, 프로세스, 운용, 사건, 시나리오, 의사결정, 절차 등으로 정의 될 수 있는 UOB (Units of Behavior)를 기본적인 문장 구성 단위로 한다. 주어진 시나리오의 전후관계 속에서 사용되는 각 UOB 는 이들 시나리오를 구성하는 표현 행위로서, 각 UOB 는 사건의 인지된 상태 혹은 주어진 시나리오와 연관된 개체의 변화상태에 따라 세상에 대한 특정한 관점을 표현한다. IDEF3 에서, 각각의 UOB 는 다이어그램 내에서 단위행위를 나타내는 이름이 붙은 하나의 박스로 표현된다. 하나의 식별번호는 UOB 가 생성될 때마다 순차적으로 붙여지며 각 박스의 왼쪽 아래에 표시된다. 그림 4-4 에서 우리는 시나리오를 구성하는 요소로서 ‘자재를 요청한다’, ‘기존의 공급자로부터 주문한다’, ‘새로운 공급자의 기준을 설정한다’와 같은 사건의 시간적 흐름을 표현하는 UOB 를 발견할 수 있다.

■ 4.2.1.2 링크(Link) 란 ?

각 프로세스 흐름 다이어그램의 엘리먼트 사이에 연결된 링크는 모델 시스템 안에서 각 엘리먼트 간의 관계를 나타낸다. 이들의 목적은 프로세스간의 순간적, 논리적, 협력적, 혹은 자연적인 제약관계를 표현하기 위함이다. UOB 간의 관계는 Link 와 Junction 을 통하여 서로 연결되는데 IDEF3 에서 Link 는 다음과 같이 분류된다.

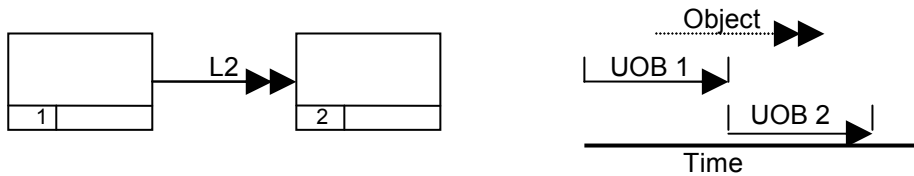
① 시간적 선행(Simple Precedence or Temporal Precedence) Link



시간적 선행 Link 는 하나의 프로세스 타입의 인스턴스와 다른 것 사이에 명료한 선후절차

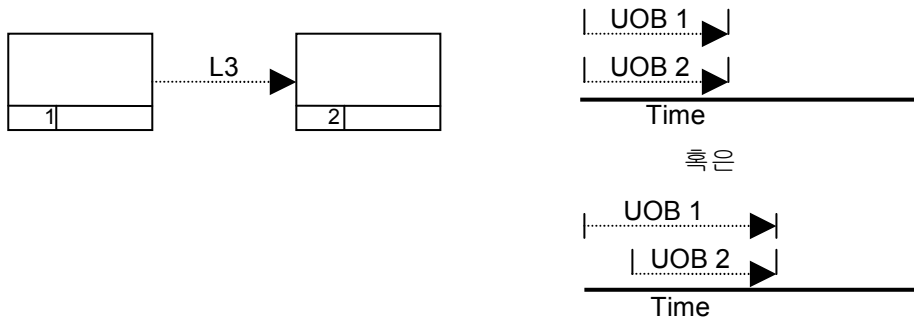
가 있는 것을 나타낸다. 각 선행 프로세스는 후행 프로세스가 시작되기 전에 종료되어야 하며, 후속 UOB 의 수행에 앞서 선행 UOB 가 완료됨이 확실할 경우로 실 선의 화살표로 표시된다.

② 개체의 흐름(Object Flow) Link



개체의 흐름 Link 는 두 프로세스 인스턴스 사이에 특정 개체(object)가 관여하고 있음을 나타낸다, 시간적 선행 Link 와 같은 관계나 프로세스의 수행과 함께 선행 UOB 에서 새롭게 생성된 중요 개체가 후속 UOB 로 이동 될 경우로 개체의 이동 여부가 후속 UOB 수행에 전제 조건으로 작용할 경우에 사용되는데 실 선에 두개의 화살표로 표시한다. 개체의 흐름 Link 로 그려지지 않았다고 해서 두 프로세스 사이에 어떤 개체도 같이 관여되고 있지 않다는 것은 아니다.

③ 관계적(Relational) Link



관계적 Link 는 일시적 선행 Link 와 객체의 흐름 Link 에 의해서는 수용되지 않는 제약을 파악하기 위해 제공되는데 연결된 엘리먼트 사이에 연결관계가 정하여지지 않았음을 나타낸다. 즉 전후 UOB 간의 완료 관계가 명확치 않을 경우를 표현하며 점선으로 표시된다. 이 링크 타입은 시간적 선행 링크와 같이 미리 정하여진 의미는 없다. 대신에 관계적 링크는 하나의 프로세스가 다른 프로세스가 시작되기 전에 시작되거나 늦어도 후행 프로세스와 같이 종료됨을 나타내고 있다. Link 의 번호는 다이어그램에 등록된 순서에 따라 L1,L2,L3 ... 와 같이 유일한 번호 체계를 갖는다.

■ 4.2.1.3 접속(Junction)

접속(Junction)은 프로세스 흐름에 있어서 UOB 의 분기/결합 관계를 표현하는데 접속은 하나의 UOB 가 두개 이상의 후속 UOB 로 분기되는 Fan Out(Divergence) Junction 과 두개 이상의 선행 UOB 가 하나의 UOB 로 결합되는 Fan In(Convergence) Junction 이 있다. junction 의 왼쪽에 나타난 짧은 수직선은 Fan-in Junction 을 오른쪽에 표시된 짧은 수직선은 Fan-out Junction 을 표시한다. 또한 이러한 접속은 각각 분기와 결합의 논리적 관계를 표현하기 위하여 AND, OR, XOR(다이아그램에서 각각 &, O, X 로 표현됨)로 구분되며 시간적 동기성 여부를 나타내는 Synchronous 와 Asynchronous(junction 박스 좌우의 수직선으로 표현되는데 Synchronous 는 양쪽에 수직선이 Asynchronous 는 왼쪽에만 수직선이 있다)로 구분된다.

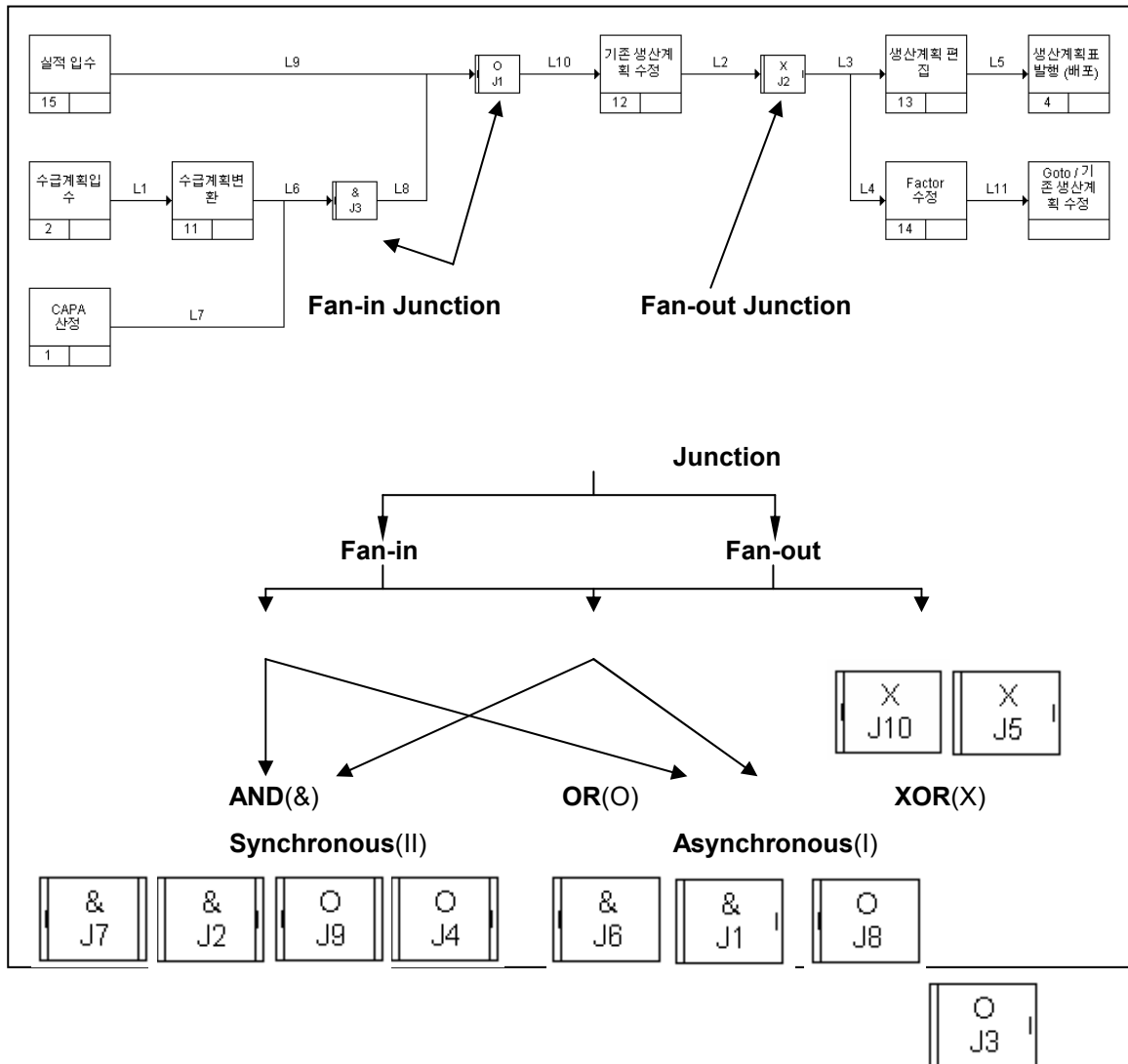
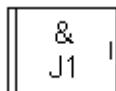


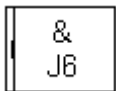
그림 4-5 : Junction 의 구조

관련된 UOB 간의 분기 및 결합에 있어서 논리적 관계 및 시간적 동기/비동기성 관계를 도형으로 표현하기 위하여 IDEF3 프로세스 다이어그램에서는 발생 가능한 순차적인 모델로서 다음과 같은 5 개의 논리적 타입을 지원한다.

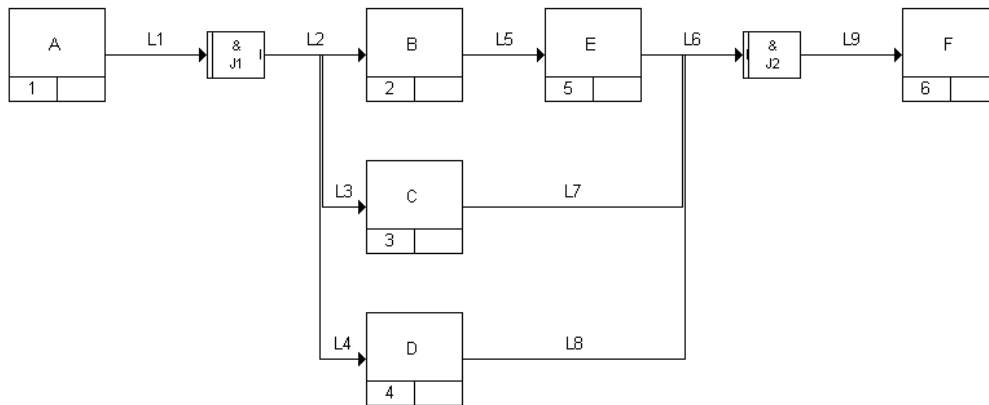
- ① **Asynchronous “AND”** 접속은 흐름이 계속되기 위하여 이 접속의 앞이나 뒤에 연결된 모든 프로세스가 수행될 것임을 – 비록 모든 것이 동시에 수행될 필요는 없지만 – 나타낸다. 그림에서 **J1 junction** 은 프로세스 A의 수행에 이어서 프로세스 B, C, D가 동시에 수행될 필요는 없지만 모두 수행되어야 한다는 것을 나타내며 **J2 junction** 은 프로세스 F의 수행에 앞서 프로세스 E, C, D가 모두 수행되어야 한다는 것을 나타내고 있다. 물론 E, C, D가 동시에 완료될 필요는 없다.



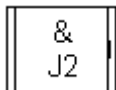
(Fan Out) 선행 프로세스의 수행은 모든 후속 프로세스를 기동한다.



(Fan - In) 모든 선행 프로세스가 종료되어야 후속 프로세스가 수행된다.



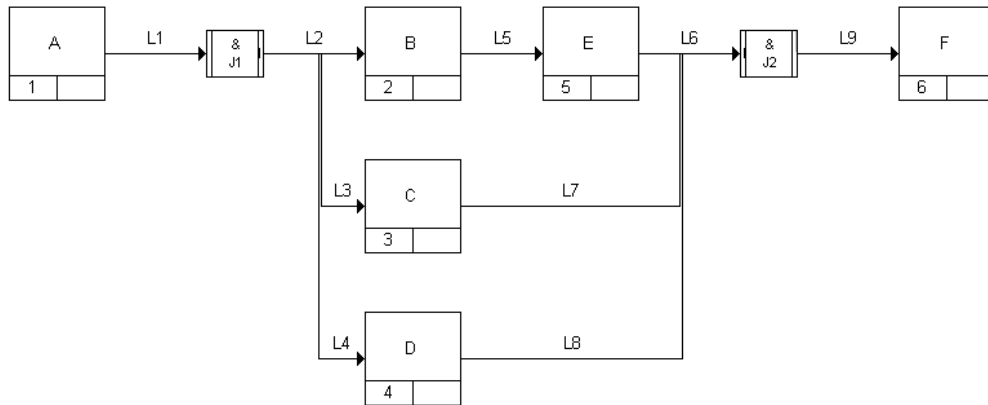
- ② **Synchronous “AND”** 접속은 흐름이 계속되기 위하여 접속의 앞이나 뒤에 연결된 모든 프로세스가 동시에 모두 수행되어야 함을 나타낸다. 그림에서 **J1 junction** 은 프로세스 A의 수행에 이어서 프로세스 B, C, D가 모두 동시에 수행되어야 한다는 것을 나타내며 **J2 junction** 은 프로세스 F의 수행에 앞서 프로세스 E, C, D가 모두 동시에 완료되어야 한다는 것을 나타내고 있다.



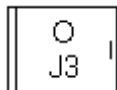
(Fan Out) 선행 프로세스의 수행은 모든 후속 프로세스를 동시에 기동한다.



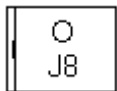
(Fan - In) 모든 선행 프로세스가 동시에 종료되어야 후속 프로세스가 수행된다.



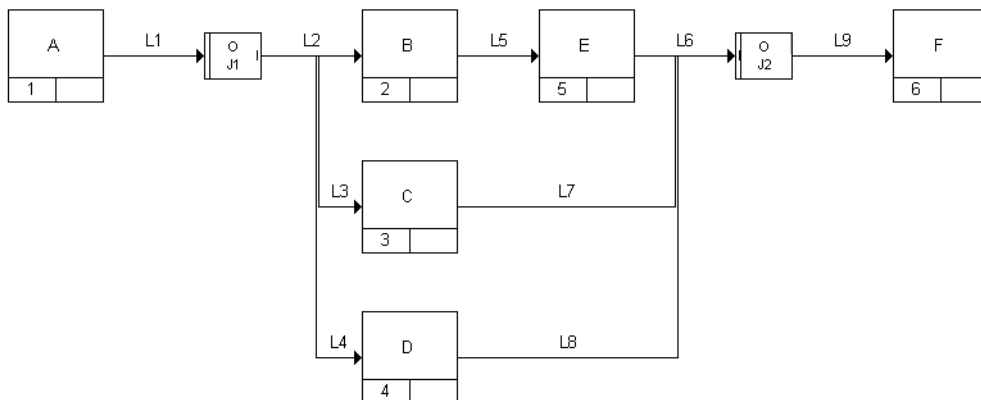
- ③ **Asynchronous “OR”** 접속은 흐름이 계속되기 위하여 접속의 앞이나 뒤에 연결된 하나 이상의 프로세스가, 동시에 수행될 필요는 없지만, 수행되어야 함을 나타낸다. 그림에서 **J1 junction** 은 프로세스 **A** 의 수행에 이어서 프로세스 **B, C, D** 중 하나 이상의 프로세스가 수행되어야 한다는 것을 나타내며 **J2 junction** 은 프로세스 **F** 의 수행에 앞서 프로세스 **E, C, D** 중 하나이상의 프로세스가 완료되어야 한다는 것을 나타내고 있다. 물론 시간의 동시성은 배제하고 있다.



(Fan Out) 선행 프로세스의 수행은 하나 이상의 후속 프로세스를 기동한다.

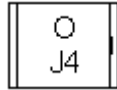


(Fan - In) 하나 이상의 프로세스가 완료되어야 후속 프로세스가 수행된다.

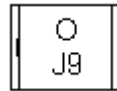


- ④ **Synchronous “OR”** 접속은 흐름이 계속되기 위하여 접속의 앞이나 뒤에 연결된 하나 이상의 프로세스가 동시에 수행되어야 함을 나타낸다. 그림에서 **J1 junction** 은 프로

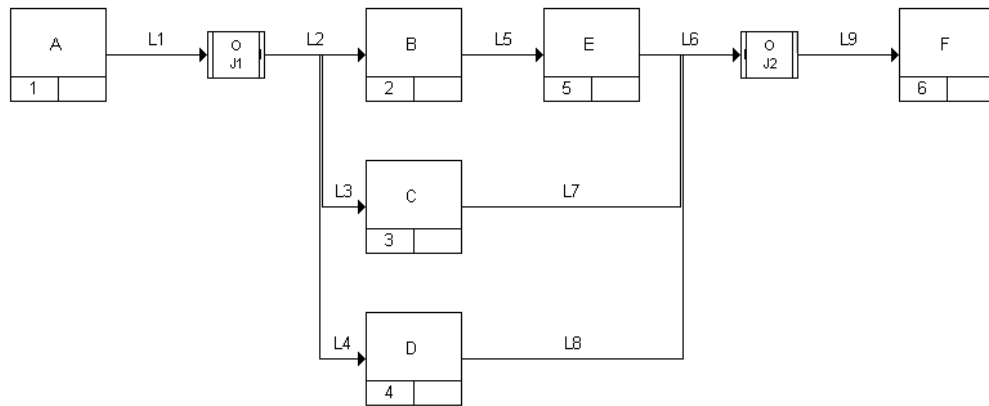
세스 A 의 수행에 이어서 프로세스 B, C, D 중 하나 이상의 프로세스가 동시에 수행되어야 한다는 것을 나타내며 J2 junction 은 프로세스 F 의 수행에 앞서 프로세스 E, C, D 중 하나이상의 프로세스가 동시에 완료되어야 한다는 것을 나타내고 있다.



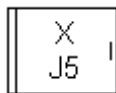
(Fan Out) 선행 프로세스의 수행은 하나 이상의 후속 프로세스를 기동한다.



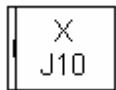
(Fan - In) 둘 이상의 선행 프로세스가 동시에 완료되어야 후속 프로세스가 수행된다.



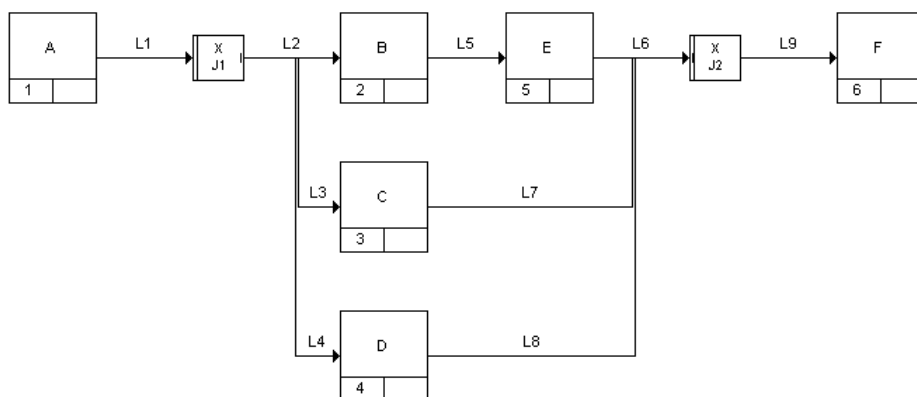
- ⑤ “XOR” 접속은 흐름이 계속되기 위하여 접속의 앞이나 뒤에 연결된 오직 하나의 프로세스만이 수행될 수 있음을 나타낸다. 그림에서 J1 junction 은 프로세스 A 의 수행에 이어서 프로세스 B, C, D 중 하나의 프로세스만이 수행 가능하다는 것을 나타내며 J2 junction 은 프로세스 F 의 수행을 위하여 프로세스 E, C, D 중 하나만 완료되면 가능하다는 것을 나타내고 있다. 물론 배타적 접속에 있어서 시간의 동시성은 용납되지 않는다.



(Fan Out) 선행 프로세스의 수행은 단지 하나의 후속 프로세스만을 기동한다.



(Fan - In) 단지 하나의 프로세스만 완료되면 후속 프로세스가 수행된다.

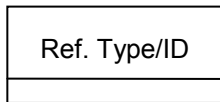


■ 4.2.1.4 참조(Referent)

IDEF3 프로세스 흐름 다이어그램은 판독자의 주의를 위하여 각 UOB 와 관련된 중요 정보를 'Referent'라는 도형으로 표시 할 수 있도록 지원한다. Referent 는 각각의 UOB 가 수행됨에 있어서 관련되는 정보를 박스 형태의 도형으로 지원 함으로서 프로세스 다이어그램 판독자가 UOB 에 관여된 중요 정보를 인지할 수 있도록 한다.

각 Referent 는 Referent Type/ID 형태로 표현되는데 Referent 의 종류에는 UOB 의 수행과 관련된 시간 관계에 따라 다음과 같은 세 종류가 있다.

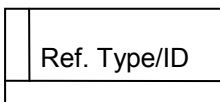
① Unconditional(Information, Go\_to)



**Information** : 단순한 정보가 등록된 Referent 로 관련된 Process, Scenario, Object, OSTN, Note 등의 내용을 참조하는데 ID 로 관련 레이블을 표시한다.

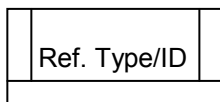
**Go\_to** : 해당 UOB 에서 항상 Referent 에 표시된 Process, Scenario, OSTN, Junction 으로 분기되어짐을 표시한다.

② Asynchronous(Call & Continue)



**Call & Continue** : 해당 UOB 의 수행 시 Referent 에 표시된 Process, Scenario, OSTN 등을 호출하고 피호출 엔트리의 응답 여부와 상관 없이 계속 진행한다.

③ Synchronous(Call & Wait)



**Call & Wait** : 해당 UOB 수행 시 Referent 에 표시된 Process, Scenario, OSTN 등을 호출하고 피호출 엔트리의 응답에 따라 진행 여부를 판단한다.

그림 4-6 에서 우리는 세 가지의 Referent 를 발견하게 되는데 '페인트를 칠한다' UOB 에 부착된 Information referent 인 'Object / 소형차 도장라인'은 '소형차 도장라인'이 UOB 에 단순

이 Object 로 관련되어 있음(Unconditional Referent)을 나타낸다. 그러나 ‘도장상태를 검사한다’ UOB 에 부착된 ‘Process Flow / 시료를 검사한다’ Referent 는 UOB 수행 시 ‘시료를 검사한다’로 명명된 Scenario 를 호출 후 시료검사 결과와 관련된 별도의 응답이 있을 때 까지 대기상태로 유지됨(Synchronous / Call & wait)을 표현하고 있으며, ‘부품을 출고한다’ UOB 에 부착된 ‘Process Flow / 도장완료 수량을 입력한다’ Referent 는 ‘도장완료 수량을 입력한다’로 명명된 시나리오를 호출하고 결과에 상관없이 계속 수행된다는 것(Asynchronous / Call & Continue)을 나타내고 있다.

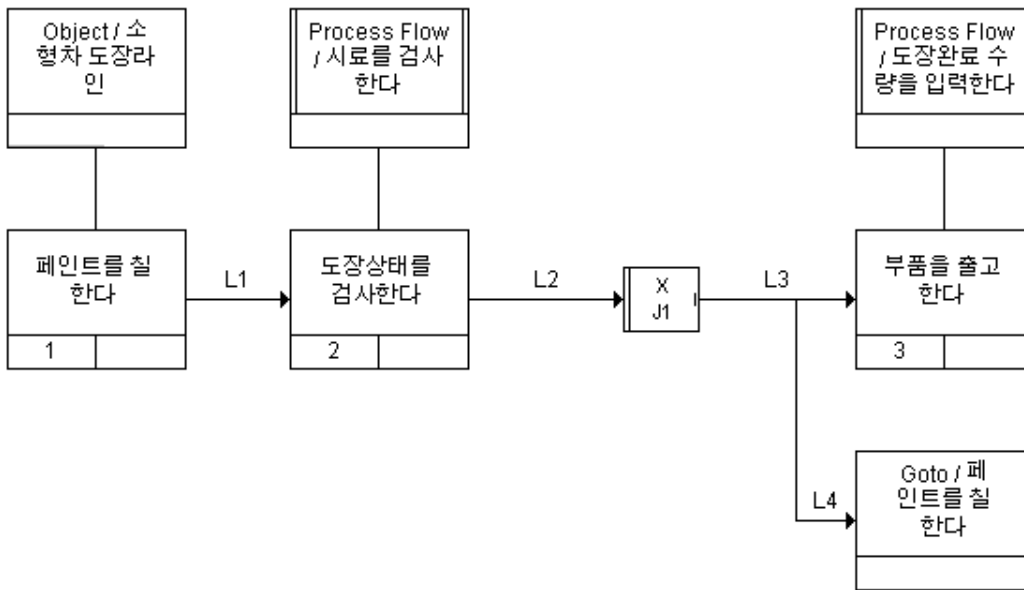
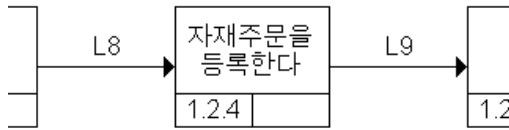


그림 4-6 : Referent 사용의 예

■ 4.2.1.5 상세화(Elaboration)

각 UOB 는, UOB 에 관련된 내용을 기술한 서술적 표현의 집합인 ‘상세 설명(Elaboration)’에 의하여 구체화 된다. 우리는 Elaboration 을 이용하여 프로세스에 관련된 추가적인 정보를 등록할 수 있는데 Elaboration 에서는 프로세스와 관련된 개체(object), 사실(fact), 제약사항(constraint), 노트(note), 자료의 출처를 알려주는 소스(source) 그리고 관련된 사항을 서술적으로 표현하는 설명을 추가할 수 있다. Elaboration 은 UOB 에 대한 자세한 설명으로서 다이어그램에서 표현하기 복잡한 서술적 내용을 별도의 양식에 표현하며 각각의 UOB 와 한 쌍을 이루는데 그 내용은 그림 4-7 과 같다.





**Elaboration :**

**UOB Name :** 자재주문을 등록한다

**Objects :** 등록요청서(Entity)  
 등록담당자(Resource)  
 컴퓨터 터미널(Resource)  
 등록접수실(Location)

**Facts :** 수입자재의 경우 외자과의 확인을 받는다.

**Constraints :** 매일 15 시 이전에 전산등록을 완료한다.

그림 4-7 : Elaboration 의 예

① 개체(Obects) : UOB 의 수행과 관련된 Object 를 나타내며 Object 의 종류에 따라 Entity, Location, Resource, Queue, Transport, Logical 로 분류된다. 개체는 프로세스나 접속에 관련된 물리적 혹은 개념적인 요소로서 기본적으로 프로세스를 이루는 것들이다. 프로세스가 수행되는 동안에 개체는 만들어지고 변경되며, 소멸된다. 또한 우리는 프로세스와 연관된 많은 개체들(부품, 장소, 사람, 기계, 정보 등)에 관한 정보를 모델 판독자에게 전달하여야 하는데 주로 다음과 같은 목적을 달성하기 위하여 개체와 관련된 정보를 파악한다.

- 프로세스가 무엇을 생산하는가? (entity)
- 어디에서 프로세스가 수행되는가? (loaction)
- 어떻게(어떤 자원을 사용하여) 생산을 수행하는가? (resource)
- entity 는 어디에 저장되며 그 용량은 얼마인가? (Queue)
- entity 의 운반에 관련된 개체는 무엇인가? (Transport)
- 프로세스의 수행과 관련된 논리적 제약은 무엇인가? (Logical)

② facts & constraints : Fact 와 Constraint 는 모두 프로세스의 수행에서 관찰될 수 있는 상태를 설명하는데 Fact 는 프로세스의 특정 상태에서 관찰될 수 있는 상황을 말하며

**constraint** 는 특정상태가 아닌 프로세스의 수행에서 항상 일어나야 될 것을 표현한다. 그림 4-7 의 예에서 ‘수입자재의 경우 외자과의 확인을 받는다.’는 **Fact** 는 프로세스가 수입자재 주문을 등록하는 특정 상황에 관한 제약이며, **Constraint** 로 등록된 ‘매일 15 시 이전에 전산등록을 완료한다.’는 특정 조건이 아닌 일반적인 지침으로서의 제약사항을 나타내고 있다. 프로세스 모델의 구축에 있어서 다양한 **Fact** 의 추출 및 기록은 **constraint** 를 찾아내기 위한 좋은 방안이 된다.

- ③ 소스(**source**) : 프로세스와 관련된 기초자료나 정보의 출처를 나타낸다.
- ④ 노트(**note**) : 모델 검토의 결과인 주석을 나타낸다.
- ⑤ 설명(**Description**) : UOB 에 대한 일반적 설명을 기술하며 프로세스에 관한 전반적인 설명을 포함한다.

#### 4.2.2 객체상태 전이 다이어그램 (Object State Transition Network Diagram)

IDEF3 에서 지원하는 또 하나의 다이어그램은 프로세스의 수행과 함께 변환되는 Object 의 전환 상태를 표시하는 객체상태 전환 다이어그램이다. 객체상태 전환 다이어그램은, 프로세스를 객체 중심적 관점에서 파악하기 위해 사용하는데 이는 프로세스 다이어그램 전반에 걸친 영역 내에서 객체의 허용 가능한 전환상태를 요약 표현하는 것이다.

객체상태의 전환 다이어그램은 두 가지로 구성되는데 첫째는 둥근 원으로 표시되는 객체의 상태와 둘째는 객체상태 간에 연결되어 그 전환관계를 표시하는 화살표(Transition Arc)이다. 그림 4-8 은 객체상태 1(Object State #1) 이 객체상태 2(Object state #2)로 전환되는 과정에서 Action2 라는 UOB(Referent)가 관여하고 있음을 보여 주고 있다.

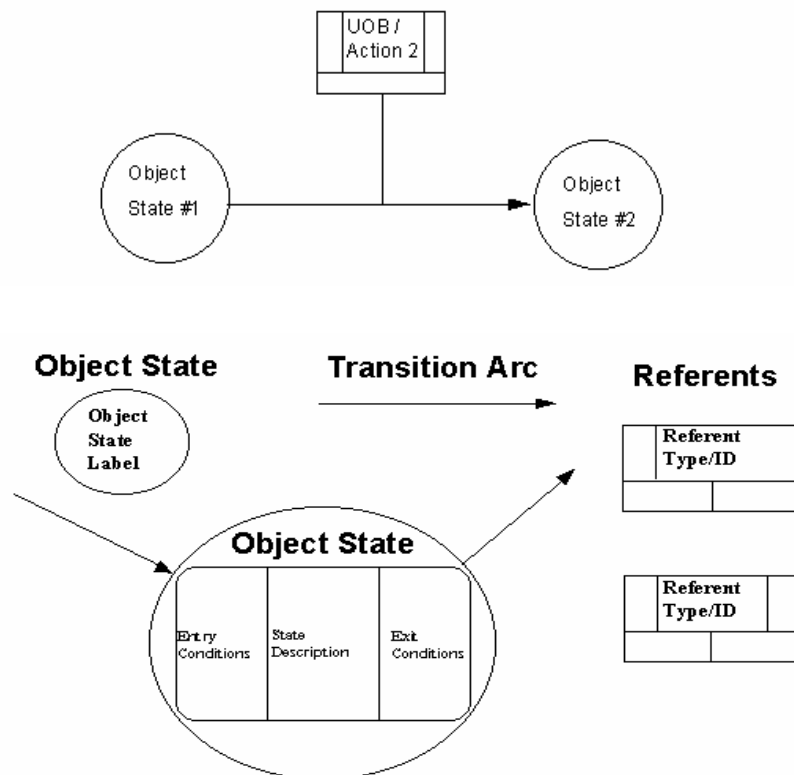


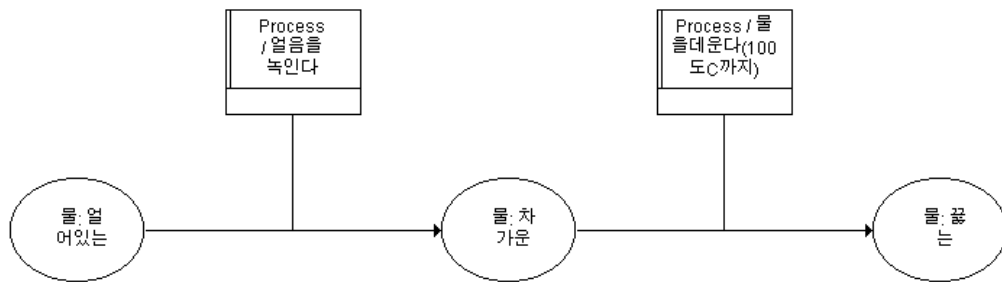
그림 4-8 : Object State Description

각 객체상태는 객체 상태를 유지하기 위한 제약조건과 그 값으로 정의되어진다. 정보 시스템 내에서 연속되는 객체상태의 값들은 IDEF1 모델에서의 속성과 같이 정의되어야 하며 이러한 값들은 객체상태 전환 다이어그램과 상호 연관되어 있다. 하나의 객체상태는 그 상태로 되기 위한 전환 전의 제약조건(entry condition, 입장조건)과 다음 상태로 전환되기 위

한 전환 후의 제약조건(exit condition, 퇴장조건)을 가지고 있다. 이 제약조건들은 상태전환이 시작될 수 있기 전의 상태와 상태전환이 완료될 수 있기 전의 상태를 규정한다. 제약조건은, 간단한 성질 과 값의 쌍(pair) 리스트 혹은 제약조건에 대한 설명에 의해 구체화 된다.(예를 들면 - 승인된 자재주문, 발주된 자재주문) 속성의 값은 그것이 되기 위하여 요구되는 특정한 값과 연결되어야 한다(예를 들면 - 승인된 자재주문:승인번호, 발주된 자재주문:발주번호).

상태를 정의하는데 필요한 세 가지 형태의 요구는 다음과 같다.

- ① 어떤 상태로 전환해 들어가는 객체에 존재해야 하는 ‘입장조건’
- ② 상태에서부터 빠져 나오는 객체에 존재해야 하는 ‘퇴장조건’,
- ③ 객체가 어떤 상태에 있는 동안에 존재하는 ‘상태설명’.



Entry:	0도C 이상의 온도 액체상태	
Add		
Edit		
Remove		
State:	0도C에서 100도C사이의 온도 액체상태	Done
Add		Help
Edit		
Remove		
Exit:	100도C이상의 온도 액체상태	
Add		
Edit		
Remove		

그림 4-9 : OSTN 과 상태정의

■ OSTN의 기능

- ① 개체 중심의 관점에서 모델의 구축을 가능케 한다.
- ② 한 분야에서 개체의 전환이 가능한 상태들을 요약한다.
- ③ 자료의 생성에서 소멸까지를 문서화 한다.
- ④ 프로세스 흐름 다이어그램을 관통한다.
- ⑤ 개체의 동적 형태를 특성화 한다.

그림 4-10은 객체상태 전환 다이어그램의 예로서 이발소에서 머리를 깎는데 발생할 수 있는 상황을 고객의 상태 변환의 관점에서 기술한 것이다. 각 각의 상태가 전환되는 과정에서 어떠한 UOB가 관련 되고 있는가를 우리는 표현 할 수 있으며 이는 각 UOB의 기능을 명확히 하고 있다 .

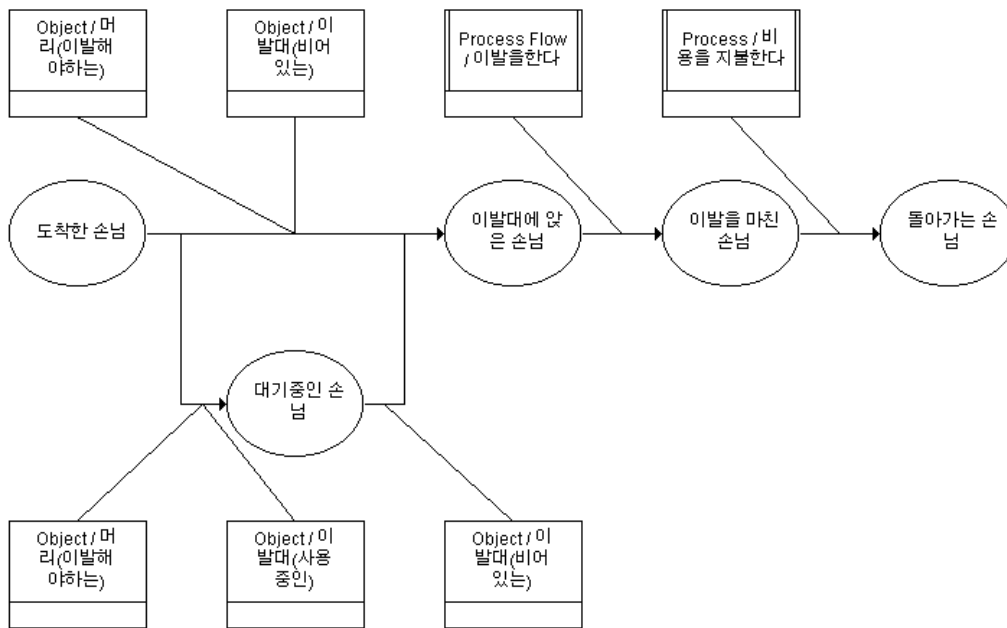


그림 4-10: OSTN 과 UOB 의 관계

### 4.2.3 프로세스의 분해

■ 분해의 목적

- ① 다이어그램의 복잡성을 감소시킨다.
- ② 다양하게 추상화 된 레벨에서의 묘사를 포착할 수 있도록 지원한다.
- ③ 같은 프로세스를 다른 지식의 원천 및 다른 관점에서 모델화 할 수 있는 능력을 제공한다.

IDEF $\emptyset$ 와 마찬가지로 IDEF3 프로세스 흐름 다이어그램은 다이어그램의 복잡성을 줄이고 다양하게 추상화된 레벨에서 다양한 관점의 설명을 포착할 수 있도록 지원하기 위하여 프로세스의 분해기법을 지원한다. 그런데 IDEF $\emptyset$ 가 최상위 레벨에서 하나의 활동만을 표시 표현하고 이를 경계조건으로 선언하는 것과 달리 IDEF3 프로세스 다이어그램은 최상위 레벨의 다이어그램에서부터 우리가 파악하고자 하는 영역내의 프로세스의 흐름을 여러 개의 프로세스간의 관계로 자유롭게 나타낼 수 있으며 이들 프로세스의 분해에 있어서도 또한 하나의 UOB에 대하여 한 개 이상의 분해 다이어그램을 허용하고 있다.

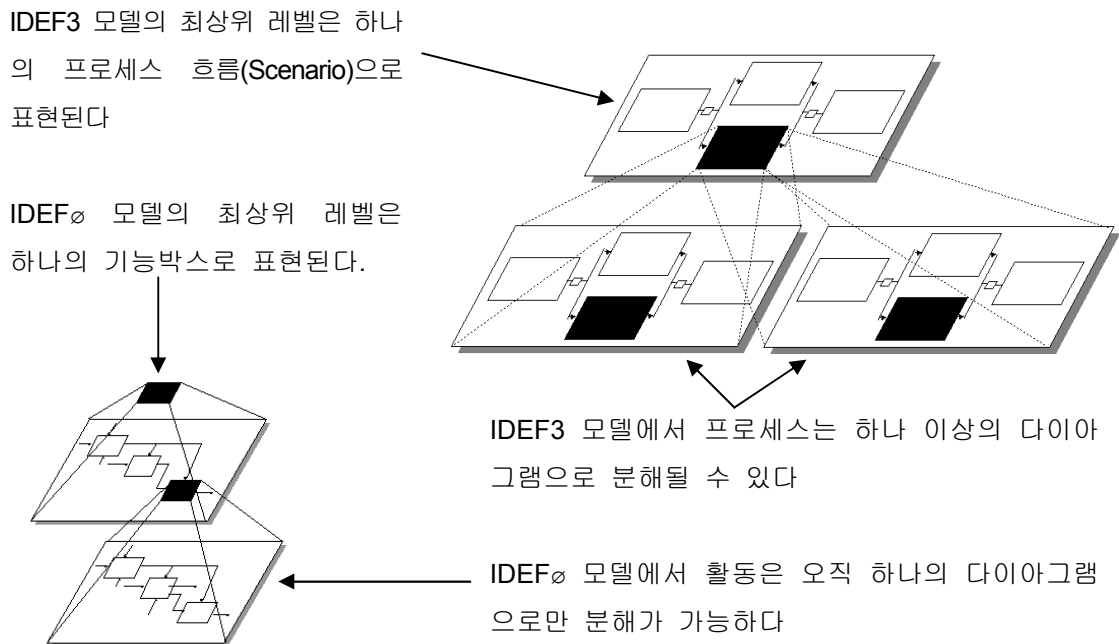


그림 4-11 : IDEF3 와 IDEF $\emptyset$ 의 분해 구조

이는 IDEF3의 기본적 개발 목표인 전문가의 설명 내용을 획득하여 표현함에 있어서 사실적, 객관적 관계의 확인 보다는 다양한 관점에서의 여러 전문가의 설명을 표현할 수 있도록 하기 위함이다. IDEF3에 있어서 분해는 추상화 된 단계의 설명을 포착하기 위한 중요한 방법 중의 하나로써 하나의 프로세스(UOB)에 관하여 다른 관점이나 지식 근거에서 모델을 구성할 수 있도록 다양한 방법을 지원한다. 문법적으로 IDEF3에 있어서 분해이란 하나의 프로세스를 구체화하는 세분화된 또 하나의 프로세스 흐름 다이어그램의 구축을 말한다.

여기서 우리는 왜 이러한 다양한 관점의 설명의 포착이 필요한지를 알아보자.

그림 4-12는 차량정비소의 업무를 분석하기 위하여 모델작업자와 관련된 담당자와의 인터뷰를 바탕으로 만들어진 세가지 관점의 인터뷰 내용 및 다이어그램이다. 우선 모델작업자가 숙련 정비공과 인터뷰한 내용과 이를 중심으로 표현된 자동차 정비 시나리오의 '정비를 수행한다' 프로세스 분해 다이어그램은 그림 4-12와 같다.

- ① 차량이 정비소에 들어오면 차량상태(엔진, 전기계통, 동력전달계통)를 점검한다.
- ② 문제가 되는 부분을 정비한다.
- ③ 차량이 정상적으로 수리되었는지 정비상태를 점검한다.
- ④ 정상적으로 수리가 종료되었으면 정비내역을 기재한다.

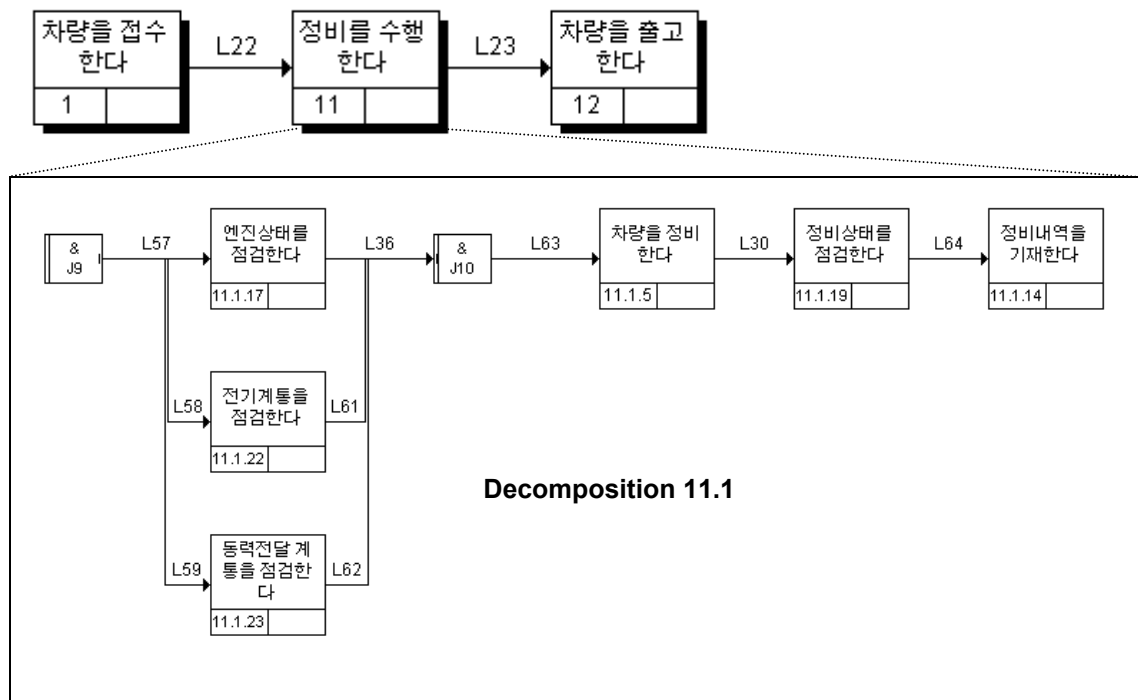


그림 4-12 : 분해의 예(숙련 정비공의 관점)

다음으로 모델작업자가 수습 정비공과 인터뷰한 내용과 이를 중심으로 표현된 ‘정비를 수행한다’ 프로세스의 분해 다이어그램은 그림 4-13 과 같다.

- ① 차량이 정비소에 들어와 접수가 되면 접수증을 확인한 후 자동차키를 손님으로부터 받아 차량을 수리장소에 주차 대기시킨다.
- ② 차량의 정비는 숙련 정비공이 수행하며 정비전이나 정비가 끝난 차량의 폐기물을 처리하고 차량을 청소한다.
- ③ 정비가 끝난 차량은 손님이 출고증을 가져오면 출고증을 확인한 후 자동차 키를 인도한다.

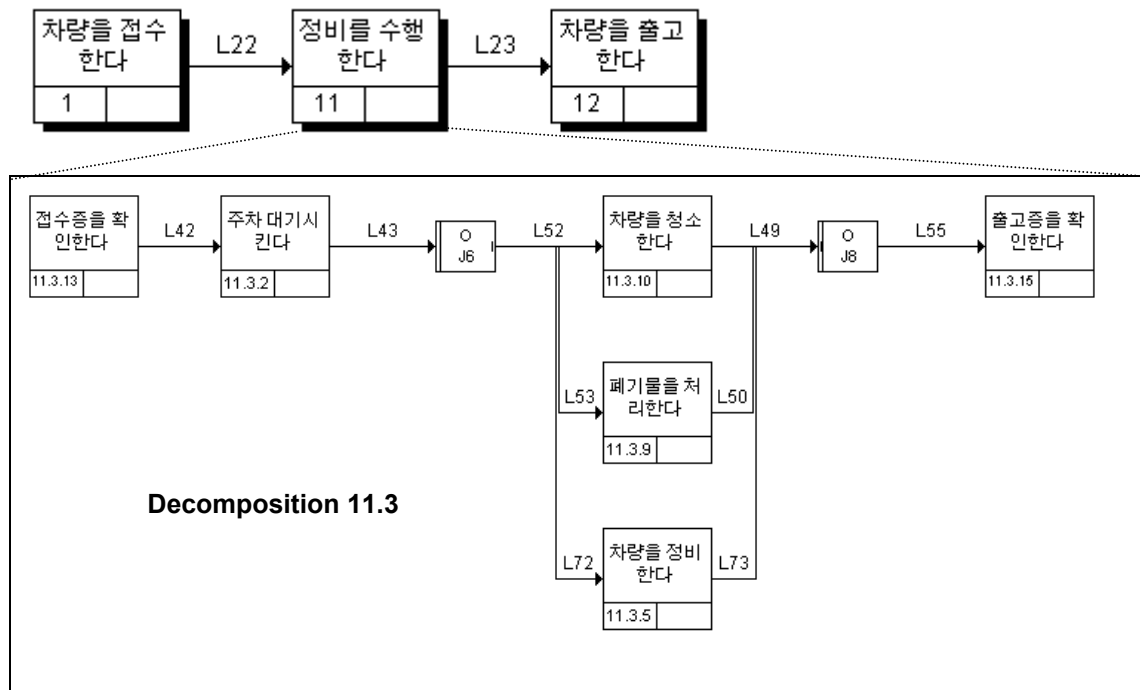


그림 4-13 : 분해의 예 (수습 정비공의 관점)

마지막으로 모델작업자가 관리자와 인터뷰한 내용과 이를 중심으로 표현된 ‘정비를 수행한다’ 프로세스의 분해 다이어그램은 그림 4-14 와 같다.

- ① 접수된 차량에 대해서는 우선 정비팀을 배정한다.
- ② 배정된 차량은 해당 정비팀에서 수리된다.
- ③ 수리내역이 전달되면 해당 내역을 전산등록한다.
- ④ 등록된 내용을 바탕으로 수리비 청구를 발행하여 청구한다.
- ⑤ 수리비용이 접수되면 해당 차량의 출고에 필요한 출고증을 발행한다.



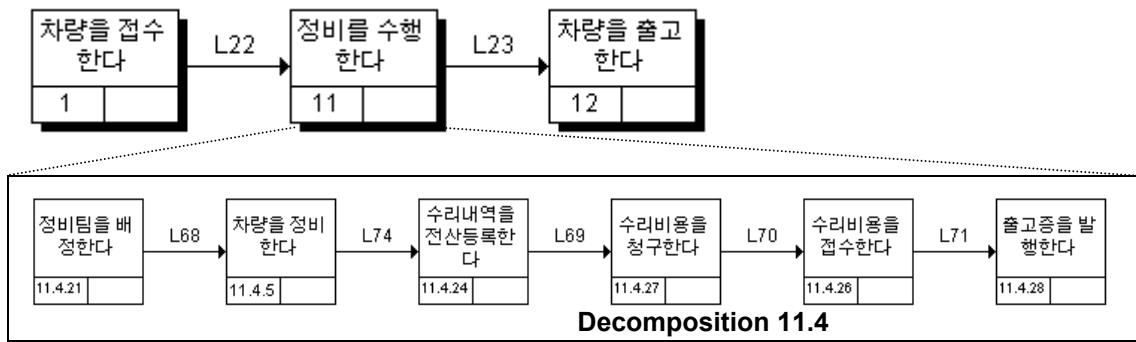


그림 4-14 : 분해의 예 (관리자의 관점)

우리는 위의 예에서 ‘정비를 수행한다’ 프로세스에 대한 설명이 인터뷰 대상에 따라 각기 다른 관점에서 다르게 표현될 수 있고 이를 바탕으로 작성된 분해 다이어그램은 결국 각기 다르게 표현될 수 밖에 없음을 알아보았다.

■ 분해의 종류

- ① **Objective view** : 다양한 관점의 분해는 아마도 개체중심의 관점으로 통합될 것이다. - **Objective view** 는 객관적인 관찰자의 관점을 수용한다. 따라서 하나의 **Objective view** 가 있을 뿐이다.
- ② **Role view** : 이해하는 바에 따라, 혹은 각자의 관점에 따라, 각각의 역할과 타입에 따라, 조직의 기능에 따른 프로세스의 관점에 따라 프로세스에는 하나 이상의 **role view** 가 있을 수 있다.

IDEF3 에 있어서 분해는 두 가지로 분류 될 수 있는데 하나는 객체 관점의 다이어그램이고 다른 하나는 프로세스를 수행하는 역할(Role)의 관점에서 기술된 다이어그램이다. 자연적(중립적) 관찰자의 입장에서 기술되는 객체의 상태 변환에 관한 객체 관점은 오직 하나의 하위 단계만을 수용 할 뿐이나 프로세스를 수행하는 역할의 관점에서 기술되는 프로세스 흐름 다이어그램은 하나의 프로세스를 여러 가지의 다이어그램으로 분해 할 수 있다. 즉 하나의 프로세스에 대하여 우리는 관찰자의 설명에 따라 다양한 관점의 지식을 표현 할 수 있는 융통성을 확보할 수 있는 것이다. 그림 4-15 는 차량정비소의 업무를 객체 중심적 관점에서 OSTN 으로 표현하고 이를 바탕으로 관리자와 숙련 정비공, 수습 정비공의 설명을 통합한 프로세스 모델을 작성한 예이다.

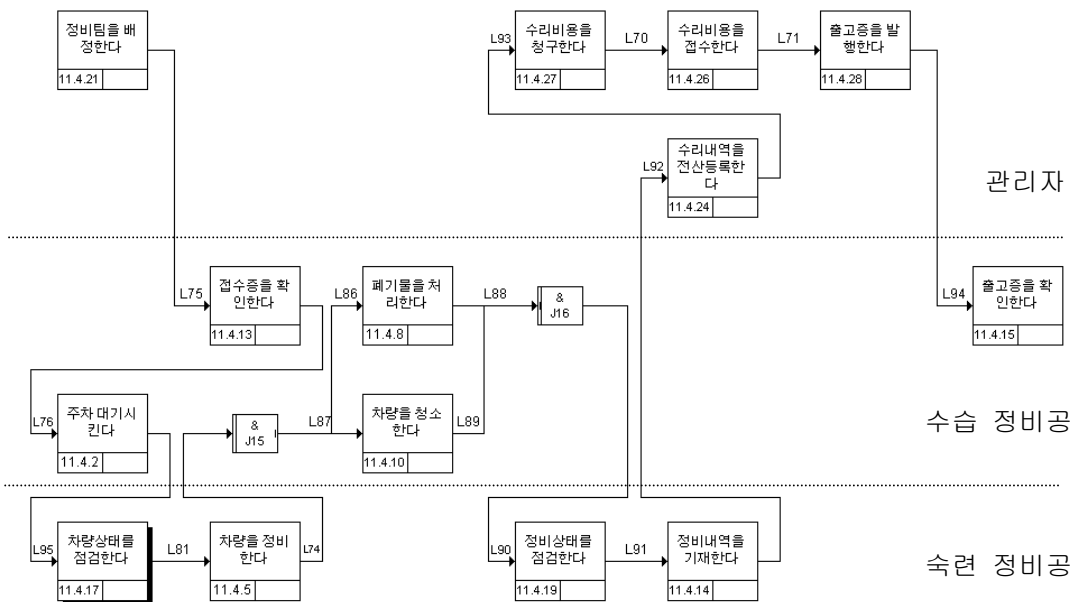
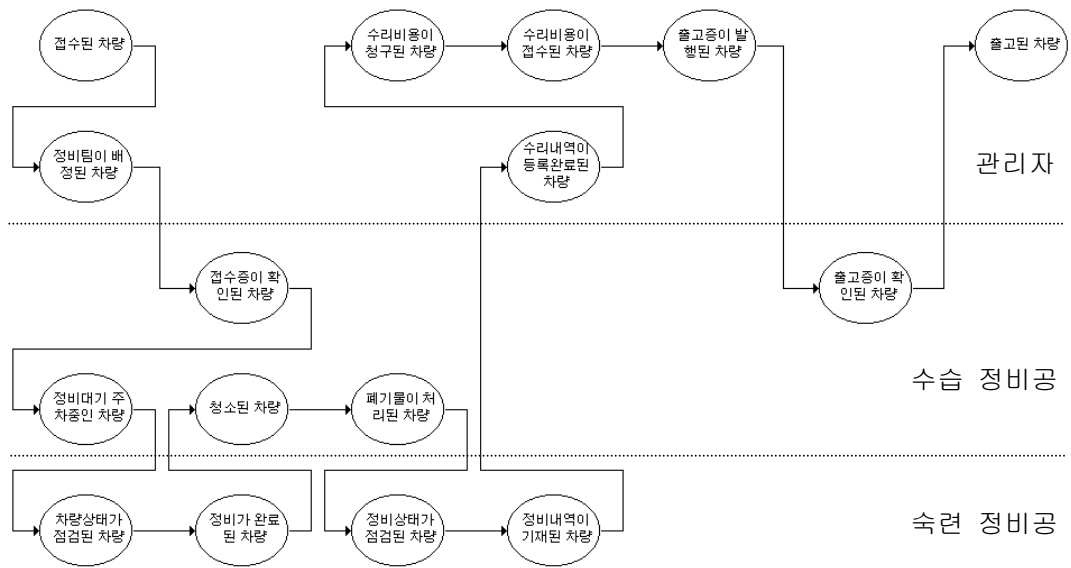


그림 4-15 : OSTN 과 프로세스 흐름 다이어그램

4.2.3 다이어그램 번호 부여

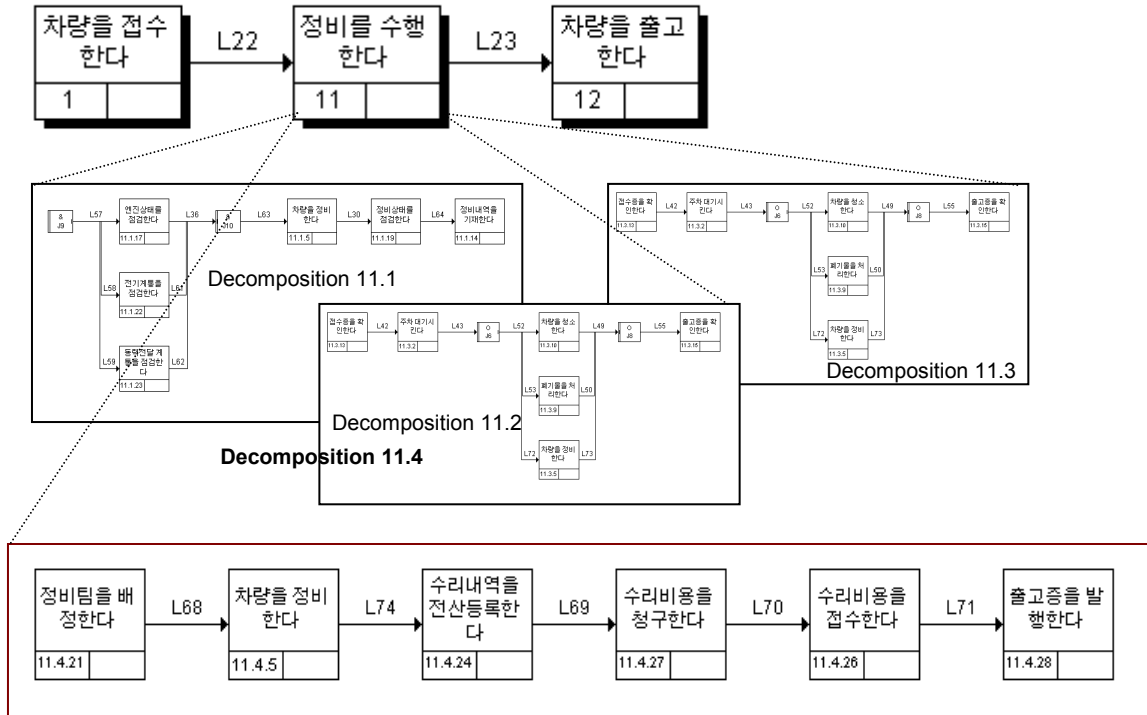


그림 4-16 : 다이어그램의 번호부여 체계

IDEF3 에서 UOB 번호는 프로세스의 분해 기법을 지원하기 위한 형식으로 표현되는데 그림 4-16 은 차량 정비업무 시나리오의 최상위 레벨에 표현된 세 개의 UOB 중 ‘정비를 수행한다’ 프로세스가 네 가지의 다이어그램으로 분해된 내용을 보여주고 있다. 최상위 레벨에서의 UOB 번호는 모델에 등록된 UOB 순서에 따라 1 부터 순차적으로 부여된다. (1, 2, 3,... 등으로). 분해 다이어그램 다이어그램에서의 번호는 두 가지 부분으로 구분될 수 있는데, 그림 4-17 에서 ‘정비를 수행한다’의 네 번째 분해다이어그램(Decomposition 11.4)에 표시된 첫번째 UOB 인 ‘정비팀을 배정한다’에 나타난 11.4.21 이라는 UOB 번호는 분해다이어그램의 번호를 나타내는 11.4(상위 레벨의 11 번 UOB 의 네 번째 분해 다이어그램)와 모델 전체에서 ‘정비팀을 배정한다’ UOB 가 등록된 유일한 순차적 번호인 21 이 결합한 상태를 보여주고 있다

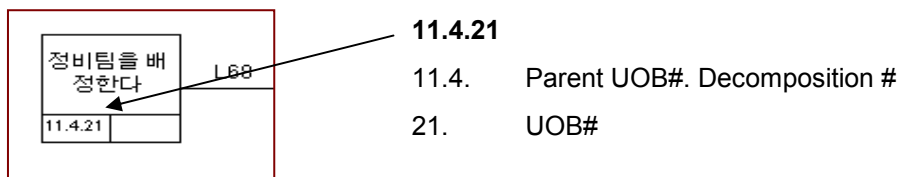


그림 4-17 : UOB 의 번호구조

그 밖에 IDEF3 다이어그램에서 표현되는 Link 의 경우는 L1, L2, L3 과 같이 링크를 나타내

는 L 과 등록된 순차적 번호인 1, 2, 3 등으로 결합되어 표시되며, 접속의 경우도 J1, J2, J3 과 같이 접속을 나타내는 J 와 등록된 순서를 나타내는 1, 2, 3 등이 결합되어 표현된다.

IDEF3 다이어그램은 종종 IDEF0 다이어그램과 같이 상호 보완적으로 이용되는데 그림 4-18 의 UOB 1 우측 하단에 표현된 Reference 번호는 이미 만들어진 기능모델을 IDEF3 프로세스 모델로 변환할 때, IDEF0의 참조된 Activity 번호를 표현하기 위한 항목으로 해당 UOB 가 IDEF0 다이어그램의 어떤 활동(Function)으로부터 전환된 것인지를 나타낸다.

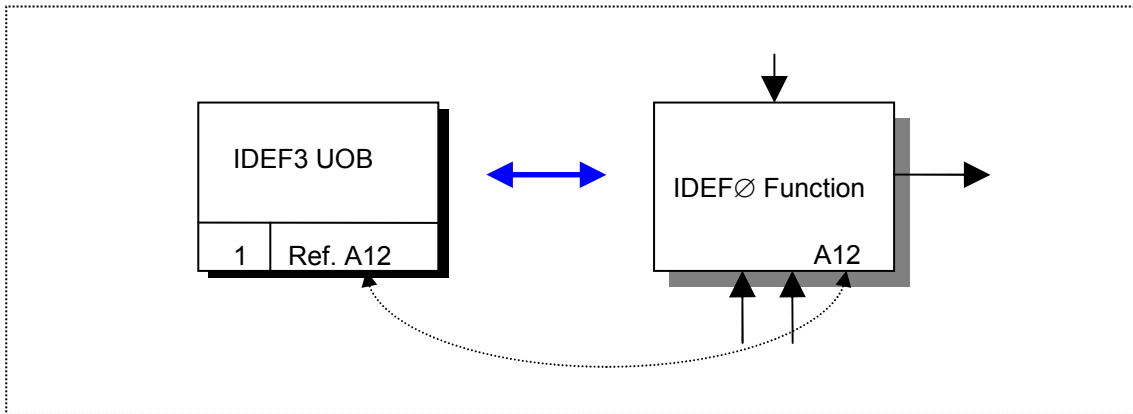


그림 4-18 : UOB 의 Reference

## 4.3 IDEF3 모델링 지침

### 4.3.1 IDEF3 모델의 검토

#### ■ IDEF3 모델의 판독

- ① 모델의 범위를 이해하기 위하여 배경, 목적, 관점을 연구한다.
- ② 프로세스 흐름 다이어그램은 왼쪽에서 오른쪽으로 맨 왼쪽에 있는 프로세스에서부터 시작한다. 이러한 다이어그램 판독 방법을 “**Workthrough**의 수행”이라고 한다.
- ③ 각 엘리먼트의 설명과 상세화(**elaboration**)를 검토한다.

#### ■ IDEF3 모델 작성의 힌트

- ① **XOR fan-out** 접속 다음에 **AND fan-in** 접속이 와서는 안 된다.
- ② 다이어그램의 맨 왼쪽에 여러 개의 프로세스가 있으면 안 된다. 이들은 해석에 혼란을 주므로, 프로세스의 흐름을 명쾌하게 하기 위하여 여러 개의 프로세스 전에 **fan-out** 접속을 쓴다.
- ③ 명료한 다이어그램을 위해 한 곳에서 분기되는 **fan-out** 접속을 피한다.
- ④ **Fan-out** 접속이 **fan-in** 접속으로 바로 이어지는 것은 다이어그램에서 생략된 프로세스가 있다는 것을 나타낸다.

### 4.3.2 프로세스 설계의 8가지 원칙

■ 원칙 1. 프로세스 설계는 디자인 활동이다.

- ① 당연히 우선 창조적 이어야 한다. 여러분은 최선의 수행방법을 적용하고, 찾고, 따라 하여야 한다.
- ② 수행에 있어서는 우선 반복적이어야 한다.
- ③ 비용/수행성/이익/위험회피 등이 요구되며, 특히 시뮬레이션과 ABC 분석에 있어서 그렇다.
- ④ 하나의 해결책만을 제시해서는 안 된다.
- ⑤ 세부적인 사항들이 작성될 때 까지는 완료되지 않아야 한다.

■ 원칙 2. 프로세스 디자인 전문기술은 여러 가지 기술의 조합과 이들을 어떻게 적절히 구사하느냐 하는 것으로 이루어진다.

- ① 프로세스 디자인은 당신이 제약사항을 조정하고 만족을 성취하는 것을 도와준다.
- ② 요구사항과 디자인 목표는 다르다.
- ③ 프로세스 모델은 플로우 차트가 아니다.
- ④ 유행에 따라서 발전할 필요는 없다.
- ⑤ 모델링은 트레이드오프 분석을 위한 주제로서 여러 가지 대안을 제시해야 한다.

■ 원칙 3. “Object design”은 프로세스 디자인에서 중심적인 역할을 제시한다.

- ① 입력과 출력들
- ② 자원
- ③ 중간에 존재하는 개체
- ④ 인터페이스 개체
- ⑤ 개체 상태 변이
- ⑥ 개체의 “품질” 측정

■ 원칙 4. 프로세스는 실행되는 환경에서 특정 자원의 분배가 될 수 있도록 지원하는 하나의 레벨에서 표현되어야 한다.

- ① 하위 프로세스로 분해한다.
- ② 프로세스 디자인의 종단 조건을 사용한다.
- ③ 기술의 변화나 수용능력의 변화등과 같이 증가되는 프로세스

■ 원칙 5. 물리적 그리고 논리적 입력/출력들은 계속적으로 관리되어야 한다(보호법칙).

- ① 각 프로세스 단위의 특정 입력/출력 그리고 프로세스 흐름에서 어떤 프로세스의 위치에서 요구되는 출력 그리고 일치되는 입력 가능한 것들.
- ② 분해를 계속한다.
- ③ 개체 설계와 높은 의존을 보이는 것들.

■ 원칙 6. 당신이 검토해야 할 오류는 항상 있는 법이다.

- ① 모델 정의의 오류
- ② 모델 분석의 오류
- ③ 하위 프로세스 디자인 검출의 오류
- ④ 하위 프로세스 디자인 작업의 오류
- ⑤ 오류에 관련된 인내.

■ 원칙 7. 프로세스 디자인은 제품 관리에 의한 프로세스 단계들의 디자인을 포함한다.

- ① 쓰레기나 못쓰게 된 것들
- ② 정의
- ③ 수집
- ④ 배치

■ 원칙 8. 프로세스 디자인은 협력의 수행과 관리를 위한 프로세스 스텝의 디자인을 포함한다.

- ① 동시공학 적 프로세스
- ② 자원 배분
- ③ 작업 아이템 우선순위
- ④ 상태, 수행, 추적, 자료수집
- ⑤ 의사소통 관리

■ 결론

IDEF3 은 시스템 작동 방식에 관하여 유력한 전문가가 알려 준 조정되지 않은 설명을 파악하는 매체를 제공하도록 디자인되었다. 그러나 이것은 반대로 객관적 사실을 조직.전달하기

위한 풍부한 메커니즘을 제공하지는 못한다. 많은 시나리오와 관점을 하나의 다이어그램으로 결합하는 데서, 지나치게 복잡한 IDEF3 다이어그램이 나올 수 있다. 그러나 다이어그램은 다음과 같은 간단한 발견에 의해 단순화 될 수 있다. 시나리오와 관련된 모든 장면으로부터 어떤 UOB 가 보이지 않을 경우, 그것은 분리된 UOB 일 것이다. 또한 수집된 데이터의 '간격을 메우려는' 경향을 조심해야 한다. 설명 파악 방법론으로서 IDEF3 은 '부분적인' 심지어는 모순된 설명에 관하여 관대하도록 디자인 되었다. 그것들을 지원하는 정보 시스템과 조직 분석 가운데, 특별한 문제의 근거가 되는 것은 바로 이러한 모순된 혹은 불완전한 부분이다. 생략에 의해 혹은 의도적으로 이들 영역을 설명에서 제외 한다면, 문제는 무시될 수 있다. 그러나 이는 '어떤 것도 가리키지 않는 이들 모델은 잘못되었다' 는 비난을 유발하게 될 것이다. 시스템의 문제점을 발견하지 못하는 것은, IDEF3 모델화 방법 자체의 부족에서 나오는 것이 아니고 모델 작업자의 개방적 관대 함에서 나오는 것이다.