

인간 및 기계 시스템 설계를 위한 업무분석의 객체지향적 업무기술방법에 관한 연구*

김 선 욱

단국대학교 산업공학과 교수

Object-Oriented Task Description in Task Analysis for Designing Man-Machine System

Sun-Uk Kim

Department of Industrial Engineering, Dankook University

This research deals with an object-oriented task description(OOTD) method to solve problems such as being hard to understand and a lack of description in existing task description methods. The OOTD method includes an object-oriented task structure diagram and an object-oriented task attribute diagram. The former shows some relationship between a task and a derived task. The latter includes important attributes to define each task. Finally, a sample case is introduced to show how efficient these two diagrams are for the task description.

Keywords : object-oriented task description, task structure and attribute diagram

1. 서 론

인간 및 기계시스템을 설계하고 구현하기 위해서 업무분석은 필수 불가결한 과정이다. 업무분석은 시스템의 목적을 달성하기 위하여 작업자가 반드시 하여야 하는 것을 활동이나 프로세스에 의하여 규명하는 것으로 정의할 수 있다. 그 동안 시스템에서 인간과 기계의 상호작용을 기술하고 분석을 지원할 수 있는 많은 방법들이 연구되어 왔다[5,6,7,8]. 이들 연구들의 대부분은 분석가나 인간 공학자들에게 자료수집, 정보조직화 및 이용을 효과적으로 지원하여 다양한 설계결정이 이루어지도록 지원하는 것에 주안점을 두고 있다.

업무자료수집과정은 인간과 시스템 상호작용에 대한 자료를 수집하는 것으로 Critical incident technique, Observation, Structured interview, Verbal protocol 등 다양한 방법이 개발되어 있다[5]. 이렇게 도출된 다양한 업무들의 생산적인 설계와 구현과정을 지원하기 위해서는 체

계적으로 기술되고 구조화 또는 조직화되어야 한다. 지금까지 이용되고 있는 체계적인 방법들은 대부분 차트나 네트워크 등이 주로 이용되어 왔다. 그러나 이 방법들은 서술화 정도가 미약할 뿐만 아니라 시스템이 복잡하고 커지면 사용자의 이해도를 크게 감소시킬 수 있다. 더욱이 너무나 많은 그 방법들의 원리와 사용방법들이 이해하기 어렵다는 단점이 있다.

또한 업무분석은 일회적 과정이라기 보다는 반복적 또는 진화적으로 이루어진다. 실제 인간 및 기계시스템을 구축할 때 시스템의 초기개념 생성단계부터 구현단계에 이르기까지 업무분석이 반복된다. 따라서 업무기술을 위한 유지 및 보수 노력을 최소화하는 문제는 생산성과 직결된다고 볼 수 있다. 본 연구에서는 언급된 업무기술 방법론의 문제점을 극복하기 위하여 소프트웨어분야의 객체지향방법론을 인간 및 기계 인터페이스 구현을 위한 업무서술과 구조화하는 과정에 접목하고자 한다.

* 이 연구는 2003학년도 단국대학교 대학 연구비의 지원으로 연구되었음.

2. 기존의 업무기술 방법론

업무기술을 위한 주요한 방법들은 Charting and network technique, Decomposition methods, Hierarchical task analysis(HTA), Link analysis, Operational sequence diagram(OSD) 등이 있다[5]. Charting and network technique[5,9]은 각종 flow chart들로 이해하기 용이할 뿐만 아니라 운영절차를 직접 만들 수 있는 양식을 제공하기도 한다. 이러한 이유 때문에 업무성격이 순차적이고 중·단기의 업무에는 지금도 널리 이용되고 있다. 그러나 정신적인 작업내용이나 복잡한 업무들이 관련될 때는 그 효과는 크게 떨어지는 단점이 있다.

Decomposition은 업무기술을 보다 상세한 일련의 서술문으로 만드는 구조화된 방법이다[6]. 이 방법을 창안한 Miller[6]는 Description, Controls used, Decisions, Typical errors, Response, Feedback 등과 같은 부제를 사용하였다. 그러나 이러한 부제는 인터페이스를 설계하는데 적절하지 못할 뿐만 아니라 필요한 정보수집과 수집된 정보를 부제로 구분하는데 소요되는 상당한 노력과 비용을 수반하게 한다. 더욱이 업무를 식별하는 방법이 자체적으로 지원되지 못하고 있으며, 업무의 상호관계를 파악하고 추적하기가 용이하지 못한 편이다.

HTA는 정신적인 작업을 포함한 광범위한 범위의 업무를 표현하는 일반적인 방법으로 개발되었다[1]. 이 HTA는 작업과 계획에 의하여 업무기술서를 개발하는 과정이다. HTA 분석결과는 표나 계층도로 표현되나 상호관계를 보여주는 업무가 대단히 많은 경우 오히려 분석가에게 혼란을 야기시킬 수 있다. 또한 HTA는 다수와 협동으로 작업이 이루어지는 것이 바람직할 때에도 많은 시간과 노력이 수반되는 단점이 있다. 더욱이 분석가가 업무를 효율적으로 분석하기 위하여 상당한 기술을 습득해야 할뿐만 아니라 업무자체에 대한 서술이 너무 미약한 약점이 있다.

또한 작업이나 시스템 일부간의 관계를 표시해주는 Link analysis[8], OSD[8] 방법 등이 있다. 전자의 방법은 Link를 도출하기 위해서는 엄청난 노력이 요구되며, Link 표나 도표의 특성상 비교적 간단한 시스템만이 이용된다. 후자의 방법은 작업들간의 관계를 표현하는데 대단히 유용하나, 복잡하거나 정신적인 작업에 대하여 적용하기에는 거의 불가능한 한계를 갖는다.

3. 객체지향업무기술 방법론

3.1 객체지향 기법

객체지향기술의 주요한 개념은 객체, 메시지, 클래스, 상위클래스와 하위클래스로 구성된다. 객체는 객체지향기술의 기초가 되는 것으로 어떤 행태를 보여주는 유형의 실체[3,4]로 정의된다. 메시지는 객체들 간의 영역과 객체들 간의 정보 전달을 지원해주며, 유사한 활동을 하는 동일한 성격의 객체 집합을 클래스라 한다.

실제로 객체들은 부모의 클래스가 생성을 요청하는 메시지를 받았을 때 생성된다. 이때, 생성된 객체는 속성과 메시지를 그 부모가 되는 클래스로부터 상속받는다. 이와 같은 계층구조에서 주어진 클래스보다 상위의 계층의 클래스들을 우리는 상위클래스 하위계층의 클래스를 하위클래스라고 한다.

객체지향기술의 주요한 특성은 데이터 추상화, 캡슐화, 상속성과 다형성으로 구성된다. 일반적으로 추상화는 어떤 현상이나 사물에 대해 중요한 부분만 취함으로써 그 현상이나 사물을 규정짓는 행위로 정의된다. 캡슐화는 하나의 물리적 또는 개념적인 캡슐안에 수많은 관련된 것들을 감싸주는 기능이다. 상속의 개념을 객체지향기술에 적용한 것으로 동일한 특성을 가진 객체들을 클래스로 정의하고 그 클래스의 특성뿐만 아니라 더 독특한 특성을 가진 객체들을 각각의 하위클래스로 정의한다. 마지막으로 다형성은 동일한 메시지에도 수신측 객체에 따라 다른 결과를 가져오는 특성을 말한다.

3.2 객체지향업무기술(OOTD)모형

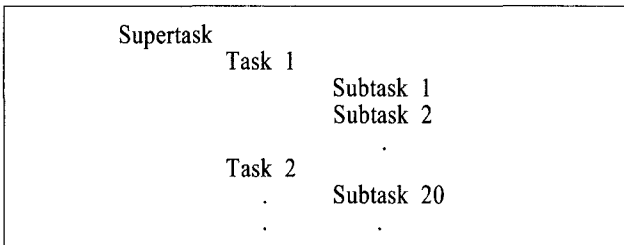
3.2.1 객체지향 업무구조도

우선 OOTD모형은 객체지향기술에서 제공하는 대부분의 특성을 갖는다. 이 모형은 구체적인 텍스트를 추상화하여 객체와 속성으로 표현되어 추상화를 실현하고 있으며, 업무간의 상하위 관계를 통하여 상위클래스의 특성을 전수하는 상속성을 포함하고 있다. 또한 업무명으로 충분히 상호토의가 가능하며, 필요할 경우 업무의 속성들을 살펴보면 상세하게 알 수 있게 함으로써 캡슐화를 구현하였다.

객체지향 업무구조도는 객체로 정의된 업무간의 상호관계를 보여주는 도표로 정의한다. 업무간의 상호관계는 객체지향 개념을 이용하여 업무의 모임을 의미하는 클래스와 클래스간 상하위 관계로 표시된다. 업무구조도는 전체업무의 관련도를 한눈에 볼 수 있도록 지원함으로써 구축하고자 하는 시스템의 전반적인 이해도를 높이기 위하여도 이용된다. 또한 하위 업무는 상위업무에서

정의된 속성들을 자동으로 갖게되며 다르게 정의된 업무에 대해서만 별도로 정의하는 상속성의 원리를 실현하고 있다. 이러한 특성은 복잡한 업무의 기술을 크게 간소화할 수 있을 뿐만 아니라 업무의 이해도를 크게 증진시킬 수 있다.

<그림 1>에서 제시된 각 업무는 클래스와 유사한 개념으로 상하위간의 관계를 보여주며, 각 업무의 복잡한 내용들을 캡슐화하고 있다. 하위업무는 추가 하위 업무로 분할되는 과정이 반복되어 마지막으로 leaf 업무까지 이르게 된다. 마지막 단계의 기준은 정확한 원칙은 없지만 통상적으로 계속 분할한 후 한 작업자가 최소의 혼란으로 그 업무설명서를 읽고 처리할 수 있는 수준으로 결정한다[2].



<그림 1> 일반 객체지향 업무구조도

3.2.2 객체지향 업무속성도

업무중심 사고를 객체지향중심 사고로 전환함으로써 자연스럽게 객체, 즉 업무에 영향을 미치는 속성들이 도출되었다. 본고에서는 이들 속성을 <표 1>처럼 Supertask와 Subtask를 포함한 13개의 속성으로 정의하였다.

<표 1> 객체지향 업무 속성도표

속 성	설 명
Supertask	해당 업무를 포함하는 상위 업무
Initiating event	업무를 시동하게 하는 이벤트
Terminating event	업무를 완료하게 하는 이벤트
Goal	업무가 달성할 목표
Performance measure	목표 달성여부를 평가하기 위한 지표
Time required	업무를 완료하기 위한 소요시간
Resources	업무를 완료하기 위해 필요한 시스템 또는 부시스템
Inputs	업무에 제공되는 입력
Outputs	업무가 제공하는 출력
Procedure/ relationship	입력에서 출력을 만드는 절차
Possible failures	발생할 수 있는 예러나 고장
Subtasks	분할되어지는 하위 업무
Human operator	업무를 완료하기 위한 필요 작업자

Supertask와 Subtask는 상속성의 원리를 구현하기 위하여 도입되었으며, 업무의 목표와 평가기준을 규정하는 Goal과 Performance measure의 속성이 도입되었다. 설계자가 설계하여야 할 대상들은 Resources에서 정의되고, 작업자가 Inputs, possible failures를 염두해 두면서 Outputs에서 제시된 내용을 수행한다.

Subtask와 Supertask는 객체지향기술의 상속성을 구현한 것으로 업무분석작업을 크게 경감할 수 있다. 예를들어 하위업무의 한 속성이 상위업무의 속성과 동일한 경우 blank나 ()를 사용하여 쉽게 속성의 기술이 중복임을 보여줄 수 있다. 또한 업무가 더 이상의 하위업무를 갖지 않는 경우 Subtask의 속성치에 none값을 기술하여 그 업무가 leaf 업무임을 보여줄 수 있다. 동일 업무 내에서도 중복될 경우 see goal과 같은 지시어를 사용함으로써 업무기술량을 크게 줄일 수 있다.

OOTD모형은 Decomposition methods의 후진방식과 달리 속성에 의하여 기동되는 전진방식을 취함으로써 시간과 비용을 크게 감소시킬 수 있다. 많은 훈련과 경험이 요구되는 HTA는 단순히 업무의 구조와 절차만을 제시하고 있으며 업무자체에 대한 기술은 대단히 취약하다. 반면에 OOTD모형은 Push방식으로 업무의 속성을 채우는 것을 요구하므로 특별한 훈련이나 교육이 없어도 용이하게 업무기술을 작성할 수 있다. 대부분의 설계 평가 방법 중 모의실험방법은 동적인 모델로 사전에 구현될 수 있어 큰 장점을 갖는다[7]. 모의시스템으로 설계를 평가할 때에 객체지향기술을 감안하면 업무기술서와의 일관성이 대단히 높다.

4. 객체지향업무기술모형 적용

4.1 사 례

본 방법론의 유용성을 보이기 위해 두 개의 엔진으로 구성된 수송기의 연료관리에 필요한 연료관리시스템 인터페이스를 설계하는 사례를 이용한다. 이 연료시스템은 중앙과 좌우에 총 3개의 주요 연료 탱크를 가지고 있다. 좌우 주탱크는 좌우 날개 있으며 그 용량은 각각 85,000 파운드를 적재할 수 있다. 반면에 중앙의 연료탱크는 동체내의 중앙부분에 위치하고 있으며 그 용량은 100,000 파운드를 적재할 수 있다.

각 탱크는 Fuel Boost Pump와 Fuel Manifold Valve로 구성되어 있으며, 각 탱크간에는 연료탱크에서 양쪽 엔진에 연료를 제공하기 위한 공급선인 Fuel Manifold가

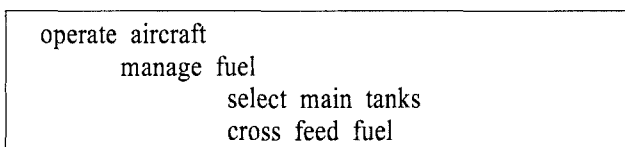
있다. 각 탱크는 한 개의 Fuel Boost Pump를 보유하고 있으며, 탱크에서 Fuel Manifold로 연료를 펌프하는 역할을 한다. Fuel Boost Pump는 on 또는 off 둘 중 하나의 상태를 갖는다. 각 Fuel Boost Pump와 Fuel Manifold 사이에는 Fuel Manifold Valve가 존재한다. 이 밸브는 펌프가 off 되었을 때 연료가 탱크로 역류되는 것을 막아주는 것이 목적이다.

또한 연료탱크에는 정상적인 주유 공급시스템이 원활하게 작동하지 않을 경우 이것을 지원하기 위한 일종의 보조 시스템이 존재한다. 이 시스템은 Fuel Cross Feed, Cross Feed Pump와 Cross Feed Valve로 구성된다. Fuel Cross Feed는 Fuel Manifold와 유사한 공급선이나 엔진에 연료를 공급하는 대신 단지 탱크간 연료를 이동하게 하는 역할을 한다. 각 탱크는 한 개의 Cross Feed Pump를 갖고 있으며 다른 탱크로 연료를 펌프하기 위해서 이용된다. 이 펌프는 on 또는 off 둘 중 하나의 상태를 갖는다. 각 Cross Feed Pump와 Fuel Cross Feed 사이에는 Cross Feed Valve가 있다. 이 밸브의 역할은 펌프의 상태 정보에 따라 달라진다. 만약 해당 Cross Feed Pump가 on일 때 열린 밸브라면 밸브를 통하여 연료가 해당 탱크에서 빠져나간다. 그러나 해당 Cross Feed Pump가 off이고 다른 탱크의 Cross Feed Pump가 on일 때 열린 밸브라면 밸브를 통하여 연료가 해당 탱크로 들어오게 된다.

연료관리는 다수 펌프의 On과 Off를 포함하여 밸브의 개폐 등으로 구성된다. 연료는 초기에 중앙 탱크에서 공급을 시작하며, 연료가 모두 소모되면 다음은 주 탱크에서 공급이 이루어진다. 필요하다면 Cross Feed System을 사용해서 어떠한 상황에서도 과일렛은 주 탱크들에 들어있는 연료량의 상호 균형을 맞추어야 한다.

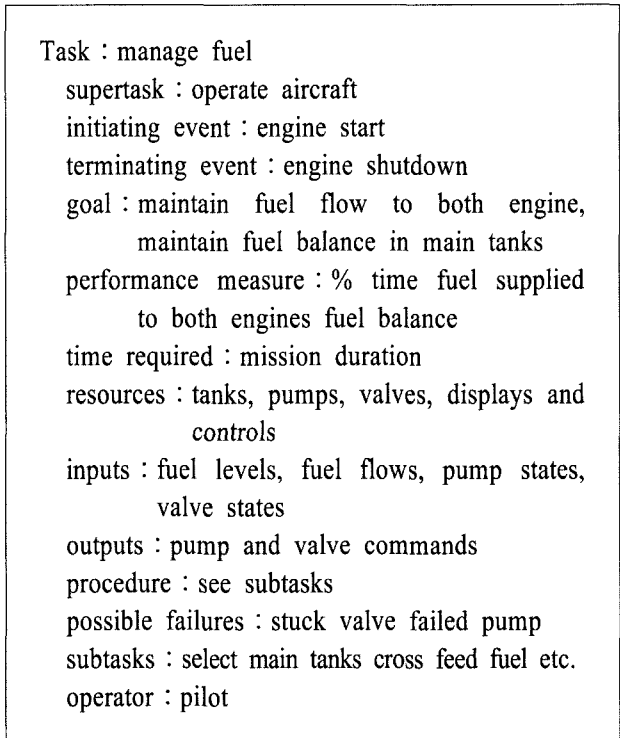
4.2 객체지향 업무구조도와 업무속성도

본 사례에 기초한 업무구조도는 <그림2>와 같다. 가장 상위의 업무는 operate aircraft로 표시되며, 그 하위 업무는 manage fuel로 정의된다. manage fuel을 수행하기 위해서는 물론 주어진 사례보다 더욱 많은 업무가 존재하지만 예시 목적상 두 개의 하위업무인 select main tanks와 cross feed fuel로 구성된다.



<그림 2> 객체지향 업무구조도 예

우선적으로 이 manage fuel 업무에 대한 업무속성도가 <그림 3>에 예시되고 있다. 이 업무에 대한 상위업무 operate aircraft는 supertask 속성치에 제시되어 있으며, 2개 목표는 goal의 속성치로 명확하게 기술되어 있다. 또한 resources를 보면 인간 및 기계 인터페이스를 구현할 때 tank, pump, valve, display와 control 등의 객체를 적절하게 설계해야 함을 알려주고 있다. 업무속성도를 보면 작업자인 pilot가 하여야 하는 업무는 input, possible failures를 감시하면서 output, 즉 펌프와 밸브에 대한 조작을 정확하게 수행하여야 함을 알 수 있다.



<그림 3> manage fuel 업무속성도

이들 두 업무에 대한 업무속성도가 각각 <그림 4>와 <그림 5>에 제시되고 있다. 이들 두 업무는 서로 다른 업무로서 각각 서로 다른 명확한 2개의 goal과 업무 소요시간을 갖는다. 각각의 업무가 시작되는 상황을 알려주는 것은 initiating event로 속성치 값이 사실로 나타날 때 업무가 시작된다.

예를들면, 중앙탱크 잔존 연료량이 5%에 이를 때 <그림 4>의 업무(Task : select main tanks)가 시작된다. 반면에 <그림 5>의 업무(Task : cross feed fuel)는 Boost Pump가 고장이 나면 시작된다. 전자의 업무는 procedure에서 기술된 것처럼 5단계로 수행되는 반면 후자의 업무는 4단계로 실시되는 것을 보여준다.

Task : select main tanks
 supertask : manage fuel
 initiating event : 5% fuel remaining in center tank
 terminating event : fuel flowing from main tanks and center tank empty and center boost pump off
 goal : fuel flowing from main tanks and center boost pump off
 performance measure : time required to complete successfully
 time required : 1 - 5 min.
 resources : ()
 inputs : ()
 outputs : ()
 procedure : turn both main boost pumps on, open both main fuel manifold valves, wait for center tank to empty, turn center boost pump off, close center fuel manifold valve
 possible failures : center boost pump off before main pumps on etc.
 subtasks : none
 operator : ()

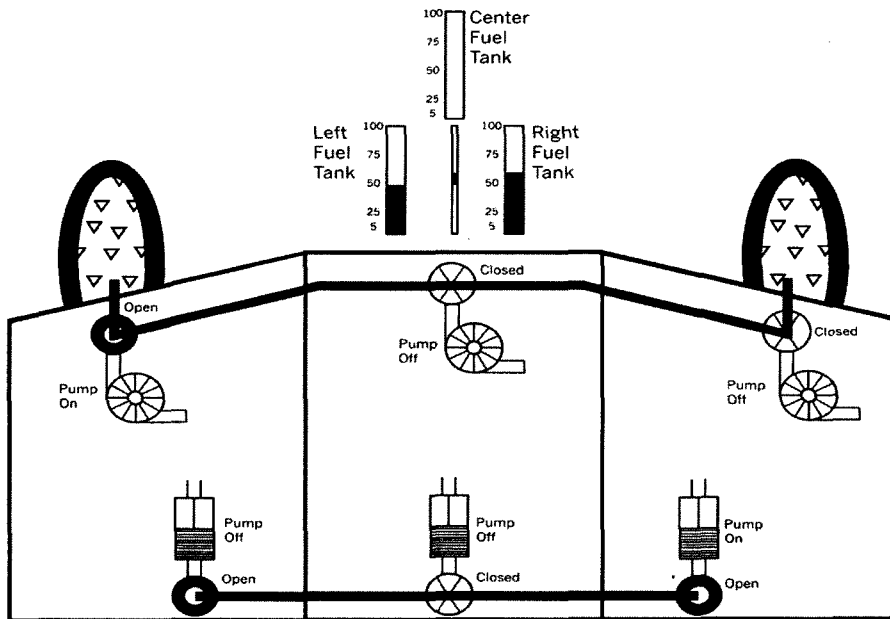
<그림 4> select main tanks 업무속성도

Task : cross feed fuel
 supertask : manage fuel
 initiating event : boost pump failure
 terminating event : see goal
 goal : fuel flow from failed tank to fuel manifold via cross feed
 performance measure : time required to complete successfully
 time required : 0.5 - 3 min.
 resources : ()
 inputs : ()
 outputs : ()
 procedure : turn failed pump off, close failed tank manifold valve, turn failed tank cross feed pump on, open alternate tank cross feed valve
 possible failures : alternate cross feed valve open before pumps on, etc.
 subtasks : none
 operator : ()

<그림 5> cross feed fuel 업무속성도

4.3 객체지향업무기술모형에 기초한 시스템 인터페이스 설계

인간 및 기계 인터페이스를 설계할 때 OOTD 모형은 주 객체인 resource, 업무과정을 보여주는 procedures, 디스플레이



<그림 6> 인간 및 기계시스템 인터페이스 설계 예

레이에 의거한 조작을 알려주는 outputs, 정신적인 작업인 inputs, possible failures 등 초점을 집중화하여 설계를 용이하게 하도록 도와준다. manage fuel 업무를 수행하기 위해서는 그 하위 업무인 select main tanks 업무와 cross feed fuel 업무를 수행해야 한다. 전자의 업무와 관련하여 정의된 속성 resource에서는 pump와 valve와 같은 control과 tank와 관련된 display를 이용해야 함을 보여준다. <그림 6>에 인간공학의 원리를 이용하여 control과 display가 상단부에 설계되어 있다. 상단부의 4개의 직사각형 중 3개는 탱크의 유량을 보여주고 있으며 제일 가느다란 사각형은 좌우 탱크 유량의 차이를 보여주는 display이다.

또 다른 하위업무인 cross feed fuel 업무는 initiating event 속성에서 규정된 것처럼 boost pump failure가 발생했을 때 시작한다. 이 업무에서도 마찬가지로 resource에서 정의된 control과 display에 인간공학의 원리를 적용하여 <그림 6>의 하단부와 같이 설계되었다.

작업자는 속성 inputs에서 정의된 것들을 display상에서 정보를 입력받아 속성 output에서 지정한 것처럼 pump와 valve에 대한 command인 on 또는 off 결정을 내리게 된다. 마지막으로 resources, inputs, output 등이 모두 정의되었다면 작업자는 possible failure를 염두에 두면서 procedure를 단계별로 점검함으로써 시스템설계의 충실성을 높일 수 있다.

5. 결 론

본 연구에서 제안한 OOTD 방법론은 객체지향 업무구조도와 객체지향 업무속성도표를 산출물로 제시한다. 전자는 업무간의 상호관계를 한눈에 볼 수 있도록 함으로써 전체시스템을 쉽게 이해하도록 지원하는 역할을 한다. 각 업무의 상세한 내용은 후자에서 완전하게 정의된다. 유도된 산출물을 이용하여 체계적이고 효과적으로 업무를 기술함으로써 업무에 대한 이해도를 높일 수 있으며, 업무가 복잡하고 큰 시스템일수록 그 효과는 증가된다. 특히 팀으로 공동작업을 할 때 업무분할, 업무관리, 의사소통 등 여러 면에서 차별화된 효과를 얻을 수 있다.

이외에 상속을 이용하여 업무기술 문서의 양을 최소화할 수 있으며, 업무의 추가, 변경, 삭제 등의 필요성이 생길 때 변경을 최소로 하면서 업무기술서를 보수할 수 있어 융통성이 대단히 높다. 또한 모듈화된 업무를 이용하여 복잡한 시스템으로 확장시 확장이 용이하며, 다른

유사한 인간기계 응용시스템을 설계할 때도 업무의 재 활용성을 극대화할 수 있다.

만약 인간기계시스템이 설계평가 도구로 컴퓨터 모의 실험을 이용한다면 객체지향 업무기술서의 특성상 객체지향 프로그램으로의 전환이 용이할 수 있으나 구체적인 메시지 작성과 같은 구체적인 방법은 추후 연구될 것이다.

참고문헌

- [1] Annett, J., Duncan, K. D., Stammers, R.B. and Gray, M.J., Task Analysis, Training Information Paper No. 6, London : H.M.S.O., 1971.
- [2] Bailey, R. W., Human Performance Engineering, Prentice-Hall, 1996.
- [3] Coad P., and Yourdon, E., Object-Oriented Analysis, Object Int'l Inc., 1991.
- [4] Jacobson, I., Object-Oriented Software Engineering, ACM Press, 1992.
- [5] Kirwan, B. and Ainsworth, L. K., A Guide to Task Analysis, Taylor and Francis Ltd., 1992.
- [6] Miller, R. B., A Method for Man-Machine Task Analysis, Wright-Patterson Air Force Base, Ohio, USA : WADC Tech. Rept. No. 53, 137, 1953.
- [7] Meister, D. and Enderwick, T. P., Human Factors in System Design, Development, and Testing, Lawrence Erlbaum Associates, London, 2002.
- [8] Sanders M. S., and McCormick E. J., Human Factors in Engineering and Design, McGraw Hill, 1992.
- [9] Singleton, W. T., Man-Machine Systems, Harmondsworth : Penguin. Books, 1974.