

ATM 시스템 개발에서의 인간-기계 계면체계 설계를 위한 인간영역 정의 (Human Domain Paradigm for Man-Machine Interface System Design in ATM Switching System Development)

박범*, 김봉수*, 이병윤*, 김한경*

요 약

본연구는 ATM 시스템 개발에 있어서 인간-기계 계면체계에 더 적합한 설계 계획을 위한 새로운 인간영역에 관한 정의이다. ATM은 광대역 종합정보통신망에 사용될 비동기 전송방식 교환시스템으로 인간-기계 시스템 상호작용에 있어서 인간 운용자에게 높은 신뢰성과 양호한 적응성이 강력히 요구된다. 대형 통신시스템의 설계영역 정의에 대한 이런 관점은 다음과 같은 관련 영역을 포함하고 있다: 인간-기계 시스템의 구성요소; 객체지향적 설계 계획; 인간 프로세서 체계 모형의 영역; 인지작용의 원칙; 인간오류 행위의 영역 등이다. 또한, 제안하는 지식베이스 체계는 인간-기계 계면기능의 평가에 있어서 인간지향적 오류의 진단분석을 유용하게 할 수 있다. 이 적합한 인간지향적 영역 정의는 ATM 통신 교환체계의 운용보전에 있어서 적합한 설계영역을 위한 주요한 지식기반이 될 것이며, 사용자에 편리한 운용체계의 개발로 효율적인 성과를 가져올 것이다.

I. 서론

정보사회로의 진입에 의하여 폭주하는 각종 통신 정보와 멀티미디어 정보를 서비스 제공하고 처리할 수 있는 광대역 종합정보통신망(Broadband Integrated Services Digital Network)의 구축과 이를 운용할 수 있는 비동기 전송방식(Asynchronous Transfer Mode) 교환기의 개발이 진행되고 있다. 기존의 전자식 TDX-10 교환기의 시스템 수준과 비교하여 그 양적 경제적 규모성(Switch throughput 640 Gbps 이상), 고속성(Transmission Speed 155Mbps), 고신뢰성(시스템 고장 시간: 1분/년), 그리고 운용보전 상태의 질적 수준 등에 있어 월등한 차별이 요구되고 있다. 일반적으로 정보통신 시스템은 광대한 규모, 시스템 기능의 복잡화, 다기능화, 분산제어 구조, 그리고 강한 실시간성이 특성이다. 특히, Hardware의 비속적인 발전에 의하여 양적 수준이 향상되어 시스템의 물리적 요구에 상응되고 있으나 그 질적인 개발 및 운용 문제는 형상적 체계나 제어 처리 수준에 있어서 본질적인 시스템 구성요소들의 적합성을 결여하고 있다.

ATM 시스템 구성요소들간의 상호작용에 의하여 요소 개개의 독립적이고 또한 상호 의존적인 문제 상황은 외재적 원인과 더불어 그 내재적 원인 요소에 의하여 결정되어 진다. 시스템 개발의 성능에 있어서, 비효율적 요소는 시스템 구성요소들의 상호 상충과 이들의 내재적 문제인 System Integration에 있으며, 특히 그 질적인 문제는 인간요소의 시스템 적합성에 기인한다. 대형 규모의 통신 시스템을 개발에 있어서 인간 지향적 적합시스템을 구현하기 위해서는 설계단계에서 인간요소 영역을 정확하게 정의하고 적용하여야 할 것이다.

* ATM 운용보전연구실
전자통신연구소

II. 인간-기계 계면체계의 기능과 시스템 구성요소

차세대 종합정보 통신망 교환시스템에[1] 있어서 인간-기계 계면체계(Man-Machine Interface System: MMI)는 각 Subsystem의 Switching Board인 Processor들을 중심으로 I/O device 등과 같은 시스템 Hardware, Operating, Administration, and Maintenance 등을 위한 Software 시스템, 그리고 이들을 개발하는 Software Engineer와 운용하고 관리하는 운용자(Operator)로 구성되어 있다. 이들의 상호작용의 고려 요소는 다음과 같다.

- 인간측면: 유용성, 능력, 성과, 생산성, 안전성
- Software 측면: 설계의 요구사항, 사용자 편의성, 계면기능, 효율성
- 시스템 Hardware: 정보 출력 표현, 표현의 특성, 자료의 구성, 대화, 절차, 자료 입력장치, 문서 등.

또한, ATM의 MMI는 구성요소들의 상호작용에 의하여 다음과 같은 기능들을 수행한다.

A. 대화 지원 처리 기능 (Dialogue Support Function)

시스템의 운용 및 유지보수를 위해 운용자와 시스템간 대화의 편의성과 운용의 용이성 및 오류 방지를 위한 기능으로, 다음과 같은 네가지 명령어 입력 방식을 제공하며, 이들 사이의 권이가 이루어지도록 하는 기능이다.

1. 직접 명령어 처리
2. 메뉴 방식에 의한 명령어 처리
3. 서식 기입에 의한 명령어 처리
4. 명령어 화입에 의한 명령어 처리

B. 도움말 제공 기능 (Help Message Provision)

시스템 운용자의 요구에 의해, 운용자가 알고자 하는 기능에 대하여 상세히 설명하며 또한 다음에 취할 행동에 대하여 알려주는 기능으로 명령어 설명이나 시스템 운용 및 유지보수에 필요한 내용등을 운용자에게 제공한다.

C. 입출력 윈도우 관리 기능 (I/O Window Management Function)

입출력 윈도우 관리는 입출력의 다중처리와 시각적 효과의 증대를 통한 메시지 전달을 위하여 하나의 화면을 여러개의 윈도우로 분할하여 이들 입출력 윈도우들을 운용자의 요구에 의해 생성, 닫힘, 크기조정, 위치조정, 및 드러냄 기능을 제공한다.

D. 시스템 입출력 허용 범위 관리 (System Resource Usage Privilege Control)

시스템을 운용하는 운용자의 시스템 접근을 제어하는 기능으로, 운용자 또는 입출력 장치의 종류에 따라 사용할 수 있는 대화 범위를 제한하며, 허용된 운용자 요구를 처리한다.

E. 시스템 입출력 관리 (System I/O Management)

운용자의 요구에 의해 시스템 이용자의 login 기록 및 출력, 입력된 명령어 기록 및 출력, 현재 수행중인 명령어 중지/현황 출력, 출력된 메시지 기록 및 출력, 메시지 출력 금지/해제/현황 출력 등을 수행한다.

F. 운용 집중 보전 센터와의 상호작용 (O&M Center Interface)

원격지에 위치한 O&M 센터에서 시스템의 운용 및 유지보수가 가능하도록 시스템의 상태, 정보 및 장애를 데이터 링크를 통해 O&M 센터로 전송하고 O&M 센터로부터 제어 명령을 받아 수행한다.

III. 인간-기계 계면체계 설계를 위한 영역 정의

A. 객체지향적 설계 계획

1. 객체지향 정의

객체지향(Object-Oriented)[2][7]이란 자료구조(data structure)와 동작(operation)으로 구체화된 객체들로서 소프트웨어를 구성하는것을 말한다. 객체지향 개념은 객체(object), 메시지, 상속(inheritance)의 세가지 요소가 근간을 이루고 있으며, 실세계의 모든 개체(entity)들은 시스템내에서 객체로모델링되며, 각 개체들은 인스턴스(instance)로 표현되고, 공통된 특성을 가지는 인스턴스들이 모여 하나의 클래스를 구성하게 된다. 객체지향의 목적으로는 시스템의 유연한 구조를 도모할 수 있으며, 소프트웨어를 부품으로 취급하여 재사용성을 향상 시켰으며, 이에 따르는 확장성이 증대되었다. 객체지향의 접근방식으로는 클래스, 객체, 상속 및 다형(polymorphism)을 취급함으로써 이에 접근할 수 있다.

2. 객체지향적 소프트웨어 개발단계

Software 개발의 설계단계에 있어서 객체지향적 접근은 주요한 설계지침이 된다. 설계시의 각 개발 단계[2]는 다음과 같다.

- a) 분석단계에서 시스템은 문제영역내의 문제에 대한 해결을 목적으로하고 시스템과 문제영역내의 기본적인 객체들을 결정하며 이 객체들간의 관계 그리고 객체의 어트리뷰트와 동작(behavior)도 결정한다.
- b) 분석단계가 종료되어감에 따라 개발결과를 검증하기 위하여 분석평가 시나리오(scenario)가 만들어지는데 이는 분석모델이 시스템의 모든 기본적인 요구사항을 적절하게 만족 시키는지를 결정한다.
- c) 시스템 설계단계에서는 시스템이 어떻게 동작하는지를 나타내는데 시스템 객체의 동작과 이 객체들간의 연동관계에 대하여 구체적으로 명시한다.
- d) 이 단계에서는 분석단계에서 결정된 객체를 상세화하고 시스템을 동작시키기 위하여 요구되는 부가적인 객체들을 결정한다.
- e) 시스템설계 내용을 검증하기 위하여 평가 시나리오를 준비하는데 이는 분석단계에서 준비된 평가 시나리오를 더 상세화시킨것으로 시스템 요구사항을 수행하는데 필요한 메카니즘들이 시스템설계 모델에 나타나기를 확인한다.
- f) 소프트웨어 설계단계는 특정 하드웨어 플랫폼과 소프트웨어 여건에서 어떤 특정 프로그래밍 언어로 이 시스템을 어떻게 구현시킬 것인지를 결정하고 분석단계와 시스템설계 단계에서 정의된 객체들에 대한 실현 정보를 추가한다.
- g) 소프트웨어평가 시나리오는 시스템평가 시나리오를 상세화한 것이며 시스템을 시험하기 위한 모든 데이터를 제공하며 소프트웨어 설계가 완료되면 이에 따라 바로 구현이 가능하다.
- h) 시스템 개발 노력에 있어서 검토의 빈도를 많이 하는 것은 필수인데 이때 평가 시나리오가 중요한 역할을 한다.
- i) 기술문서는 서술형태와 도표 형태를 취하는데 이는 전체 개발 과정을 통하여 수정되어지고 매우 중요한 역할을 한다.

3. 객체지향적 정보분석

정보분석의 목적은 문제영역 내에 있는 개체들과 이들 사이의 관계를 모형화하는 것인데 개체들은 문제영역에 무엇이 존재하는지를 나타내는 데이터요소들을 내재한다. 따라서 분석자는 문제영역에 포함되는 모든 개체를 추출할 수 있는 기술문서를[2] 이용해야만 하는데 이는 다음과 같은 것이 있다.

a) 개체상관도 (ERD: Entity Relation Diagram): 정보모델이라고 부르기도 하며 문제영역내에 무엇이 존재하는가를 결정하기 위하여 요구사항 분석 및 사건분석 업무와 관련하여 이용한다. 어떤 개체가 객체로 귀착되어야 하는가를 결정하고 본체상관도에 표현된 개체들을 객체로 변환하기 위하여 사용된다.

b) 객체관계도 (ORD: Object Relation Diagram): 객체와 그 속성(attribute), 부객체, 동작(services or method) 및 객체간의 관계를 나타내는데 데이터 요소는 객체이거나 또는 객체의 속성이 되고 객체관계도내의 객체들은 서브젝트로 통합하여 문제영역을 관리가 가능한 단위로 더 세분화할 수 있다.

c) 객체설명서 (OD: Object Description): 각 객체에 대한 문장형태의 설명으로 객체관계도내에 있는 각 객체에 대한 구체적인 특징을 기술하며 개발순기를 통하여 능동적으로 보완이 되고 내용이 확충되어야 하는 핵심적인 문서이다.

d) 객체참조 문서(OCR: Object Cross Reference): 개발순기를 통하여 보완되고 확장되어야 할 기술문서로 테이블 형태로 표현되며 모든 객체와 그들이 제공하는 서비스(동작), 다른 객체로부터 처리요구되는 서비스(동작)를 나타낸다. OD로부터 파생되어 나타나는 객체참조 문서는 이후의 유지보수 활동에 있어 가장 유용한 문서가 되며 이는 컴퓨터 이용 소프트웨어 엔지니어링 도구를 이용하여 자동적으로 생성하는 것이 바람직하다.

e) 상속도 (ID: Inheritance Diagram): 객체와 그들의 속성, 서비스, 그리고 객체 클래스간의 상속관계를 나타는데 상속이란 일반화-특수화 관계인데 슈퍼클래스는 일반화된 것이고 서브클래스는 특수화된 것이다.

B. 인간 프로세서체계 모형의 영역

인간 운용자는 연관 기계와 시스템과의 상호작용과 전이에 의하여 외부 정보 자극을 습득하고 이를 자체 판단하여 이에 응답하고 상황에 순응한다. 이러한 인간의 처리체계 영역에는[4] 감지, 인지, 그리고 운동기능의 세가지 처리체계로 분류되고 이들은 각각 저장능력(μ), 감퇴시간(δ), 코드화 타입(κ) 등의 파라미터에 의해 정의되어 진다.

1. 감식 처리체계(Perceptual Processor System): 이 체계는 신체의 감각장치에 의해 탐지된 물리계의 외부 자극 정보인 감각들이 통합된 감각체계의 수단에 의해 마음의 내부적 지각 표현으로 전이하는 처리과정이다. 또한, 이 체계는 감각기와 감각체계의 결과가 부호로 작성되어지는 동안 이를 보관하기 위해 시각 형상저장이나 청각 형상저장과 같은 연관 기억완충역(Memory Buffer)으로 구성되어 있다.

2. 인지 처리체계(Cognitive Processor System): 인지체계는 부호로 작성 인식된 정보를 활동 기억력에 있는 감각 형상저장으로부터 받아들이고, 또한 어떻게 응답할 것인가를 결정하기 위해 장기 기억장치에 있는 이미 저장된 정보를 사용한다.

3. 운동기능 처리체계(Motor-Skill Processor System): 이는 응답과정의 결과를 실행한다. 체계 운용자의 생각은 수의근의 실행 패턴들에 의해 일련의 이산 미동작으로 된 행위로 옮겨진다.

C. 인지작용의 원칙

인간과 컴퓨터 상호 관계론의 입장에서 신체적 그리고 심리적인 인간의 인지작용의 원칙은 다음과 같이 열가지로 분류 할 수 있다[4].

1. 인지처리장치의 식별행동 주기
2. 변동감지 처리율의 원칙
3. 기호화 한정성 원칙
4. 차별력의 원칙
5. 변동인지 처리율의 원칙
6. Fitts의 법칙
7. 숙련의 능력 법칙
8. 불확실성의 원칙
9. 합리성의 원칙
10. 문제 영역의 원칙

D. 인간 오류행위적 영역

인간-컴퓨터 관계에서 인간지향적 오류의 속성을 진단하는데 있어서 인간행위의 오류에 근거한 영역은[9] 숙련, 규칙, 지식, 그리고 모형에의한 행위영역으로 나눌 수 있다.

1. 숙련에의한 행위영역은 감각기 운동 패턴 조정 이나 자동적 행위이며, 인간신경체계에 저장된 적합한 양식의 구조에 의해 조정된다. 이것은 인간오류 행위는 인간-소프트웨어 처리 구조를 통제하는 생리적인 법칙에 의하여 조정되어지는 것을 뜻한다. 숙련에의한 행위영역에 잘 어울리지

않는 오류 구성의 특성은 감각-운동근육 변이성, 최신성과 빈도성, 지형적 오지향성, 환경적 조정 신호, 진부한 취득, 공유한 윤곽 특징들, 적합성과 정교한 조율 등이다.

2. 규칙에의한 행위영역은 유사한 작업 환경에서 덜 빈번한 과업을 대해 목표지향적이고 전형적이다. 규칙베이스 체계는 패턴매칭의 간단한 형태로서 어떻게 일련의 심볼들이 다른 일련의 심볼들로 변환되는지를 지정하는 규칙들의 세트로서 어떤문제의 해결을 표현한다. 이 영역의 오류형태의 특성은 습관의 견고성, 전형적인 고착, 유효성, 고립된 기능의 생략, 과도한 간결성, 대안의 실수, 과신, 진부한 취득, 조화하는데 있어서의 편견 등이다.

3. 지식에의한 행위영역은 소프트웨어 개발에 있어 설계나 프로그램 적용에 작용하는 인간행위 단계이며, 그러한 적용은 문제해결, 원칙의 영역에서 과업의 인식, 목표로 부터의 객체지향, 소프트웨어 제품과 인간-체계와의 상호작용 처리와의 동시적 설계, 그리고 최적의 체계설계 등과 같은 인간의 인지 능력을 동격 실행할 수 있다. 이 영역의 오류형태의 특성은 인간의 가변성, 선택성, 적합성, 실행 기억력의 한계, 원인구조의 순차적 이유의 계속적인 추적, 유효성, 시스템요구사항에 적합하지 않는 행동, 주제의 방랑과 형성, 되돌아오는 조화에의 편견, 기억의 실마리나 유추에 의한 원인규명, 부정확하고 불완전한 지식 등이다.

4. 모형에의한 오류 행위영역은 인간-소프트웨어 정보처리과정이나 인지적 모의실험으로 인간의 오류방지 기능을 개발하는데 있어 분석적 설계구조에 조력한다. 이들의 특성과 분류는 표징적 또는 수량적 모형, 성과적 또는 인지적 모형, 정적 또는 동적 모형, 구문론적 또는 의미론적 모형, 상태-전이 모형, 단일 또는 복수 모형 등이다.

IV. 지식 베이스 구조와 인간 오류의 진단 분석

인간-기계 계면체계의 적합한 설계를 위해 동일한 특성과 비슷한 데이터구조를 갖는 지식과 정보들을 귀합하고 처리하여 설계작업에 생산적이고 경제적인 규범을 구조화 할 수 있다. 설계 체계 영역은 문제영역(Problem space)계층, 목표(Goal)계층, 분류(Class)계층, 객체(Object)기본 계층, 그리고 동작(Operation)계층으로 구조화 할 수 있고 각 계층간에는 그영역의 특성과 속성에 의하여 상태천이가 이루어진다. 문제영역으로부터 그 통합적 목표, 유사한 범위의 부류인식과 본질적 단위 객체, 그리고 객체의 기능을 설정하는 동작결정 등으로 분석되어가는 흐름을 하향연쇄(Down-load chain)하고, 그 반대의 지식 정보의 통합과정을 상향연쇄(Up-load chain)라 한다.

인간-기계 상호작용에서 체계의 구성요소가 정합상의 상충이 발생하고 이들을 프로토퀴이 호출접속하며, 그 처리규범의 영역을 정의하면 다음과 같다.

1. Let V be the set of MMI system design,
 $V_i \subseteq V, i=1, 2, \dots, n$ for n product.
2. Set of common cause mode C_{ij} :
 $C_{ij} = C_{ij}(I_x, P_y, B_z)$,
 where $i: 1, 2, \dots, n$ for n product;
 $j: 1, 2, \dots, m$ for m failures;
 $x: \#$ of identification mode
 $y: \#$ of pattern recognition mode
 $z: \#$ of behavior domain.
3. Set of processing inference:
 $I_{ij} = C_{ij} \cup K_{ij}, C_{ij} \subseteq C, K_{ij} \subseteq K$
 where C : set of common cause mode,
 K : set of knowledge base.

V. 결론

ATM 교환시스템의 운용보전부문의 인간계면체계의 기능과 시스템 구성 요소를 분석 이해하고, 인간지향적이고 효율적인 객체지향적 설계 계획을 제안했다. 인간의 처리장치와 인지작업 원칙, 그리고 인간오류 행위적 영역은 인간-기계 계면체계의 설계 정의에 중요한 본질이 되며, 오류진단과 유지보수를 위한 지식베이스는 어떤 상황의 적절한 규칙이되어 처리되어진다. 또한, 제안하는 지식베이스 체계는 인간-기계 계면기능의 평가에 있어서 인간지향적 오류의 진단분석을 유용하게 할 수 있다. 이 적합한 인간지향적 영역 정의는 ATM 통신교환 운용보전에 있어서 적합한 설계영역을 위한 주요한 지식기반이 될 것이며, 사용자에 편리한 운용체계의 개발로 효율적인

그리고 경제적인 성과를 가져 올 것이다.

VI. 참고문헌

- [1] ATM 기술연구부, "차세대 교환기술(ATM) 연구개발," ETRI, Dec. 1992, pp.403-466.
- [2] ATM 운용보전연구실, "The Concept of Object-Oriented and the Outline of B-ISDN," ETRI, May, 1993, pp. 16-33.
- [3] Basu, Amit, "A Knowledge Representation Model for Multiuser Knowledge based System. IEEE Trans. on Knowledge and Data Engineering, Vol.5, NO.2, Apr. 1993, pp.177-189.
- [4] Card, Stuart K., Thomas P. Moran. "The Psychology of Human-Computer Interaction." Lawrence Erlbaum Assco., 1983, chap. 2 and chap. 5.
- [5] Henley, Ernest. J., H. Kumamoto, "Designing for Reliability and Safety Control." Prentice Hall, Inc., Englewood Cliffs, N.J., 1985, pp. 273-275.
- [6] Hughes, J. G. "Object Oriented Knowledge Bases," Object Oriented Database, Prentice Hall, 1991, pp.234-251.
- [7] Jacobson, Ivar. "Object-Oriented Software Engineering," Addison-Wesley, 1992, pp. 43-107.
- [8] Lee, J. H., Hyung Ill Choi, "Adaptive Edge-detection Using Fuzzy Inference," Vol.20, No. 8, Journal of the Korean Information Science Society, Aug. 1993, pp. 1137-1147.
- [9] Park, Peom, "Common Cause Analysis in Human-Software Interaction: System Design, Error Control Mechanism, and Prevention." Iowa State Univ., Ames, Iowa, 1992.
- [10] Randell, B. "Fault Tolerant and System Structuring." IEEE, Preceeding of 4th Jerusalem Cof. on Information Technology, May 1984.
- [11] Ross, D., and K. Schoman, "Structured Analysis for Requirements Definition." IEEE Trans., Software Engineering, vol. 3, no. 1, Jan. 1977, pp. 6-15.
- [12] Shooman, M.L., "Software Engineering" McGraw-Hill Book Comp., New York, 1983, pp. 296-403.