

지능적 지도를 위한 인지진단 모형

신동익

<적요>

최근 실제상황에서 유용할 뿐아니라 인간인지과정을 동시에 연구할수 있는 컴퓨터 시스템을 설계하는 분야에 많은 관심이 쏠리고 있다. 그와같은 시스템의 한가지 종류로 Intelligent Tutoring System (ITS)이라는 것이 있다. ITS는 학생들의 학습을 도와주는 시스템이다. 본 연구의 목적은 지능적 지도 (intelligent tutoring)을 위한 인지진단 (cognitive diagnosis) 시스템을 설계 및 개발하는데 있다. 인지진단은 사람의 인지상태를 그의 실행을 통해 추론 (inference)하는 것을 말한다. 인지진단은 ITS 연구에 핵심적 문제이다. 본 연구는 PSCD (production system for cognitive diagnosis)라는 인지진단 시스템을 제안한다. 이와같은 인지진단 시스템은 지능적 지도에 뿐만 아니라 의사결정 지원에도 사용되어 질수 있다.

본 논문은 다음과 같은 순서로 구성되어 있다. 첫째, ITS를 설명하는 서론뒤에 인지진단의 필요성과 역할이 설명되어 진다. 그 뒤로 인지진단 시스템을 설계하는데 도움이 될 방법론 (methodology)이 설명된다. 제시된 방법론에 기초하여 인지진단의 기초구조 (architecture)를 형성하는 3가지 제약조건 (constraints)이 설명된다. 이와같은 기초구조에 입각한 인지진단 모형이 제안되며, 마지막으로 본 연구의 결론이 보여진다. 본 논문은 영어로 되어 있으므로 독자들의 편의를 위해 좀더 자세한 확대된 요약물 아래에 제공한다.

<확대 적요>

ITS는 학생들이 인지기술 (cognitive skills)의 학습을 도와주는 컴퓨터 프로그램이다. ITS는 많은 인지심리학자 (cognitive psychologists)와 인공지능학자 (artificial intelligent researchers)의 관심을 끌고 있으며, 또한 사회에 유능한 인재를 양성하여 배출하여야 할 책임을 가지고 있는 교육자 역시 많은 관심을 갖고 있다. 이외에도 ITS는 회사에서 신입사원의 교육 및 재교육, 군대에서 특수한 분야의 효과적 교육등에도 널리 사용될수 있으므로 그 유용성은 매우 넓다 하겠다. ITS의 궁극적 목적은 경험많은 가정교사와 같이 학생들을 지도할수 있는 능력을 갖는데 있다.

ITS는 일대일 지도 (one-to-one tutoring)을 표방한다. 이와같은 일대일 지도의 효용성은 Bloom (1984)의 연구에서 나타난다. Bloom은 그의 연구에서 일대일 지도가 다른 어떤 재래식 교육방법 (e.g., 강의식 교육)보다 뛰어난을 보여주었다. 이와같은 일대일 지도의 효용성은 일대일 지도에서는 교사가 학생의 욕구나 부족한 지식 또는 잘못알고 있는 지식 (misconception)등을 참조하여 각 학생에 적합하게 지도할수 있기 때문이다 (그림 1 참조). 따라서 ITS는 학생의 문제수행이나 혹은 직접적인 질문으로부터 그 학생의 문제점을 찾는 장치 (mechanism)이 필요하다. 이와같은

의 자동화 (automation) 를 목적으로하며 이는 구체적으로 적용을 설명하는 것 뿐만 아니라 data 모델화 과정을 포함하고 있다. Fachkonzept 에 해당하는 적용시스템은 프로그램 시스템과 처리해야 되는 자료로 구성되어 있다. 정보시스템은 개방시스템으로서 시스템을 둘러싸고 있는 주변시스템들 (environment) 과 끈임없이 자료를 교환하고 있으며, 자료교환에 관여되는 모든 종류의 자료들은 언제나 특정 type 의 대상 (object) 을 설명하고 있기 때문에 정보처리과정에서 투입되고 산출되는 모든 종류의 자료들을 특히 object type 이라 부른다. 정보시스템은 또한 다른 시스템 소위 기본시스템 (basic system) 을 계획하고 실행하며 이를 통제하기 위한 시스템이라 정의할 수 있다.

OrgIS 의 기본시스템은 대상기업의 조직²⁾이며 이에 는 구체적으로 구성조직, 운영조직, system configuration 그리고 해당분야에서 컴퓨터 지원하에 이루어지는 적용 (application) 과 적용시스템 (application system) 자체가 포함된다. OrgIS 는 이 구성요소로부터 고유의 data 모델을 형성하게 되는데 OrgIS 의 기능은 이들 data 모델을 구체적으로 구상하고 이를 관리하며 더 나아가서 분석 평가함으로써 이를 실지로 기업에 이용하는 것이라 할 수 있다. OrgIS 는 지금까지 약속한 기본 개념을 통해서 볼때에 기업에서 실제로 사용 되어질 수 있는 구체적인 정보시스템이라 할 수 있다.

1.2. Object type, object type net 그리고 object type system

OrgIS 의 data 모델화과정에서³⁾ 나타나는 주요대상 (Objektbereich: ObjBer) 은 상술한 바와같이 기업의 조직과 컴퓨터 지원하에 이루어지는 적용방법과 구체적인 적용시스템이다. 이 대상으로 부터 OrgIS 는 object type net 라 불리우는 data 모델을 형성한다. Object type 은 많은 속성 (attribute) 들에 의해서 특징지워 지는데 그 속성중의 일부는 주요 key (source key) 속성으로서 나머지는 보조 key (descript key) 속성으로서 object type 을 설명한다. 특히 후자의 속성들이 모여서 object 를 확인 (identification) 하는 속성으로서의 역할을 담당하기도 한다. 모든 속성들은 또한 특정범위의 값을 가지고 있으며, object 는 이를 특징짓는 속성과 이 속성의 특정한 값이 1:1 대응되는 모형으로 이해되어 질수 있다. 같은 종류에 속하는 object 들이 동일 type 에 소속되기 위해서는 일련의 조건들이 충족되어야 하는데, 이때에 이 조건들을 충족시키는 모든 object 들을 동일 object type 에 속한다라고 한다. Data 모델을 구성하기 위해서는 일반적으로 많

2. 여기에서 조직은 기업의 경영진에 의해서 의식적으로 형성된, 조직적인 형태의 한 부분이라할 수 있다 (KOS, NOR 참조). KOS, NOR 에서 system configuration 은 전산시스템의 작업수단으로 해석되어짐.

3. 표준적인 data 모델화과정은 (CHE, COD, DAT, ULL) 에서 발견할 수가 있다. 여기에서 사용된 object type net 의 방법은 Entity-Relationship-Concept 의 발전된 모형이며, object type net 의 특징은 모델형성시 요구되는 조건들을 단계적으로 작성함으로써 모델화대상의 의미를 data 모델에 구체적으로 반영했다는 것이다. 모델의 정규화 (normalization) 과정에 대해서는 (ULL) 참조.

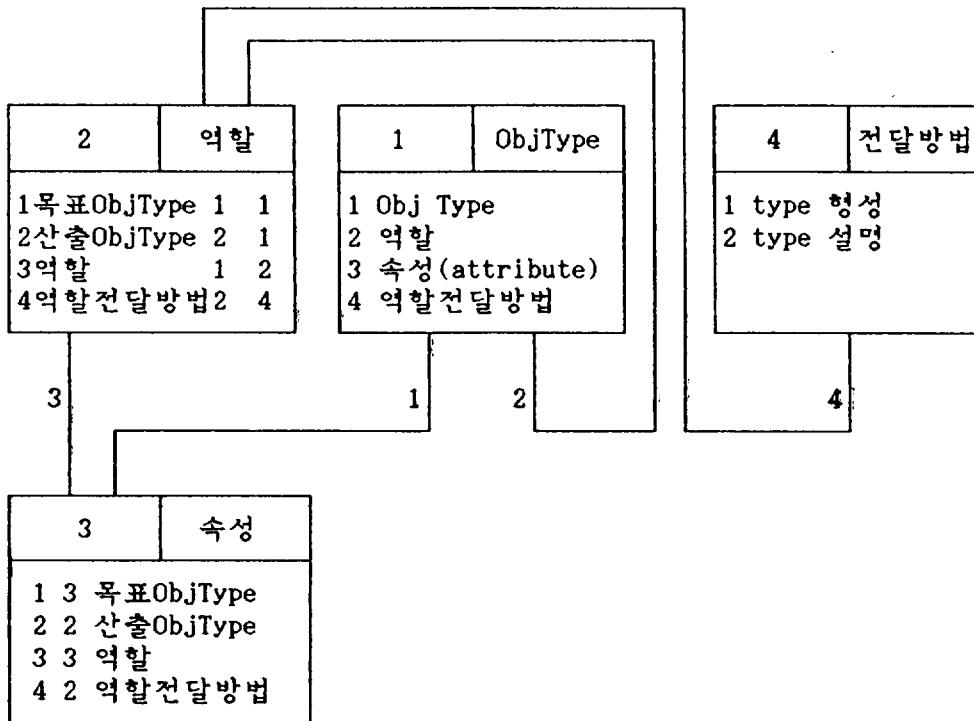
은 object type 이 필요한데 부분적으로는 basic type 으로 주어지기도 하지만 대부분의 object type 들은 basic type 으로부터 유도 되어지는 파생적인 object type 들로 이루어져 있다. 이 파생적인 object type 들을 특히 relation type 이라 부른다. Relation type 의 존재여부는 다른 object type 들 즉, basic type 또는 다른 relation type 의 존재에 종속되어 있는데, 후자의 object type 들은 자신들의 주요 key 속성을 소위 역할 본담시켜서 전자의 relation type 에 전달하면 여기에서 다시 주요 key 속성으로서 역할을 담당한다. Relation type 의 모든 주요 key 속성은 이런 방법으로 다른 object type 으로부터 계승 받는다. Relation type 의 존재여부가 하나의 또는 여러 object type 에 종속되어 있을때 relation type 은 단 단위 또는 다 단위라 불리우며, 특히 단 단위의 relation type 을 sub type 이라 칭한다. 주요 key 속성을 이어받는데 있어서 상기한 방식이 유일한 방법은 아니며 다른 형태를 취할수가 있는데, object type 은 자신의 key 속성으로부터 형성된 역할을 새로운 object type 을 만들어 내는 것이 아니라 단지 다른 object type 을 설명해 주는 것으로 전달해 줄수도 있기 때문이다. 일반적으로 한 object type 의 주요 key 속성으로부터 많은 역할이 형성될수 있는데 이 역할들은 동시에 많은 relation type 에 새로운 type 을 형성하는 방식으로 또는 단지 그 type 을 설명해 주는 방식으로 전해진다. Data 모델에 기본이 되는 모든 object type 은 이를 바탕으로 정의 되어지는 역할전달형식과 더불어 object type net 를 이룬다. Object type net 는 이에 소요되는 모든 object type 에 동일 type 의 모든 object 들을 투입함으로 이루어지며 이렇게 형성된 object type net 는 다시 일정 조건을 충족시켜야 하는데 이 조건이 충족될때 그 전체를 object type system 이라 부른다. 예를 들어 object type net 를 형성할때 다른 object type 으로부터 이어받는 속성과 이 속성에 상응하는 역할이 1:1 대응 되어야 한다는것이 그 조건의 한 예로서 OrgIS 의 관점에서 볼때에 이와같은 조건들과 object type net 가 합쳐서 비로서 data 모델이 형성된다.

1.3. Object data 모델, meta data 모델 그리고 data 모델의 기본 개념

현재 개발중에 있는 발전된 OrgIS 는 data 모델을 meta data 를 통해서 파악하며 다른 data 모델, 소위 meta data modell⁴⁾ (*MetaDatMod*) 로서 설명할수 있는 가능성을 제시한다. OrgIS 의 발전된 모형에서는 data 모델을 object 분야의 meta data 모델의 시야로 보기 때문에 Object data model (*ObjDatMod*) 이라고도 부른다. Meta data 모델 또한 앞절에서 기술한 의미의 data 모델로 이해될수 있지만 이에선 object data 모델과의 meta 관계가 OrgIS - object type net 의 기본개념, 즉 object type, object type 을 특징짓는 속성, 이 속성의 역할 그리고 이 역할이 전해지는 방법 등에 내포되어 있기때문에 기본개념들 사이의 관계와 그에 상응하는 meta object

4. OrgIS 의 meta data 모델개념은 관계시스템 (relational system) 에서의 data base 목록개념과 비교할 수가 있다 (참조 : ULL, p.225 ; COD, p.277). 그러나 data base 목록과는 달리 meta data 모델에서는 모델형성시 요구되는 조건들이 설명될수 있다는 차이점을 발견할 수 있으며 이로 인해서 데이터 관리기능이 수월하게 된다.

type 사이의 관계가 새로이 정의되어야 한다. Meta data 모델은 이러한 방법으로 object type net 를 설명하기 때문에 그 스스로가 모델화 될수 있다. 이를 위해서는 meta 분야의 최소한의 부분 net 가 필요하며 이에는 object type, object type 을 특징짓는 속성, 이 속성의 역할이 전수되는 방법 그리고 object type net 의 형성을 설명해 주는 역할분담이 포함된다.



네모상자는 object type 을, 선들은 전해지는 역할이 새로운 object type 을 형성하느냐 또는 단지 설명 하느냐에 따라서 유전자로부터 나와서 상속자에게 이어지는 전달방법을 나타낸다. Key 는 네모상자의 상단 왼쪽에, 오른쪽에는 object type 이 각각 기록된다. 여기에서 모든 선들은 표시된 역할의 key 를 포함하는 수로 표시된다. 네모상자의 하단부분에는 key 와 고유의 표식을 통해서 같은 tpye 에 속하는 object 들이 기록된다.

도 1 : Data model basic in OrgIS

OrgIS-object type net 는 일종의 Semi-hypergraph* 로 표시되는데, 그래프 상에 나타나는 매듭 (node) 은 object type 을, 한 object type 으로부터 나와서 여러 object type 으로 연결되는 선들 (edge) 은 object type 이 다른 object type 을 만들어내는지 또는 단지 설명해 주는지를 나타낸다. 모든 선들은 정확히 한 object type 으로부터 시작되어 다른 object type 으로 연결되기 때문에 이로부터 역할이 구분된다. Meta 분야의 최소한의 object type net 는 이런 의미에서 OrgIS-object type net 를 형성하는 시스템이며 이를 data model basic (*DatModBas*) 이라 부른다.

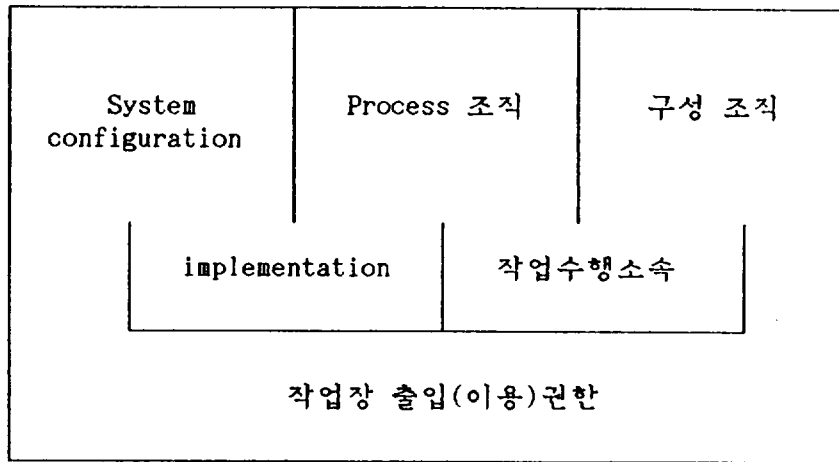
2. OrgIS 의 기본시스템 (basic system)

2.1. 조직과 그의 정보시스템

OrgIS 가 모델화 대상으로 삼고 있는 object 부분의 핵이라 할 수 있는 기업의 조직은 구성조직 (struktur organization) 과 process 조직 그리고 system configuration 으로 분류된다.

Process 조직은 정보처리기능과 정보처리과정에서 투입되고 결과로서 산출되는 모든 object type 을 포괄하며 process 조직을 통해서 자료의 투입 산출 관계뿐만 아니라 정보흐름의 의미에서 본 정보처리순서가 결정된다. 이외에도 process 조직을 통해서 정보처리기능의 구성 뿐만 아니라 다시 기능과 object type 를 포함하는 요소로 구성된, object type 의 구성에 관한 자료도 얻을 수 있다. 여기에서 말하는 object type 은 확대된 개념으로 단순 또는 복합자료를 불문하고 또한 어떤 정보처리수반자에게 속한 자료인지에 관계없이 인적 또는 물적(기계적) 정보처리과정에서 나타나는 모든 종류의 자료가 이에 포함된다. 복합적인 정보처리기능을 관찰하고 지각하는 역할은 구성조직중의 한 부서를 통해서 이루어 진다. 이런 부서들로 기업의 최소한의 조직단위를 이루며 이들 부서들은 상급부서에 소속되어 있다. 이 상급부서들은 또한 더 높은 부서의 지휘를 받는다. 이러한 조직구성은 상급 조직부서에 소속 단위부서의 정보처리에 관한 지휘권을 행사할 수 있도록 권리를 부여함은 물론 작업처리의 개개의 수반자에게까지 영향력을 행사할 수 있도록 구성되어 있다. 정보처리기능을 담당하는 부서와 이 부서가 어느 상급부서의 통제를 받느냐에 따라서 process 조직에서 나타나는 정보흐름도가 조직 단위들 사이의 communication 흐름도로 바뀔 수 있다. 정보처리의 성격을 띠는 기능은 모든 조직의 근간을 이루며 실제작업의 임무만을 띠는 것이 아니라 기업경영 전반의 지휘에 관한 그리고 작업의 전문적인 임무까지 띠고 있다. 정보처리기능의 수행은 기업 구성조직중의 한 부서에서 이루어 지며 해당 부서에 기능수행의 책임이 전가 된다. 이러한 기능의 수행은 인적 수단에 의해서 뿐만 아니라 물적(기계적) 수단에 의해서도 이루어 지기 때문에 기능의 수행자는 담당 부서의 책임자 또는 기계 즉, system configuration 의 한 구성 요소로서의 적용시스템이 될 수 있다. Process 조직에 의해서 수행되는 이 모든 기능은 최종기업전략상의 목적에 부합되는 작업과 깊은 연관성을 갖는다.

* Semi-hypergraph 는 이에 소요되는 매듭 (node) 과 서로 연결된 선들 (edge) 그리고 모든 선들에 정확히 하나의 시작매듭과 많은 목적매듭이 조합된 모형으로 구성되어 있다.



도 2 : OrgIS 의 조직모델

기업 조직에 관여되는 system configuration 은 시스템중의 시스템으로 이해되어 지는데, 이는 사용자 (user)조직의 정보처리 기술상의 기본구조를 이루며 적용시스템 (application system)과 그의 수반 시스템 (일종의 bearer system)으로 구성되어 있다. 수반시스템은 구체적인 자료를 처리하기 위한 중앙장치, 주변장치, 상호 정보를 교환할수 있는 통신 (communication) 장치 그리고 hardware 의 작동을 위해 필요한 기본프로그램 (basic software)으로 구성되어 있으며 적절한 적용시스템의 수용과 사용자의 공급을 위한 시스템으로 이해된다. 모든 적용 시스템은 적용 software 와 포괄적인 data base 의 한 단면으로서 조직되고 관리가능한 수반 시스템에 저장되어 있는 data set 로 구성되어 있다. System configuration 은 결론적으로 software, hardware, 그리고 hardware 를 연결시켜주는 매개체나 hardware connecting 과 같은 정보통신을 위한 장치들 그리고 시스템 communication 과 시스템 조건들과 시스템 구성체계 (system structur)로 이루어져 있다. Process 조직의 컴퓨터지원은 적용시스템의 기능과 object type 이 실험적용 (implementation)되고 사용자와 시스템 사이 뿐만 아니라 사용자들 사이 또는 시스템 자신들 사이의 communication 이 시스템을 통해서 이루어짐을 의미한다. 여기에서 사용자라함은 terminal 이나 system configuration 의 작업장에 출입할수 있는 자격을 가지고 있으며, 기능수행의 소속권한으로 적용 시스템을 이용할수 있는 권리를 부여 받은 구성조직상의 담당부서 책임자를 의미한다. 구성조직과 process 조직이 소속규정을 통해서 서로 연결되어진 것처럼 system configuration 과 구성조직과 process 조직 또한 서로 연결되어 있다. 이러한 관계는 정보처리기능과 그의 object type 이 적용 시스템에 implementation 됨으로 그리고 system configuration 의 작업장이나 terminal 을 갖춘 구성조직의 담당부서를 설치함으로써 형성된다.

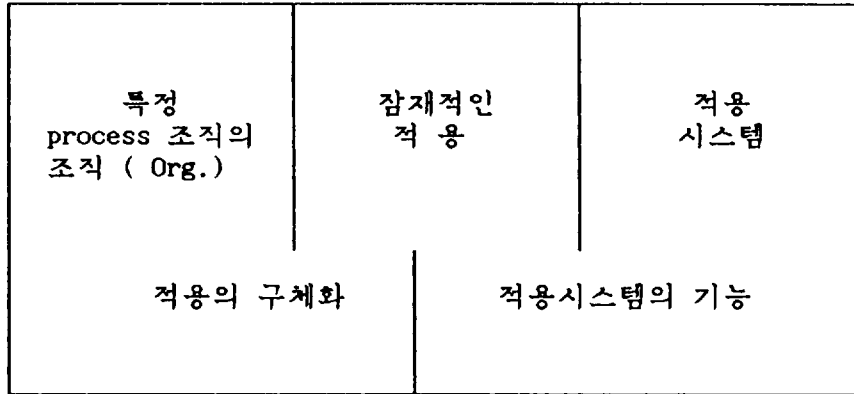
2.2. 적용 (application) 과 적용시스템 (application system)

Process 조직은 정보흐름도의 한 부분, 특히 이 부분에 포함되어 있는 기능과 object type 들이 구체적인 적용 시스템에 implementation 됨으로 해서 자동화 된다. 이러한 자동화 과정은 인적 기능수행자가 기계로 또는 인적 수행자가 직접 처리하던 자료가 시스템화되어 처리되어지는 것에만 국한되는 것이 아니라 일반적으로 전체 process 조직의 검토와 재 구성 또는 새로운 조직에 상응하는 기존 구성조직과 system configuration 의 변형까지를 포함한다. 상기한 process 조직의 자동화를 위해 먼저 process 조직에서 거론되는 모든 자료를, 그의 object type 이 특정의 정규화 요구를 충족시키고 더이상 분류될수 없기 때문에 자료 손실이 있을수 없다는 의미에서 간편하게 됐다라고 할 수 있는 data 모델에 환원시키고자 하는 노력을 경주하게 되었다⁹⁾. 이를 통해 복합적인 object type 으로 설명되어지는 자료와 사용자가 필요로 하는 모든 자료가 비토소 구체적으로 모델화 될수가 있기 때문이다. 이 뿐만아니라 아직 사용자들에 의해서 수행되어야할, 관리적인 성격으로서가 아니라 실무 처리상의 기능을 이 기능의 결과로서 산출되는 자료가 기본자료를 투입 시켰던 사용자의 interface 를 통해서 다시 적용 시스템에 전달될수 있도록 조직하려는 노력 또한 경주되고 있다. 여기에서도 마찬가지로 process 조직은 새로이 구성되었으나 구성조직내의 담당 부서간에 필수적인 communication 은 역시 communication 지향적인 적용 시스템과 사용자-interface 그리고 이에 따르는 수반시스템을 통해서 이루어 진다.

적용 시스템과 구체적인 적용은 개념상으로 구별되어야 한다. 이러한 개념상의 차이와 상응해서 컴퓨터 지원 적용은 process 조직중 그의 기능이 적용 시스템에 의해서 수행되며 이 기능의 단순 또는 복합적 object type 은 시스템화 되며 또는 사용자-interface 로서 시스템을 이용하여 처리되는 모든 부분으로 이해된다. 적용개념은 전적으로 사용자에 의해서 수행되는 기능에 한정되는 것이므로 적용이라 할때는 단지 적용 시스템의 procedure 에 상응하는 시스템을 통해서 처리되는 기능과 적용의 data 모델에 투입되는 단순 object type 만을 포괄하며 이에 는 적용 시스템의 사용자-interface 는 포함되지 않는다. 이런 의미에서 적용은 process 조직의 한 단면이라 할수 있으며 이는 구체적으로 적용시스템이 process 조직에 미치는 영향을 나타낸다. 반면에 적용 시스템은 궁극적으로 같은 목적을 달성하기 위해 투입되지만 세부적으로 이 목적을 효율적으로 달성하기 위한 시스템의 여러가지 기능의 선택을 활성화할 수 있는 여러 process 조직에 구체적으로 사용될 수 있는 시스템이라 할 수 있다. 이런 견지에서 볼때에 process 조직에 포괄적으로 적용될 수 있는 가능성을 보이는 잠재적인 적용도 존재할 수가 있다. 그러므로 적용에서도 두가지 상이한 개념을 발견할 수 있는데, 한가지는 특정한 process 조직의 한 부분으로서 이 조직에 구체적으로 적용되는 실질적인 적용이며 다른 한가지는 적용시스템 자체의 포괄적인 적용 가능성을 표현하는 잠재적인 적용이 그것이다. 실질적인 적용은 그래서 잠재적인 적용과 언제나 일치되는 것은 아니며 혹 더많은 또는 적은 잠재적인 적용 가능성을 포함할 수 있다. 적용 시스템자체는 기본적으로 system con-

5. 정규화과정에 관해서 (ULL ; DAT) 참조.

figuration 의 근간을 이루는 부분이라 생각할 수 있으며 이는 적용 software 와 적용시스템을 수반하는 시스템 (bearer system)에 포함되어 있는 구체적인 data set 로 이루어져 있다. 적용시스템이 표준화된 시스템인 경우에 이 시스템은 일반적으로 시스템을 설치함으로 특히 구체적인 적용을 목적으로한 parameterizing 을 통해서 구체화된다. 여기에서도 적용시스템 상의 두가지의 개념을 구별할 수 있는데 하나는 system configuration 의 한 구성 부분으로서 설치되어진 실제적인 적용을 목적으로 한 적용시스템이요 다른 하나는 프로그램과 자료를 포괄하는 적용시스템 자체이다.



도 3 : OrgIS 의 기본시스템 (basic system)

OrgIS 의 목적은 기업의 조직과 이에 따르는 잠재적인 적용 그리고 구체적인 적용시스템을 연구 분석하여 발전시키는데 있다. OrgIS 의 이런 목적을 생각할때 조직과 잠재적인 적용 그리고 적용시스템 자체는 OrgIS 의 기본시스템을 이루며 동시에 OrgIS 에 의해서 형성되는 data 모델의 대상이 된다. 여기에서 OrgIS 에 의해서 형성되는 data 모델은 상기한 OrgIS 의 대상 (object)을 모델화할 뿐아니라 이 모델이 다시 OrgIS meta data 모델의 object 로 나타내기 때문에 object data 모델이라 부른다. 잠재적인 적용과 조직 그리고 잠재적인 적용과 적용시스템 사이에는 종속관계가 성립하는데, 한편으로 잠재적인 적용은 여러 분야의 process 조직에서 구체화될 수 있다는 관계와 다른 한편으로 적용시스템의 어떤 프로그램을 통해서 어떤 잠재적인 적용기능이 수행될 수 있는지를 나타내는 적용시스템의 기능이 확정된다는 관계이다.

3. OrgIS 의 기본 모형 (version) - object data 분야의 적용

OrgIS 의 모델화 대상은 수차 기술한 바와같이 기업의 조직과 적용 그리고 적용시스템으로 구분된다. 이에 따라 OrgIS 기본 모형 (version)의 주된 임무는 효율적인 조직 개발과 *Fachkonzept* 의 개발 (이에는 data 모델화와 조직의 구체적인 적용이 포함된다) 그리고 적용시스템 개발에 있다고 할 수 있다.

3.1. 조직의 개발 (organization development)

OrgIS 의 기본임무는 기업의 조직 또는 기업의 조직에 관여되는 모든 정보 시스템을 컴퓨터 지원하에 모델화하고 이를 분석하며 계속적인 통제를 통하여 단계적으로 발전시키는데 있다. 이를 다른 말로 표현하면 기본시스템의 관점에서 본 기업조직과 그에 관여되는 정보시스템을 지배하는데 있다고 할 수 있다. 이러한 개발작업은 대상기업의 management 층 뿐만아니라 system configuration 의 계획과 통제를 맡고 있는 부분부서 또는 조직과 정보시스템을 전담하는 중앙 전산 부서의 상호협력을 통해서 이루어져야 하기 때문에 OrgIS 의 조직개발은 담당분야의 부분적인 작업을 통해서도 가능하다. 조직개발의 이같은 본업화 과정에서는 사용자의 요구 (user requirement) 가 주요변수로 작용하게 되는데 이상적인 정보시스템 구성을 위해서는 이 모든 요구가 충족 되어야 한다. 이를 통해서만이 개발대상이 되는 process 와 구성조직 그리고 system configuration 이 하나의 집약된 시스템으로 화하게 된다⁶⁾. 그러나 사용자의 요구는 OrgIS-project 와 지속적인 OrgIS 개발행위의 최종적인 책임이 대상분야의 지휘부에 있을때에만 반영될 수 있다는 사실을 유의해야만 한다.

OrgIS 에 의한 조직개발 작업에 있어서는 도입부분과 실지작업부분 (operation) 으로 나누어 생각할 수가 있다. 도입부분에서는 기존의 process 와 구성조직 그리고 기존의 system configuration 분석을 통하여 organization data model (OrgDatMod) 의 관점에서 최초로 OrgIS-data base 가 구상되며 포괄적인 variance 에 관한 전반적인 연구분석이 이루어 진다⁷⁾. 이 연구분석 결과를 토대로 하여 OrgIS-data base 에서 관리되는 target concept 의 기본모형이 설계된다. OrgIS 의 도입부분은 project 의 성격을 띄고 있으며 project 의 성격에 부합되게 조직되어질 수 있다. 도입부분에 이어서 조직의 근간을 이루는 OrgIS 실지작업부분에 들어가게 되는데 작업부분의 과정은 보통의 management process 나 business process 와 흡사하

6. 조직과 그의 정보시스템의 집약적인 개발의 필요성에 관해서 예를들어 (OST) 에서 강조하고 있다.

7. 전통적인 system analysis 의 방법 (DEM, MCM, MAR 참조)은 데이터 분석과 기능분석에 치중했지만, OrgIS 에 있어 이 분야는 process 조직에서 담당하며, OrgIS 를 이용할때 특히 구성조직과 process 조직, system configuration 의 집약적인 분석과 모델형성을 통해서 전통적인 system analysis 의 분석방법이 할수 없었던 variance (예를들면 작업능력의 한계) 를 식별할 수 있다는 장점이 있다.

다. 작업부분에서는 먼저 작업을 통해서 얻은 자료와 도입부분의 결과가 끈임없이 비교검토되어 기재된다. 이 비교검토 기재과정은 조직의 기존 상태에 관한 것 뿐만아니라 target concept 로서 구상되어지는 version 에 대해서도 이루어 진다. OrgIS 의 작업부분은 따라서 조직의 기존상태와 조직 개발계획을 시간적인 관점에서 연속계획의 원칙에 따라 모델화하고 결과보고나 기존상태를 분석하는 project 의 결과 또는 target concept 개발과 밀접한 관계를 맺고 있는 조직의 기존상태의 단점을 보완할수 있도록 조직 data 모델로서 OrgIS-data base 에서 관리되어질 수 있도록 구성되어 있다. 그러나 이를 위해서는 먼저 조직의 기존 데이터와 목표 (target) 데이터를 포함하는 OrgIS-data base 가 project 의 형식에 의해서 설계되어야 하며 이어서 지속적인 기업조직의 한 부분으로서의 OrgIS-organization 이 기업내에 설치되어야 한다.

조직 concept 개발단계에 이어서 실질적으로 조직이 변형되는 단계가 따른다. 조직의 변형 또한 concept 개발시의 구조와 같이 실질적인 면과 시간적인 면으로 구분되어 이루어질 수 있다. 그러나 모든 세부 concept 는 조직변형의 부분 project 로부터 전체적인 시스템을 유도해낼 수 있도록 가능한 한 project 형태로 구체화 되어야하며 이를 위해서는 project 조직이 부수적으로 확정되어야 한다. 일반적으로 조직변형을 통해서 작업처리의 자동화율이 더 높아지기 때문에 조직변형의 근본적인 사실은 계속적인 새로운 적용시스템의 도입과 새로운 적용시스템에 상응하는 구성 조직과 process 조직 그리고 system configuration 에서의 수반시스템을 새로이 구성하는데 있다. 새로운 적용시스템의 도입은 개괄적으로 자료의 구성, 시스템의 설치, 검사 그리고 실제운용의 단계로 나뉜다. 그러나 사실은 software 분석이나 선별 그리고 개발단계가 이보다 선행되어야 하는데 software 의 분석이나 선별은 시중에 공급되어지고 있는 software 에 한하며 software 개발은 자체적인 프로그래밍이나 위탁을 통해서 이루어 질수 있다.

도표를 통해 설명할때에 조직은 수직선 상에 세워진 세 기둥 즉, process 조직, 구성조직, system configuration 과 수평으로 쌓여진 각각의 경영정보시스템으로 이루어져 있으며 여기에서 같은 기능수행자는 여러 정보시스템에 동시에 관여될 수 있다. 이러한 조직 concept 의 준비로부터 연구개발되어 실지로 기업조직의 변형으로 연결되는 조직개발의 양상은 첫째로 개개의 정보시스템 개발을 지향한 것이다. 여기에서 각기 다른 정보시스템들은 하나의 집약된 결합체를 형성하는데 이는 다시 상위의 정보시스템으로서의 역할을 한다. 따라서 정보시스템은 특정의 한정된 범위에서 그리고 집약단계에 상응하는 정도에따라 단계별로 actual concept 와 target concept 로서 개발되며 그 결과는 조직 data 모델로서 OrgIS-data base 에 입력되어 관리되고 이용되어 진다. 여기에서 OrgIS-data 모델은 object type net 로 표시되는데 object type net 상의 basic object type 은 구성조직과 process 조직 그리고 system configuration 의 기본개념을 나타내며 relation type 은 이를 형성한 mechanism 과 함께 조직상의 관계와 configuration 기술상의 상호연관관계를 모델화 한다. Object type net 는 구성조직과 process 조직 그리고 system configuration 에 상응하는 부분 net 와 이를 연결시켜주는 relation type 으로 나뉜다. 특히 후자의 부분 net 는 구성조직과 process 조직사이의 상관관계를 형성하는 소속관계, process 조직과 system

configuration 을 연결하는 기능과 object type 의 시험적인 implementation 그리고 컴퓨터 작업장뿐만 아니라 system configuration 과 구성조직을 연결해 주는 최종 컴퓨터장치를 연결해준다.

3.2. Fachkonzept 의 개발

Fachkonzept 는 개발대상이 되는 적용시스템의 구체적인 적용개념과 data 모델을 포함하며 이는 적정한 표준 software 의 평가와 새로운 software 의 구입을 결정하는데 기본정보를 제공할 뿐만 아니라 software 개발계획과 실행을 위한 기본이 된다. Software 를 구입할것인지 새로이 개발할것인지에 관한 결정은 이 기본적인 정보에 바탕을 두어야 함은 물론 더 나아가서는 보완하고 변형시킬수 있느냐의 여부 또한 고려되어야 한다. Fachkonzept 개발을 위해서 OrgIS 에의해 지원가능한, 이에 부합되는 project 조직이 마련되어야 한다.

Fachkonzept 의 개발은 먼저 고려되는 적용시스템의 잠재적인 적용을 위해 concept 를 개발하는데 있다. 잠재적인 적용이라 함은 대상전문분야에 있어 컴퓨터지원하에 이루어지는 기능의 전형적인 process 조직과 시스템화된 object type 그리고 시스템지원 가능한 사용자-interface 를 의미한다. Application concept 는 이에 상응하여 전술한 바 있는 기본개념 이외에 정보처리기능의 투입 산출 관계, 정보처리상의 작업순서, 하부기능들로 구성된 본 기능의 구성 그리고 object type 을 나타내는 요소로 구성된, 복합 object type 으로 이해되는 사용자-interface 를 모델화 한다. 여기에서 그 구성이 언제나 종적일 필요는 없으며 정보흐름도가 상이한 집약단계의 기능을 표시하며 이 기능 (상급기능)이 정보흐름도에서 다시 분리되어질 수 있도록 구성되면 된다. 잠재적인 적용은 조직개발의 target concept 를 위해 이미 존재하고 있는 하나의 또는 많은 실질적인 process 조직과 관련되어 있으며, 이는 또한 일반적으로 잠재적인 적용의 전문화 (spezifikation)로 표시되어질 수 있는 시스템의 예상되는 실질적인 적용과 부합된다. OrgIS 는 잠재적인 적용을 위해 소위 application model (AppliMod) 이라고 하는 data 모델과 그의 관리분석을 위한 특수기능을 가지고있다. 이 적용모델은 다시 process 조직과 흡사한 object type net 와 일치하며 또한 process 조직과 잠재적인 적용을 위한 양 OrgIS-data 모델사이에는 process 조직중 구체적으로 어디에서 잠재적인 적용이 현실화될 수 있는지를 나타내는 관계가성립된다. OrgIS 는 계획중에 있거나 기존 적용시스템의 잠재적인 적용과 정보시스템의 process 조직내에서의 구체적인 적용을 모델화하고 분석함으로써 Fachkonzept 개발의 한부분으로서의 적용개발을 지원하게된다.

어느 한 시스템의 application concept 는 주로 시스템지원 가능한 정보흐름 또는 communication 흐름도를 포함하고 있기 때문에 아직은 data 모델의 관점에서 본 개별 object type 이 어떻게 구성되어 있는지는 알수 없다. 그러기 때문에 Fachkonzept 를 완벽하게 구상하기 위해서는 적용의 data 모델이 새로이 정의 되어야 한다. Fachkonzept 개발은 따라서 구체적인 적용 개발과 이에 관련된 data 모델화를 통해서 이루어진다. 이 모델화 과정의 결과는 OrgIS-data base 에서 여러가지 각도에서 관리되어진다. 모든 잠재

적인 적용모델이 object data 차원의 적용모델로 분류 되는 반면 적용 data 모델은 OrgIS 의 meta data 차원의 data 모델로 분류되며 적용 data 모델 자체는 OrgIS 의 object data 모델로서 구체화된다. 이는 OrgIS 에 의해 개발되는 모든 *Fachkonzept* 에 뿐만아니라 OrgIS 자체의 *Fachkonzept* 에도 적용된다. OrgIS 의 개별 application concept 는 도입부분과 작업부분의 잠재적이고 시스템지원 가능한 process 조직으로 정의되어지며 이런 개념으로 정의된 사실은 object data 차원의 적용 data 모델에서 찾아볼 수가 있다. 반대로 조직 data 모델의 모델은 이미 meta data 차원의 data 모델에 포함 되어있으며 개별 조직 data 모델은 object data 차원에서 구체화되어진다.

3.3. 적용시스템 개발 (application system development)

Fachkonzept 는 표준 software 의 분석과 선택을 위해서 뿐만아니라 개별 software 의 구상과 implementation 그리고 software 를 개발할 것인지 구입할 것인지를 결정하기 위한 전제 조건이 된다. 자체개발 이나 외부의 도움으로 이루어지는 적용시스템 개발의 경우에 있어서 *Fachkonzept* 는 시스템개발을 담당하는 engineer 에게 있어서 필수적인 것이다. 시스템의 자체 개발의 경우에 있어서는 *Fachkonzept* 개발은 기업의 중앙전산부서나 담당 전산부서 관할하에 이루어 지며 개괄적으로 시스템 구상과 시스템 구체화 과정으로 나누어 진다⁸⁾. 시스템 구상과정에서는 프로그램 시스템과 data base 가 구상되어진다.

적용시스템 구상은 프로그램 시스템과 data base 를 특수화 (spezifikation) 시킴을 의미하며 시스템으로 구체화 시키는데 필요한 조건들을 충족시켜야 한다. Data base 는 *Fachkonzept* 의 data 모델을 기본으로 구상되며 기존의 data base 시스템을 통하여 모든 요구를 완전히 충족시킬수 있도록 구상된다. 여기에서 meta data 차원에서 OrgIS 에 의해 관리되고 object data 차원에서 구체화 되어지는 application data model (*xDatMod*) 은 data base design 의 모든 특성을 받아들이기 에 부족함이 없도록 구상된다. 적용 data 모델은 그리하여 OrgIS 가 *Fachkonzept* 의 모델화 뿐만아니라 적용시스템 개발을 위한 발전된 data base design 을 관리분석할수 있는 초석으로서의 역할을 담당한다. 프로그램 시스템 구상을 위해서는 먼저 *Fachkonzept* 가 구상되고 data base 가 design 되어야 하며 modul 로 구성된 프로그램구성과 구성요소 그리고 모든 프로그램의 통제구조가 관리분석될 수 있을 정도로 OrgIS 에 의해 지원을 받는다. 소위 통제구조모델 (*KontStruMod*) 이라 불리는 프로그램 시스템의 관리분석을 위한 이 모델 또한 기본개념과 그들 사이의 관계를 나타내는 object type net 로 표시된다. 이를 통해서 implementation 규정의 의미에서 프로그램의 통제구조와 시스템 구성이 구체화되어 질수 있으며 여기에서 design 의 세부화 정도는 자유로이 선택할수가 있다⁹⁾. 통제구조모델과 적용 모델 사이에는 구체적인 개별

8. Software 개발과정의 각 단계별 구조와 이에 사용되는 방법과 tool 의 준비는 software engineering 분야의 대상이 된다 (FAI, RAD 또는 CASE : Computer Aided Software Engineering 참조).

9. 통제구조와 흐름을 설명하기위한 데이터 모델은 절차적 프로그래밍언어

application concept 의 어떤 기능이 구체적으로 어떤 프로그램 prozedure 에 의해 수행되는지의 연관관계가 성립하며 이를 통해서 적용시스템의 기능이 정해진다.

적용의 data 모델은 이 모델을 이미 *Fachkonzept* 를 통해서 기본적인 사실로서 정착시키고 적용시스템구상을 통해서 완성 시키는 중요한 역할을 담당한다. *Fachkonzept* 는 이 이외에도 잠재적인 그리고 시스템지원 가능한 process 조직과 그의 적용시스템 design 상의 기능을 내포하여 이를 수행하는 프로그램의 구성과 통제구조를 특수화하고 통제구조모델로서 dokumentation 되어지는 구체적인 개별 application concept 를 포함하고 있다. 일반적으로 *Fachkonzept* 개발단계 다음으로 적용시스템 개발과 표준 software 의 분석과 선택과정이 뒤따른다. 표준화된 적용시스템이 이의 구입을 결정할 수 있을 정도로 *Fachkonzept* 의 요구를 충족시킨다면 표준시스템 뿐만 아니라 구체적인 application concept 또한 기존의 *Fachkonzept* 를 완벽하게 충족시킬 수 없기 때문에 software 분석결과를 주장할 수가 있다. OrgIS 는 그것이 표준시스템의 구입이든 개별적인 개발이든 모든 적용시스템을 위해서는 구체적인 application concept 와 적용 data 모델을, 설치된 적용시스템의 핵을 이루는 정보시스템을 위해서는 기업의 포괄적인 조직 data 의 부분인 특수 조직 data 를 가진다. 적용시스템에 있어 개별적인 해결의 방법을 택하는 경우에 있어서는 이 이외에도 통제구조를 별도로 갖는다. OrgIS-data base 에서는 그러므로 모든 정보시스템을 위한 조직 data 이외에 적용 data 모델과 모든 구체적인 개별 적용 concept 그리고 경우에따라서는 적용시스템의 통제구조가 object data 또는 meta data 형식으로 관리된다.

4. OrgIS 의 진보된 모형 (version) - meta data 분야의 적용

기업 전체 또는 기업의 부분에 관한 조직은 OrgIS 의 기본시스템을 이루며 이는 구성조직, process 조직, system configuration 그리고 이들 세 구성요소 사이의 관계로 구성된다. 조직은 그러나 기업과 기업의 해당부분에서 이루어지는 작업을 지원하는 정보시스템의 총체로도 이해되어질 수가 있다. 이 정보시스템의 총체가 기업조직 전체를 포괄하는 집합으로 이해되어질 때 조직의 각부분들은 최소한 어느 한 정보시스템에 영향력을 행사하며 일반적으로 인적 기계적 조직기능의 수행자는 동시에 여러 정보시스템에 관련되어 작업을 담당한다. 그래서 조직은 OrgIS 기본시스템의 다른 한 표현방법이라 할 수 있다. 모든 정보시스템은 자신의 기본시스템으로부터 data 모델을 형성하는데, data 모델은 object type net 즉, data 모델에 소요되는 모든 object type net 가 구체적으로 현실화된 총체로 이해되어진다. 이는 적용

에 바탕을 두고있다 (SET). 통제흐름모델을 각기 다른 세부화정도에 따라 달리 표현함으로써 OrgIS 는 (WIR) 에서 제안한 것과같이 단계별 세분화의 원칙을 뒷받침한다. 통제흐름모델은 전술한바 있는 object type net 와 같이 Semi-hypergraph 로서 표시될 수 있기때문에 Jackson-Diagrammen (JAC) 에서 처럼 데이터와 통제흐름에 관한 하나의 통일된 모델형태와 표현형태가 구상될 수 있다. 그러나 Jackson-Diagramm 방법과는 달리 OrgIS 에 있어서는 잠재적인 process 조직을 통해서 프로그램 시스템의 도입 또한 이에 반영되고 모델화된다는 차이가 있다.

data로서 정보시스템의 한 구성요소가되며 동시에 정보시스템의 기본시스템을 나타낸다. 예를 들어 OrgIS의 각각의 data set (*OrgISDB*)는 정보시스템의 한 주요부분이며 이는 구체적으로 기존상태와 target 상태에서의 조직모형을 나타낸다. 이와같은 논리는 기업의 조직에 속하는 모든 정보시스템 (*xIS*)에 적용되며 또한 이를 위한 적용 data 모델 (*xDatMod*)과 이에 상응하는 정보시스템의 data set (*xISDB*)가 정의된다.

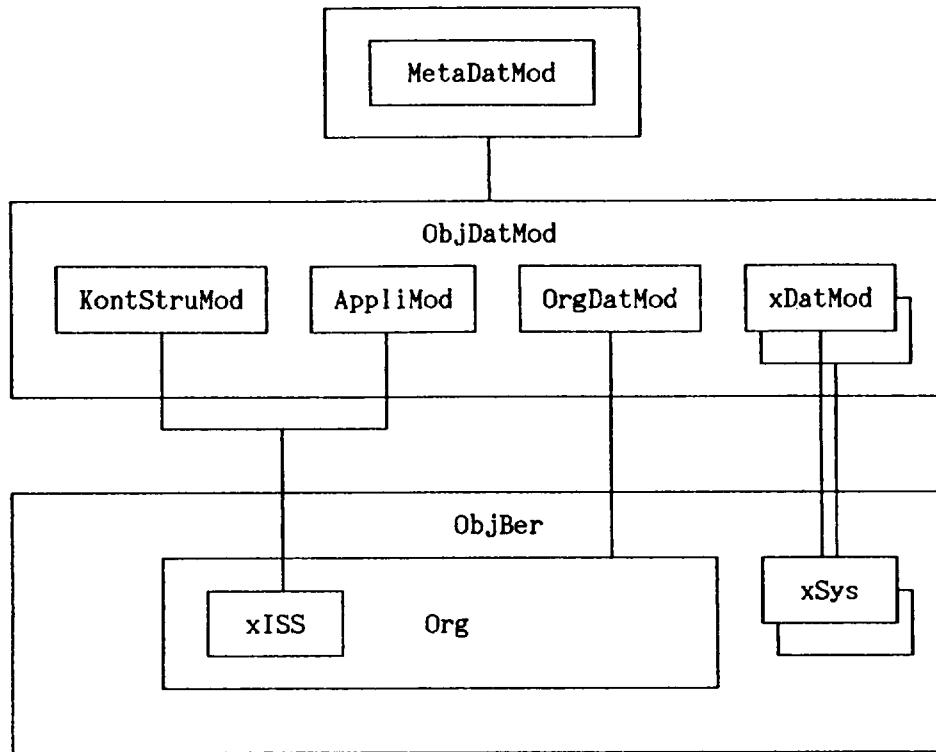
컴퓨터지원하에 이루어지는 모든 정보시스템은 그의 적용시스템의 핵심부분이라할 수 있는 정보시스템 software (*xISS*)를 소유하는데 이는 구체적으로 정보시스템이 자체개발된 software에 의해서 지원되느냐 또는 표준 software를 구입하여 이를 이용함으로써 지원되느냐를 나타낸다. 자체개발을 통한 해결이든 또는 표준 software의 구입을 통한 해결이든 두 경우에 있어서 시스템지원 가능한 process 조직을 이상적으로 나타내는 구체적인 application concept가 먼저 개발되어야 한다. 개별적인 software가 기업내부에서 자체적으로 개발되고 관리되어지는 경우에는 이 이외에 이에 상응하는 특수통제구조가 요구된다. 이 경우에 있어 구체적인 application concept나 통제구조는 모두 data set로서 OrgIS에 의해서 이루어진다. 이를 위해서는 그러나 OrgIS에 잠재적인 적용과 정보시스템 software의 통제구조에 적용될 수 있는 이론적 토대를 마련할 수 있는 data 모델이 정의되어야 한다. 여기에서 이론의 모델이라 함은 이론적으로 가능한 모든 해석을 의미한다. 이 data 모델은 application data model (*AppliMod*) 또는 통제구조모델 (*KontStruktMod*)이라 불리운다.

OrgIS의 대상분야는 조직 즉, 기업의 모든 정보시스템과 정보시스템의 모든 기본시스템을 포함한다. 여기에서 OrgIS는 특수한 정보시스템으로서 그의 기본시스템은 조직과 일치된다. 그리고 기업의 다른 기본시스템들은 기업의 기본기능과 관련된다. 예를 들면 구매 생산 판매시스템 등과 같은 기업의 기본기능이 그 예가 될것이다. 이 모든 object 분야에서 OrgIS는 소위 data 모델을 구성한다. 대표적으로 조직 data 모델, 적용모델, 통제구조모델 그리고 이들 모델사이의 관계가 OrgIS에 속하는데, 언급한 모델과 그들 사이의 관계가 OrgIS의 이론적인 핵심을 이루기때문이다. object 분야의 다른 data 모델 즉, 모델의 실체가 기업의 기본기능을 수행하기 위한 기본시스템으로 나타나는 data 모델은 *Fachkonzept* 개발과 적용시스템개발의 범위내에서 OrgIS-사용자에 의해 구상되고 세분화되어야 한다. 언급한 모든 data 모델은 해당분야의 OrgIS-data base의 data set에 해당되며 이 data set를 통해서 정보시스템과 그의 기본시스템의 기존상태와 계획된 상태가 묘사되어진다.

언급한 전형적인 모델뿐만아니라 다른 적용모델을 포함하는 OrgIS 대상 분야의 모든 data 모델은 OrgIS의 발전된 모형을 통해서 관리되고 구체적으로 사용되어지는 meta data로서 설명되어질 수 있다. 이를 위해 OrgIS-data base는 다시 OrgIS의 한 부분을 이루는 data set를 소유하는데, 이 data set는 meta data set (*OrgISMetaDB*)라 불린다. 이 data set는 조직 data 모델, 적용모델, 통제구조모델과 이들 모델사이의 관계를 설명해 줄뿐만아니라 이에 더 나아가서 정보시스템을 위한 *Fachkonzept* 개발과 적용시스템개발의 차원에서 이루어지는 모든 적용 data 모델을 설명해 준다.

그러므로 OrgIS 대상 분야의 모든 data 모델은 meta data 를 통해서 설명되어지고 분석되어질수 있다는 원칙에 도달하게 된다. 그러나 이를 위해서는 OrgIS 가 data 모델들을 포괄하는 상위 data 모델, 소위 meta data model (*MetaDatMod*) 을 소유해야 한다는 조건이 필요하다. 이 meta data 모델은 어느정도 data 모델을 이론적으로 뒷받침 해주며 이는 또한 data 모델이 구체화되는 것을 나타내는 object type system 으로 표시된다. Meta data 모델 또한 전형적으로 OrgIS 에 속하며 이를 통해서 OrgIS 의 이론적인 중요부분이 보완되며 완성된다. 예를 들어 meta data 모델을 통해서 OrgIS 에서 이루어 지는 자료관리자 (data manager)의 여러 다른 기능이 수월하게 이루어질 수 있는 토대를 마련한다. 이러한 meta dat 모델의 기능은 기업 대상분야의 복잡한 자료들을 추상적인 meta data 분야의 차원에서 통일시키고 규칙을 정하며 이를 조정할수 있는 역할을 담당하기 때문에 집약적인 조직정보시스템 개발을 위해서 큰 의미를 갖는다. OrgIS 의 발전원 모형에 나타나는 모든 정보시스템의 적용 data 모델이 meta data 차원에서 관리되어 질수 있다는 가능성은 조직정보시스템의 기능을 완벽하게 해주는 동시에 기업의 모든 경영정보시스템을 위한 조직개발, *Facchkonzept* 개발, 적용시스템 개발이 언급한 이 조직정보시스템에 용화되어 포함될 수 있도록 해준다.

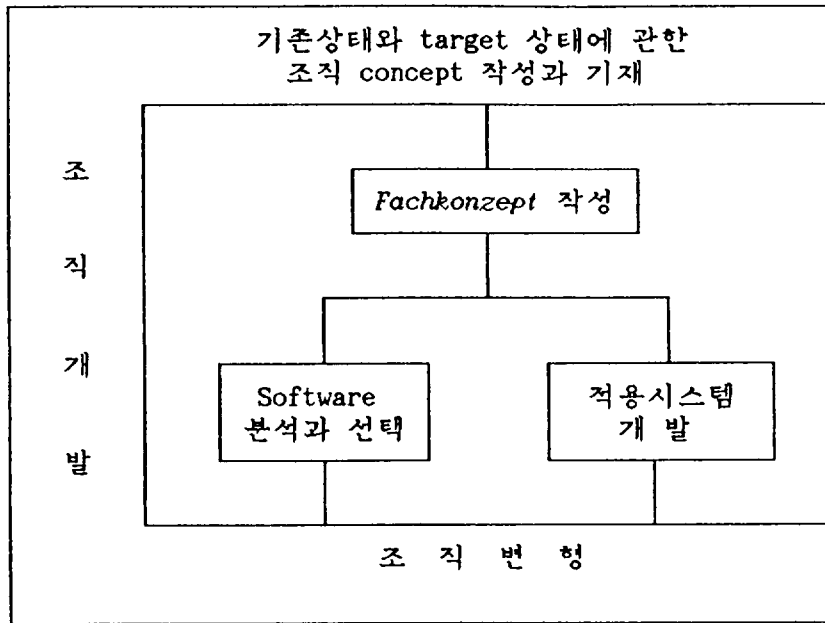
Meta data 모델은 모든 data 모델을 설명해 줄 뿐만아니라 그 자체가 data 모델로서의 성격을 띄고 있기 때문에 meta data 모델은 스스로 모델화 되어질 수 있어야 한다. 그렇기 때문에 meta data 모델의 object type system 은 이 시스템이 현실화되는 구체적인 시스템을 통해서 스스로 정의되어야 하는것이다. 이는 OrgIS 에서 meta data 모델을 나타내며 특히 다른 object type system 으로 표시된다. 이상을 종합하여 볼때 OrgIS 를 설명하기위한, 세 차원으로 분리된 다음과같은 도식을 생각해 낼수 있다. 상단 부분에는 meta data 모델과 meta data set 가 놓이며 중간부분은 모든 object data 모델과 그에 상응하는 object data set 를 포함하고 있으며 마지막 하단부에는 모든 대상분야 즉, 조직을 이루고있는 모든 정보시스템과 그의 기본시스템이 놓인다.



도 4 : OrgIS-Schema

5. 적용시스템 개발의 OrgIS 에의 도입

OrgIS 에서 기본이되는 것은 기업의 정보시스템이 기업조직의 집약적인 중요 부분으로 이해되어 진다는 사실이다. 이러한 견해는 이미 OrgIS-적용단계에서 구체적인 방법으로 나타나고 있다. 조직정보시스템을 기본으로 이루어지는 조직의 분석과 개발작업은 일정한 형식에 따라 진행된다. OrgIS 의 진행모델은 *Fachkonzept* 개발이나 표준 software 분석과 선택 그리고 적용시스템 개발과 같은 주로 전산과 관련된 작업들을 일괄적으로 조직개발의 범주로 분류하여 차후의 정보시스템 작업자의 작업범위에 예측시키는 모델이라 특징지워 말할 수 있다. 경영진의 소속과 책임하에 이루어 지는 조직개발은 조직의 기존상태의 분석과 새로운 target concept 개발로 부터 시작되어 진다. 정보기술상 그리고 communication 기술상의 기본구조를 변화시키지 않고서도 조직상의 진보를 꾀할 수 있는 이 두 작업의 결과는 조직 data 로서 OrgIS-data base 에서 관리되며 target-process 조직을 통해서 *Fachkonzept* 개발을위한 기준척도를 정한다.



원본에는 조직개발의 전단계를 집약시킨다는 의미에서 모형을 \square 이 아니라 그리스문자 Σ 로 표기함.

도 5 : OrgIS - 진행모델

Fachkonzept 개발의 임무는 실현 되어져야할 target-process 조직을 위해 필요한 정보시스템 software 의 구체적인 application concept 와 data 모델을 구성하는 것인데 이를 위한 software 에 있어 표준 software 를 구입할 것인지 또는 자체적으로 개발할 것인지에 관한 문제는 아직 고려대상이 되지 않는다. *Fachkonzept* 개발의 결과 또한 OrgIS-data base 에서 관리되어 지는데, 구체적으로 잠재적이고 시스템지원 가능한 process 조직으로 이루어지는 application concept 는 object data 로서 OrgIS-data base 에 이를 위해 마련된 부분에 분류 저장되는 반면 적용 data 모델은 OrgIS-data base 에서 meta data 를 통해 설명 되어진다. *Fachkonzept* 는 표준 software 의 분석과 선택 그리고 적용시스템 개발을 위한 기준척도를 형성한다. 표준 software 의 분석과 선택 그리고 적용시스템 개발작업은 중앙 전산부서나 부분 전산부서의 책임하에 이루어진다. 표준 software 의 분석과 선택작업의 결과로부터 표준 software 를 구입하는 것이 유리하다라는 결정이 내려지는 경우 software 분석의 결과가 서류화 (dokumentation) 되기 위해서 표준 software 에 상응하는 개별 application concept 와 data 모델은 계속 기재된다. 그러나 직접개발이나 위탁에 의한 적용시스템 개발이 유리하다라는 결론에 도달한 경우는 시스템으로 현실화 되기 위해서 의무규정이 세워져야 하는데 이 의무규정은 한편으로 data base design 을 위한 적용

data 모델의 개발에 관한 조항과다른 한편으로는 implementation 되어야하는 정보시스템 software 의 구성과 통제구조를 정하는 조항 즉, OrgIS-data base 에서 특히 object data 의 형태로 관리되어지는 시스템 design 을 확정하는 조항으로 이루어 진다. 정보시스템 software 개발의 문제가 표준 software 를 구입함으로 또는 직접개발을 통해서 해결되어지는 정보시스템 software 가 확정된후 대상 부분의 경영진의 책임하에 구체적인 조직변형이 이루어진다. 조직의 변형작업은 근본적으로 새로운 적용시스템의 도입을 의미하며 새로운 시스템에 따른 필요한 구성조직과 process 조직 그리고 system configuration 을 새롭게 변형시키는 작업 또한 조직의 변형작업에 포함된다.

목 차

1. 기본 개념
 - 1.1. 정보시스템과 그의 기본시스템 (basic system)
 - 1.2. Object type, object type net 그리고 object type system
 - 1.3. Object data 모델, meta data 모델 그리고 data 모델의 기본 개념
2. OrgIS 의 기본시스템 (basic system)
 - 2.1. 조직과 그의 정보시스템
 - 2.2. 적용 (application) 과 적용시스템 (application system)
3. OrgIS 의 기본 모형 (version) - object data 분야의 적용
 - 3.1. 조직의 개발 (organization development)
 - * 기존 시스템 분석 (actual analysis) 과 target concept 개발
 - * 조직의 변형
 - 3.2. *Fachkonzept* 의 개발
 - * 적용 방법의 개발 (application development)
 - * Data 모델
 - 3.3. 적용시스템 개발 (application system development)
 - * 시스템 구상 (system design)
 - 데이터 베이스 구상 (data base design)
 - 프로그래밍 (program design)
 - * 시스템으로의 전환
4. OrgIS 의 진보된 모형 (version) - meta data 분야의 적용
 - 4.1. Object 분야의 data 모델
 - 4.2. Object 데이터를 위한 data base 의 generation
 - 4.3. Data 모델에 의존하는 data 관리기능의 generation
5. 적용시스템 개발의 OrgIS 에의 도입
 - 5.1. 기업 data 모델의 개발
 - 5.2. *Fachkonzept* 의 coordination
 - 5.3. 적용시스템의 세부 모듈화 (modularization)

참 고 문 헌

- CHE Chen, P.P (1976) "The Entity-Relationship Model", Toward a Unified View of Data, ACM Transactions on Database Systems, Vol.1, No.1, p.9.
- COD Codd, Edgar F. (1990) The Relational Model for Database Management, Addison-Wesley.
- DAT Date, C.J. (1987) An Introduction to Database Systems, Vol.II, Addison-Wesley.
- DEM DeMarco, Tom (1978) Structured Analysis and System Specification ; Yourdon, New York.
- FAI Fairley, Richard E. (1987) Software Engineering Concepts, Berlin, Heidelberg, New York.
- GUT Gutenberg, Erich (1971) Grundlagen der Betriebswirtschaftslehre Bd.1, Die Produktion, Berlin, Heidelberg, New York.
- JAC Jackson, Michael (1983) System Development, Prentice-Hall
- KOS Kosiol, Erich (1972) Organization der Unternehmung, Wiesbaden.
- LOC Lockemann, Peter (1978) Rechnergestuetzte Informationssysteme, Berlin, Heidelberg, New York.
- MAR Marca, David A.; McGowan, Clement L. (1987) SADT Structured Analysis and Design Technique, McGraw-Hill.
- MCM McMenamin, Palmer (1988) Strukturierte Systemanalyse, Hansa.
- NOR Nordsieck, Fritz (1972) Betriebsorganization-Lehre und Technik, Stuttgart.
- OST Oesterle Hubert ; Brenner Walter ; Hilbers, Konrad (1991) Unternehmensfuehrung und Informationssystem - Der Ansatz des St. Galler Informationssystem-Managements, Stuttgart.
- PAR Parnas, D.L. (1972) "On the Criteria to be Unsed in Decomposing Systems into Modules", Communications of the ACM, Vol.15, Number 12.

- RAD Radice, A.Ronald ; Phillips, Richard W. (1988) "Software Engineering - An Industrial Approach", Vol.I, Prentice Hall.
- RAV Ravi, Sethi (1989) Programming Languages - Concepts and Constructs, Addison-Wesley.
- SCH Scheer, August-Wilhelm (1990) "Modellierung betrieblicher Informationssysteme", Wirtschaftsinformatik. Jg.32, Heft5, S.403.
- ULL Ullmann, Jeffrey D. (1988) Database and Knowledge-Base Systems, Computer Science Press.
- WIR Wirth, Niklaus (1971) "Programm Development by Stepwise Refinement", Communications of the ACM, Vol.14, No.4.