

構造計劃에서의 知識基盤시스템 導入研究  
An application of a Knowledge-Based System  
for Structural Planning

김상철\*                      김홍국\*\*                      이병해\*\*\*  
Kim, Sang Cheol              Kim, Hong Kook              Lee, Byung Hai

ABSTRACT

This study describes an application of a knowledge-based system for a part of the development of an integrated structural design system. In preliminary structural design procedure, most structural design operation are performed by structural engineer's manual method. These lack of systematic operation hampers the effective system integration. By introducing expert system to the structural planning stage, structural engineer can automate structural Planning process of an integrated structural design system for complex design. Engineering data management is receiving increasing attention due to complexity of information necessary for performing structural engineering operations. So, in this paper, we describe a methodology for automating conceptual structural design and developing a knowledge-based system integrated with databases. At the end, we use an implemented example to support our methodology.

1. 序 論

構造物의 設計는 컴퓨터의 급진적인 進歩와 많은 研究努力의 結果로 全構造設計過程의 統合 自動化 方向으로 활발히 進行되고 있다. 그러나, 통합과정의 初期段階인 構造計劃은 設計者의 經驗에 의존하는 複雜성과 기존 프로그램의 限界性으로 인하여, 現在까지는 경험적인 構造設計者에 의해 設計圖面을 肉眼으로 分析하고, 그 結果를 手作業으로 입력데이터를 作成하는 과정에서의 한계성 때문에 시간적인 效率성과 精確성이 缺如되고, 각 설계단계에서 緊密한 協助가 이루어지지 않아 데이터의 흐름이 斷絶되고, 二重作業이 되어 통합구조설계시스템의 具現에 있어서 걸림 돌이 되고 있다. 이러한 문제점을 解決하기 위해 여러 연구진이 새로운 方法을 摸索하고 있으며, [1, 2, 14, 15] 本 研究는 統合構造設計化의 초기연구과정의 一環으로서 知識基盤시스템導入을 통한 構造計劃의 自動化 方法論을 제시하고자 한다.

構造計劃은 多樣한 구조시스템중에서 構造物의 安全性, 機能性, 經濟性을 평가하면서 많은 潛在的인 代案들을 검토하고 물리적인 대안모델들을 시험하여 最適의 시스템을 결정해야 하는 어려운 作業들을 內包하고 있다. [14] 즉, 設計者의 구조계획단계는 구조물의 概念的인 설계 자동화의 研究에서 중요한 部分을 차지하는 것이다. 이러한 作業은 설계문제에서 合理的인 형태를 결정하기 위하여 構造設計過程에서 설계자의 問題解決 方法을 모델링하는 것이 要求되어지며, 專門家の 經驗과 지식을 바탕으로 構造計劃을 시스템화하여, 설계자의 초기 설계단계에서부터 構造的인 문제와 他分野의 相關關係를 同時作業(Concurrent Engineering)화할 수 있는 기초작업으로 設計情報로부터 構造設計에 이르는 과정을 一括處理하여 설계자의 判斷과 正確性을 높여주는데 이 論文의 기초적인 研究目的이 있다.

本 研究의 範圍는 객체지향개념과 知識基盤시스템을 導入, 一般高層 事務所건물의 구조설계과정을 分析, 각 段階에서 발생되는 설계정보를 客體모델링하여, 客體들간의 關係성과 制限事項을 지식으로 표현, 知識基盤시스템으로 構築하며, CDB(Centralized DataBase system)의 領域안에서 하나의 프로젝트를 수행하여 豫備設計段階에서 知識基盤으로부터 生成된 정보를 後續段階로 連結시켜주는 것으로 한다.

- \* 한양대학교 건축공학과 석사과정
- \*\* 한양대학교 건축공학과 박사과정
- \*\*\* 한양대학교 건축공학과 교수

## 2. 知識基盤시스템과 데이터베이스의 考察

### 一 知識基盤시스템

知識基盤시스템은 特定分野의 專門家로부터 獲得한 知識, 能力을 컴퓨터 시스템화하여, 이를 基盤으로 制御, 設計, 計劃, 診斷 등의 기능을 遂行할 수 있는 道具이다. 知識基盤시스템은 推論機關(Inference Engine), 知識基盤(Knowledge Base), 問題表現(Representation of the problems at hand)으로 구성된다. 추론기관은 結論을 만들어내기 위한 문제들의 표현과 지식기반을 사용하기 위한 컴퓨터 프로그램이며, 知識基盤은 知識工學者에 의해 抽出된 事實, 過程, 規則들을 모아둔 것이다. [4]

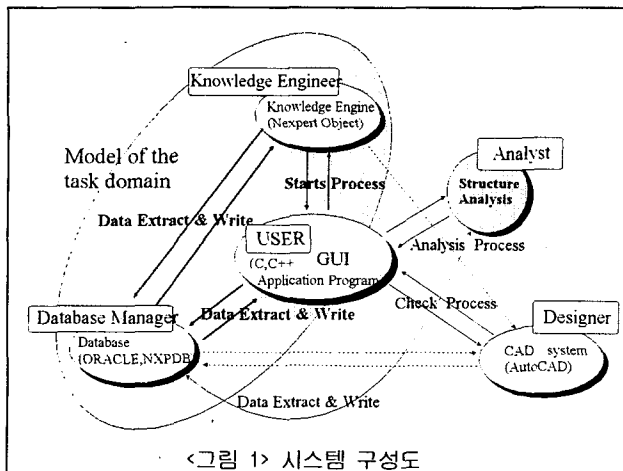
本 研究에서는 知識基盤시스템과 CDB(Centralized DataBase System)의 接續을 위하여 互換성이 높은 汎用 專門家 시스템 構築道具인 NEURON DATA사의 Nexpert Object모듈과 GUI(Graphic User Interface) 개발모듈을 가지고 있는 Smart Elements을 사용한다. [5]

### 一 데이터베이스

最近에 強調되는 엔지니어링 데이터베이스(EDB)는 데이터 交換, 複雜한 관계로 聯關된 엔지니어링 설계정보의 維持, 설계 진행과정의 管理를 목적으로 한다. 대부분은 설계자가 설계과정에서 여러 觀點으로 推論되는 物理的, 技能的, 舉動的 特性을 가지며, 설계과정에서 엔지니어링 客體들은 하나의 狀態에서 다른 狀態로, 하나의 스키마에서 다른 스키마로 계속적으로 增加(進化)되는 動的인 傾向이 있다. 프로젝트의 進行過程을 통해 각 段階에서 大量의 異質性 데이터들을 生成, 維持, 交換하는 것이 必然的이다[6]. 본 연구의 데이터모델링은 客體指向으로 하고, 具現은 本 研究過程에서 쓰여지고 있는 Smart Elements 와의 互換性문제를 考慮하여 관계형 데이터베이스인 ORACLE DBMS와 Flat-file형태인 NXpdb를 사용하였다. [5]

### 一 시스템간의 接續(Interface)

知識基盤시스템을 構築하는데 있어서 지식기반자체만으로는 한계점을 가지고 있다. 따라서 데이터베이스를 통한 시스템의 擴張이 統合化를 推進하는데 있어서 重要한 部分을 차지하는 것이다.



<그림 1> 시스템 구성도

現在의 DB(Database)와 知識基盤의 接續 問題點은:

• 대부분의 知識基盤 시스템은 DB와 統合化가 잘 되지 않았다는 것.

• 統合化는 느린 DB檢索과 全體的으로 制限된 DB의 인터페이스로 인해 방해를 받고 있다는 것.

• 시작점에서의 문제로 DB가 먼저 先行되고 항상 知識基盤시스템이 後續적으로 이어진다는 것.

• DB는 構造物의 應력을 계산하는 것과 같은 알고리즘적인 變換을 포함하고 있어 處理過程을 理解할 수

있는 시스템을 제공해주지만, 知識基盤過程은 專門的인 것보다는 一般的인 變換을 사용하고 抽象的이라는 것.

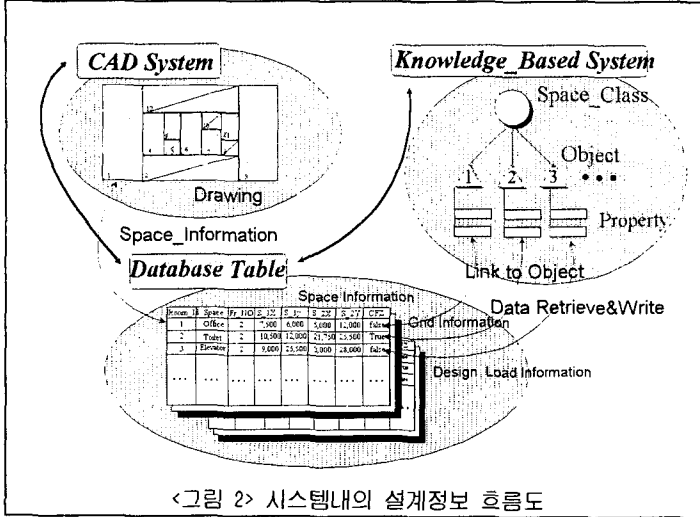
• DB와 知識基盤시스템의 相互關係가 明確하지 않다는 것, 등을 들 수가 있다.

現在의 研究와 開發成果를 볼때 DB와 知識基盤시스템과의 단순한 接續이 아니라 直接 DB로 接近하는 方法을 통하여 특정된 작업은 데이터를 入力하기 위하여 DB를 사용하고 技術的인 諮問은 知識을 獲得하여 知識基盤으로 구축함으로써 速度, 安定性, 데이터형태에서의 可溶性을 제공해 줄 수 있게 된다[7]. Nexpert Object에서 DB자료에 接近하는 方法으로는 DB\_Bridge를 통한 Atomic, Sequential, Grouped Operations이 있으며, [8] 데이터의 處理過程은 클래스내에서 그룹화된 客體들에 의해 이루어지는데 <그림2>에 보여진다.

## 3. 設計情報 모델링

統合化를 目的으로 구조시스템을 決定하는 과정에서 外部環境(DBMS, CAD system 등)과의 接續

을 위해서는 效率인 데이터 모델링이 필요하며, 시스템을 構築하기 위한 사실의 客體모델링과 이들의 相互關係性을 나타내는 知識모델링이 先行되고 설계정보들의 空間인 關係를 分析하기 위한 空間모델링이 이루어져야 한다. 本 研究에서 表現하고자 하는 정보들은 物理인 要素를 가지고 있는 "建物"에 대한 것이며, 設計者는 여러 形態의 建物 혹은 적어도 하나의 建物에 있어서 建물의 用途와 諸般要件에 따라서 여러가지의 구조적인 形態가 나올 수 있다는 事實을 알고 있다. 이러한 形態를 分析하고, 建物에 대한 情報를 組織化하기 위해서 專門家의 經驗과 構造設計規準 등의 構造的, 派生的인 制限事項들은 有效적절하게 知識으로 表現되어져야 한다는 것을 알게 된다. 여기서, 컴퓨터시스템에서 事物들을 表現하기 위해서 심볼을 사용하기에 앞서서 이 方法에서 사용하는 假定들과 問題들을 認識하고자 하는 것이다. [7]



<그림 2> 시스템내의 설계정보 흐름도

— 設計情報의 객체모델링

객체모델링은 豫備設計, 構造解析 및 部材選定段階에서 발생될 수 있는 설계정보자료를 體系의으로 표현하게 되며, 설계정보에 대한 質疑와 論理가 논리적이고 容易하게 되고, 통합시스템내의 中央 데이터베이스의 Schema모델이 된다. 객체, 속성, 클래스, 메소드에 基本을 두고 설계의 정보들을 處理하는 方法은 다음과 같다.

客體와 關係性(Objects and Relation)

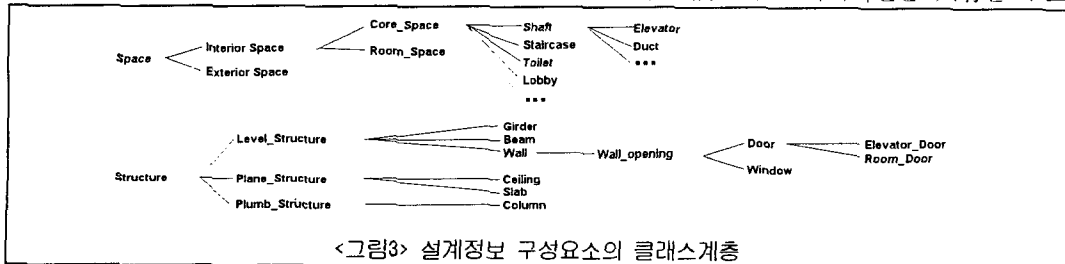
本 研究에서의 客體는 情報의 가장 작은 單位이며, 知識을 表現하는데 있어서 어떠한 情報의 單語를 說明하기 위한 클래스의 例 (Instance)이다. 客體에는 두가지 形態의 客體(動的客體, 靜的客體)로 구분된다. [8]

- 1) 動的客體 (Dynamic Object): 動的인 客體는 推論中에 클래스내에 生成, 消滅되어지는 客體를 말하게 되며 1:M, M:M의 關係를 가지게 된다.
- 2) 靜的客體 (Object): 推論되어지기 이전의 客體로 항상 그 자신의 屬性을 가지고 있어 推論中에서 屬性의 값을 주고 받는 1:1의 關係에서 成立되는 客體이다.

여기서 構造計劃을 위한 物理的이고, 非物理的인 客體를 어떻게 처리하는 것에 대한 方法論이 必要하다. 建物에 대해서 기둥, 보, 바닥등과 같은 수많은 객체가 生成이 된다. 關係性에 대한 文章은 "기둥은 보에 인접해 있다.", 에서 밑줄은 關係性을 나타내는 것이다.

클래스(Class)

위에서 定義된 客體들은 共通된 特性이 있다. 즉, 클래스는 客體들의 모임으로 一般化된 概念이며, 클래스는 시스템의 組織化와 開發을 容易하게 하고 많은 客體들에 대한 管理가 쉬워지고, 階層構造的인 組織化가 된다. "철골기둥은 기둥이다."라는 文章에서 기둥은 클래스가 되고 客體는 철골기둥이 되는 것이다. 上位클래스와 下位클래스상에는 屬性과 操作을 共有할 수도



<그림 3> 설계정보 구성요소의 클래스계층

있는데 이를 "相續性"이라 하며, 推論過程中에 制御를 해주게 된다.

### 메소드와 메시지(Method and Message)

우리가 考慮할 수 있는 表現들의 方法論에 대한 모든 가능한 組合들이 設計者에게 意味가 있는 것일 수는 없다. 다른 프로젝트에서 동일한 점들이 모두 包含될 수가 없기 때문이다. 選擇한 메소드는 "그 方法論이 有用한가" 혹은 "設計作業에서 우리의 要求에 適切한가"에 달려 있게 되는 것이다. 메소드는 推論過程中 어떤 狀況下에서 客體와 클래스, 屬性, 슬롯에 連結되어 一連의 行動을 일으키도록 하는 것이다<sup>[5]</sup>. "Send\_Massege"는 어떤 動作을 일으키는 原因이 되고, 그 動作은 連結되는 클래스 혹은 客體의 메소드를 實行시키고 경우에 따라서는 다른 메소드에 다시 메시지를 보내기도 하며, 메시지가 傳達됐을때 客體나 클래스에 該當되는 오퍼레이션이 없다면 Gate에 의해 차례로 모든 連結되는 規則을 찾기 시작하여 해당 오퍼레이션을 찾아 動作을 하게 된다. 메소드는 클래스 혹은 객체내에 貯藏되며, 여러개의 메소드를 附加시킬 수가 있다. 즉, 推論하는 過程에서 어떠한 狀況에 맞는 메소드에 "Send\_Massege"를 줌으로써 객체, 클래스에 割當된 行動을 하게 되는 것이다. 예를 들면, 構造計劃過程中에 코어計劃에서 전단벽, SRC, 순수 철골중 어느 形態가 適合한가라는 몇 개의 代案이 생길 수가 있으므로 推論過程中에서, 각 시스템에 맞도록 構造配置變更이 이루어질 수 있다는 것이다. <그림7>는 構造計劃을 하는 과정중에서는 코어를 전단벽으로 處理할 경우, 그 狀況에 맞는 부재를 採擇하고 修正하는 메시지의 傳達過程을 보여준다.

### 一 設計情報의 知識基盤 모델링

知識表現을 하기 위한 객체들사이에는 무수히 많은 關係性이 存在하며, 룰(RULE)과 메소드등에서 "If ... Then"으로 각 클래스와 客體에 대한 物理的, 非物理的인 關係性을 지식으로 표현하게 된다. 知識을 표현하는데 있어서 룰(RULE)의 표현은 知識基盤시스템에서 가장 基本的인 推論의 單位이다. 하나의 룰에는 반드시 하나 이상의 條件附를 가지고 있으며, 하나의 假設에서 반드시 모든 條件附를 만족하여야 참이 되고, 條件附가 만족되어질때 實行附에 實行시켜야 할 것이 있을 경우에만 실행이 되어지고 없는 경우에는 그 假設은 단지 참으로만 남아 있게 된다. 이러한 여러개의 룰은 知識群(Knowledge Islands)을 이루게 되고, 여러개의 知識群들은 獨立된 지식을 표현하게 되며, 추론과정중에서 의미적으로 連結이 되어진다.

<p><b>RULE 01 : Rule R_Determine_Long_Direction</b></p> <p>If MAX(&lt;Grid_X_Dir&gt;.Grid_Coor) is less than MAX(&lt;Grid_Y_Dir&gt;.Grid_Coor) Then Determine_Long_Direction is confirmed. And "X" is assigned to Long_Direction.XY Else Determine_Long_Direction is not confirmed. And "Y" is assigned to Long_Direction.XY</p>
<p><b>RULE 02: Rule R_Hypo03_Shear</b></p> <p>If Lobby_Count_Ink is less than or equal to Lobby_Count And &lt;Unsort_Space&gt;.Space is not equal to "Office" And &lt;Unsort_Space&gt;.Space_Node_1x is less than Lobby_First_Point Then Hypo03_Shear is confirmed. And Create Object 'Core_Lobby_Count_Ink  Sort_Core_space  And MIN(&lt;Unsort_Space&gt;.Space_Node_1x) is assigned to 'Core_Lobby_Count_Ink Space_Node_1x ..... And MAX(&lt;Unsort_Space&gt;.Space_Node_2x) is assigned to 'Core_Lobby_Count_Ink Space_Node_2x ..... And Lobby_Count_Ink is assigned to 'Core_Lobby_Count_Ink Space_ID ..... And Hypo04_05_X_Shear is assigned to Hypo04_05_X_Shear And Reset Hypo04_05_X_Shear</p>

RULE01은 건물의 장변과 단변 方向을 推論하기 위한 간단한 例이며, RULE02는 설계정보의 用途 및, 位置를 추론하여 전단벽 코어공간을 規定하기 위한 룰이다.

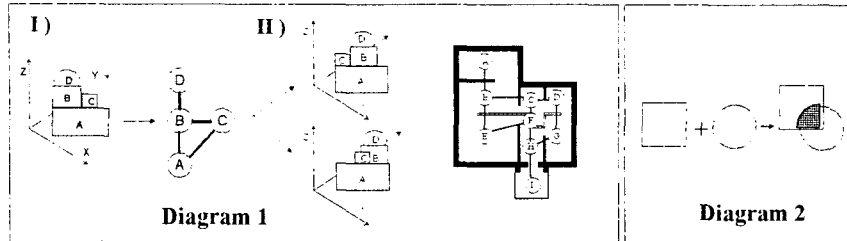
本 研究에서 룰(RULE)의 조건부는 LHS(Left Hand Side)를 構成하고, 假設과 실행부는 RHS(Right Hand Side)로 構成되며, 룰의 評價에서는 後方向 推論(Backward chaining)과 前方向 推論(Forward chaining)에 의해서 同時에 이루어지며<sup>[5]</sup>, 一律的인 과정하에서는 後방향추론이 어떠한 目的에 적합한지 妥當性을 檢討하기는 쉬워도, 構造設計에서 一律的인 설계로는 많은 構造的인 문제들을 檢討하기에 불충분하며, 構

造設計에서의 複雜性을 나타내기 위해 並列的인 형태를 취해야 한다는 問題點이 생기게 되고, 알고자 하는 事實은 다른 知識群(Knowledge Islands)과 連結이 되기 때문이다.

### 一 設計情報의 空間모델링

지금까지 우리는 그래픽적인 심볼을 가지고 設計圖面으로 描寫해 왔다. 設計情報表現의 主要점은 컴퓨터의 주어진 한계점을 가지고 構造計劃을 어떻게 修行할 것이며, 設計圖面으로부터 空間分析을 하고 구조부재의 位置를 決定하는 作業을 하는 것이다. 그러나, 이러한 과정에서 모든 구성요소의 무게, 비용등을 그래픽적으로 表現한다는 것은 어려우며, 실제의 숨겨진 特徵들을 描寫하고 模糊性을 줄이기 위해 文字形態의 索引을 사용해야 한다. 치수와 부분적인 구성요소의 座標에 관한 설계의 幾何學的인 것을 알고 있다면, 설계의 형태를 類推해 낼수가 있는데 예를들면,

隣接한 구성요소의 관계를 <그림4>와 같은 과정을 통해 알 수 있다. [7] <그림4-Diagram1>은 좌표 값을 가진 設計情報의 多樣한 處理方法을 說明해 준다. I)에서 A, B, C는 서로 接해있고, B, D가



<그림4> 설계정보 구성요소의 기하학적인 상관관계도

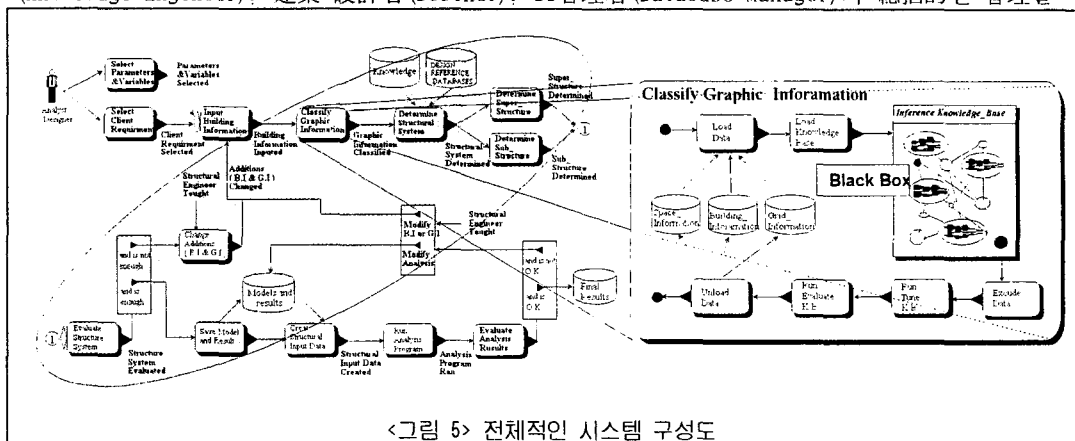
接해있다는 관계를 알 수가 있다. 만약, 표현된 형태가 複雑하면 空間에서의 치수와 위치에 관해서 설계의 表現은 구성요소들간의 많은 幾何學的인 關係性을 가지게 되게 되는 것이다. 즉, II)는 같은 隣接關係를 나타내지만, 위치의 좌표값이 다르기 때문에 다른 形態가 나타남을 보여준다. 이러한 抽象化體系에 대한 概念은 구조계획에서 일반적인 單純表現보다는 같은 형태에 대하여 多樣한 表現으로 생각할 수 있다는 事實에서 중요한 것이다. 이와 같은 概念은 여러設計代案에 대해서 수행되어진 變化의 設計를 表現하는데 있어서 一貫性을 어떻게 유지하느냐 하는 것은 既存의 節次的인 프로그램技法으로는 具現하기가 어려운 것이다.

또, 다른 空間分析에서의 특이한 事項은 設計는 尙상, 選擇, 構成, 整齊된 要素들의 問題點을 감소시키지는 않는다는 것이다. 만약, 우리가 구성요소들을 選擇하고 특별한 方法으로 그것들을 組合한다면 계속되는 구성요소들의 屬性의 단순한 合計와는 다른 屬性들이 存在하는 구성요소를 만들기도 한다(<그림4-Diagram2>). 즉, 하나의 構造部材는 다른것과 相互關係를 가지는데 構造部材의 形態들과 機能이 組合될때 適用되는 것과 마찬가지로인 것이다. 컴퓨터에서 그러한 부분적인 操作을 모델화하더라도 空間에서 그러한 流動性을 모델화한다는 것은 매우 어려운 것이다. 한 예로서 <그림8>의 空間分析段階에서 有用한 지식으로 區分이 되며, 코어에서 로비와 다른 수직코어의 空間分析을 보여준다.

#### 4. 知識基盤시스템의 具現

##### 一 全體시스템 構造

本 研究에서 사용되어진 CAD systems, DBMS, Smart Elements, 外部 應用프로그램의 概要는 <그림1>과 같다. 그림에서 보이듯이 각 시스템은 全體의으로 GUI(Graphical User Interface)를 中心으로 하나로 묶여지고, 각 段階에서 데이터의 交換이 相互 시스템간에 이루어지게 되며, 각 段階에서 결과물은 DB에 貯藏이 된다. 각 시스템에 대한 管理는 자료량이 龍大해 질 경우에는 知識工學者(Knowledge Engineer), 建築 設計者(Designer), DB管理者(Database Manager)가 總括的인 管理를

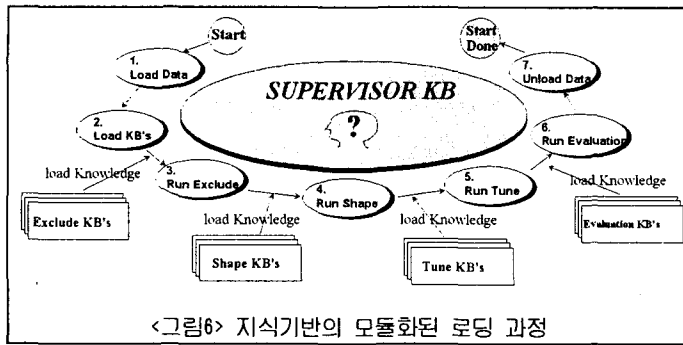


<그림 5> 전체적인 시스템 구성도

한다. 이와 같이 相互 獨立的인 시스템을 構成하는 것은 각 시스템에 관한 管理가 容易해지게 되고, 시스템의 구성요소를 다른 시스템으로 代替시켰을 境遇의 管理, 데이터의 部分修正이 간편하게 이루어질 수 있기 때문이다. 이 시스템構成에 의한 設計情報의 흐름이 <그림 5>에 보여진다.

빗금친 부분은 본 연구에서의 지식기반시스템과 DB의 상호 인터페이스와 지식기반構築에 대한 방법론을 보이기 위한 중점 연구범위를 나타내며, 이에 대한 결과를 CAD시스템에서 그래픽적으로 확인한 후 구조설계정보를 DB로構築하고 있다.

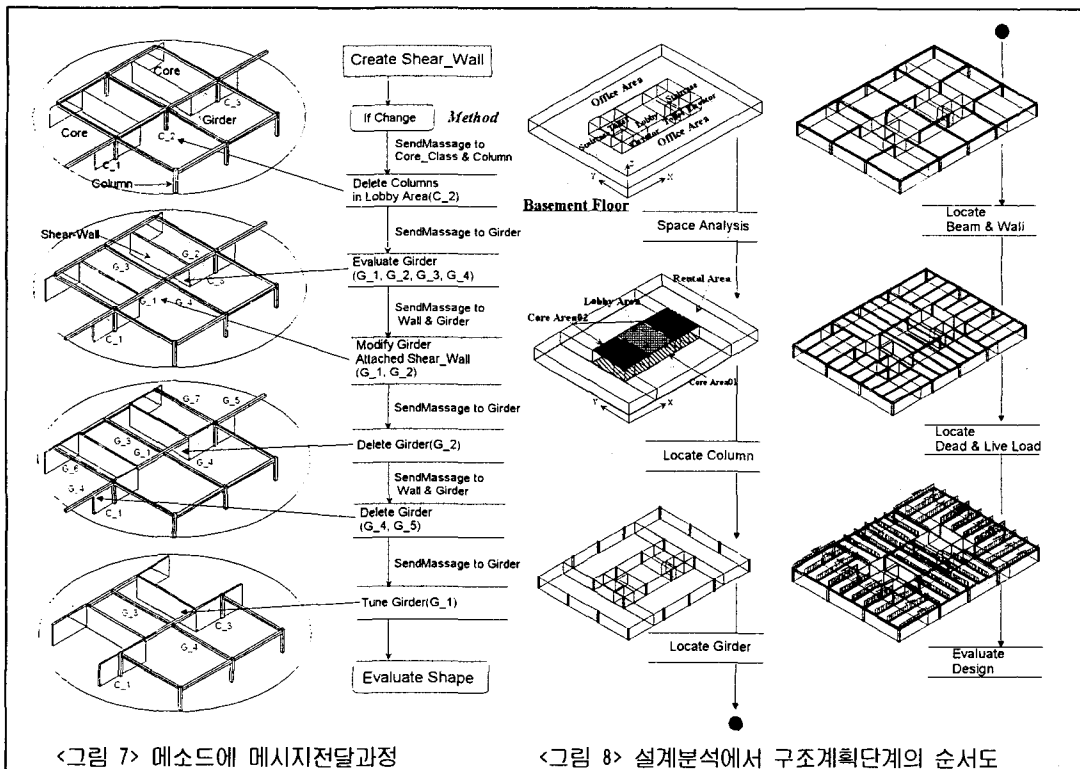
一 知識基盤 시스템 構造



知識基盤을構築하는 과정에 서 知識의 獲得, 增加, 變化로 인해 知識의 양은 龍大해지게 되며, 시스템상에서 지식로딩을 하기 위해서는 問題의 복잡성만을 增加시키게 되므로, 體系的인 문제해결이 可能하기 위해 <그림6> 과 같은 知識基盤의 모듈화가 이루어져야 한다. [6] 이 과정은 人間의 사고판단을 模倣한 것으로, 概括的인 概要를 설명하고, 문제를 간단히 하며, 해결책을 먼저

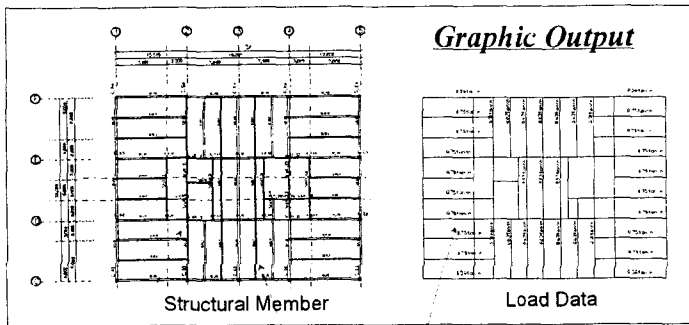
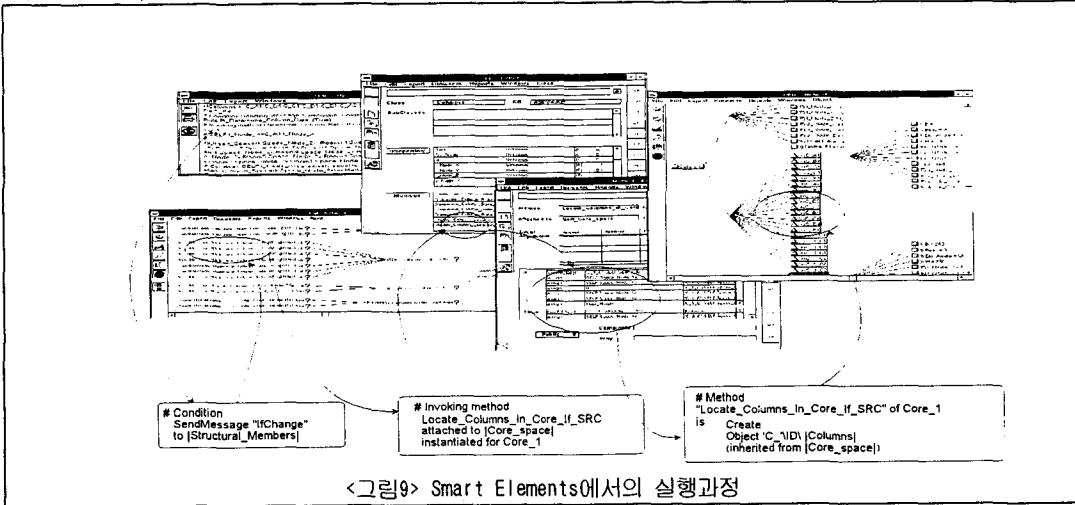
찾게 되며, 그 해결책을 改選시키고, 우리가 만들어낸 해결책에 대한 說明을 가능하게 해준다. 또한, 다른 문제들에 대해서도 適用이 가능해지며, 지식에 대한 修正 및 增加에 따른 지식의 管理가 容易하게 된다.

本 研究에서의 각 知識基盤(KB: Knowledge Base)은 文字形態로 貯藏이 되며, 추론과정중에서 필요한 知識을 로드하게 되고, 추론과정이 끝났을 경우에 知識을 削除하므로서 作業메모리에 대한 管理가 容易해진다. 지식기반의 추론과정에서 필요한 데이터는 DB에서 혹은, 추론과정중에서 使用者로 하여금 入力이 되며, 얻어낸 結果는 DB에 貯藏후, <그림10>와 같이 文字 혹은 그래픽적인 方法으로 使用者에게 結果를 보여주게 된다.



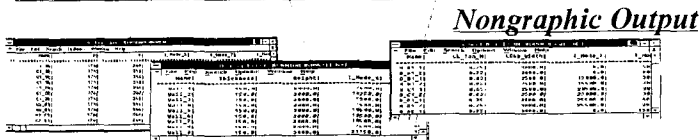
一 프로그램 實例例

實行된 對象建物は 内部코어의 平面을 가진 事務室建物이다. <그림8>은 그리드 情報에서 10.5M, 7.5M間隔의 意味制限을 주었을 때의 推論과정을 나타내며, <그림11>의 代案은 意味制限을 안 두었을 때의 平面構造計劃을 보여준다. 데이터抽出段階 (LOAD DATA단계)에서 건물에 대한 정보는 데이터베이스로부터 空間的인 정보와 派生的인 정보를 抽出 (Retrive)해오는 과정에서 필요한 정보만을 이에 該當하는 클래스에 객체로서 生成이 되며 (그림2참조), 推論이 끝난후의 데이터베이스로의 貯藏은 이와 반대의 經路를 택해서 이루어진다. <그림9>는 실제 본 연구과정에서 具現된 設計情報에 대한 지식의 推論과정과 클래스와 객체가 生成이 되고 메소드가 傳達되는 과정을 보여주며, <그림8>과 같이 空間分析段階를 거치는 동안 기둥의 위치가 決定이 되고 각 水平 시스템과 垂直 시스템의 工法的인 條件에 따라, 가장 適切한 형태를 결정하기 위한 基礎的인 지식이 推論되고, 該當된 메소드를 선택하여 평가단계 (Tune and Evaluation)에서 形成된 구조배치를



평가하고 구조부재에 대한 修正作業을 하게 된다.

<그림10>는 시스템상에서 推論이 되어진 부재와 生成된 荷重分布를 DB의 테이블형태와 AutoLISP으로 CAD시스템 상에 構造圖面으로 檢討段階를 보여준다. 構造解析用 입력데이터는 CDB로 넘겨진 부재정보에서 構造解析에 필요한 정보를 抽出하게 된다.



5. 結 論

以上과 같은 研究過程을 통하여 知識基盤시스템과 데이터베이스를 이용한 豫備構造設計段階의 統合化에 대해서 다음과 같은 結論을 얻었다.

<그림10> 실행 결과 1(문서와 그래픽결과)

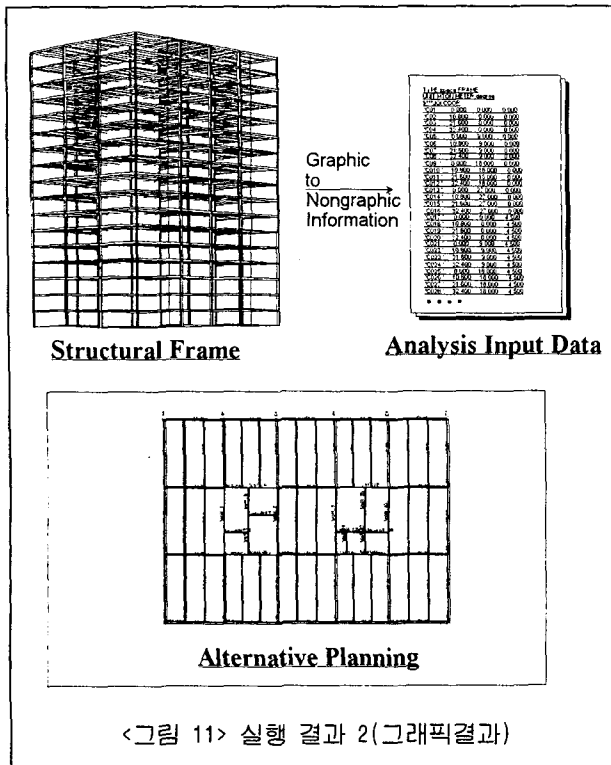
1. 構造設計過程을 表現하고, 多様な 要求와 變化를 가진 設計知識의 抽象化를 객체지향기법과 知識基盤시스템을 통하여 쉽게 知識基盤化할수 있으며, 汎用셀을 사용함으로써 현재 構築된 지식의 變更과 增加에 대한 對處가 容易하며 계속적인 開發이 可能해진다는 것을 알 수 있었다.
2. 多様な 建物形態와 工法을 知識基盤化하기위해 Neural Networks, CBR (Case\_Based\_Reasoning)

등과 接續을 통한 하이브리드 지식기반 시스템(Hybrid KBS)을 構築함으로써 廣範圍한 시스템구축이 可能하다는 것을 알 수 있었다.

3. 시스템의 決定段階에서부터 統合的인 데이터베이스로 後續段階(見積, 施工 등)의 情報를 效率性있게 管理할 수 있다는 것을 알 수 있다.

4. 객체지향 데이터베이스(OODB)를 主軸으로 설계과정에서 엔지니어링 프로젝트에 관계되는 情報를 管理할 필요가 있으며, CAD시스템의 Intelligent화에 대한 研究가 進行되어야 하고, 이러한 시스템이 接續될 때 效率的인 統合化 構造設計시스템을 具現하는데 容易하다는 것을 알 수 있었다.

## 6. 參 考 文 獻



[1] Chee-Kiong Soh, Ai-Kah Soh and Kum-Yew Lai, "Making Data Banks More Useful with Knowledge Embedded Data-Base Systems", J. Comp. in Civ. Engrg.(1993), pp 162-180

[2] Mary Lou Maher, "HI - RISE: An Expert System for Preliminary Structural Design", Expert Systems for Engineering Design(1988), Academic Press, INC: Page 37-52

[3] Marcel de Waard, "Computer Aided Comformance Checking: Checking Residential Building Designs against Building Regulations with the Aid of Computers", (1992) NTIS

[4] HOJJAT ADELI, "An overview of expert systems in civil engineering", Expert Systems in Construction and Structural Engineering, Chapman and Hall(1988), pp45-80

[5] NEXPERT OBJECT Ver 3.0 "Database Intergration Guide", "Reference Manual" Neuron Data(1993)

[6] 1994 NEURON DATA USER GROUP MEETING

[7] R.D COYNE & M.A.ROSEN MAN et al, "Knowledge-based DESIGN SYSTEM" Addison-Wesley Publishing, 1990.

[8] William J. Rasdorf, P.E and TsoJen Wan, "Expert System Integrity Maintenance for the Retrieval of Data From Engineering databases",

COMPUTING IN CIVIL ENGINEERING, PP654-668

[9] H.C Howard, J.A. Abdalla, and D.H Douglas Phan, "A Primitive-Composite Approach for Structural Data Modelling", Journal of Computing in Civil Engineering, ASCE, Vol 6, No 1, January, 1992, pp 19-40

[10] Musen, M., "Automated generation of model-based knowledge-acquisition tools." SIGLunch Presentation. Stanford Univ., 1989

[11] Kwang C. Choi, "An Object-Oriented Historical Cost Information Base to Support Concurrent Engineering", University of California, Berkeley, 1992

[12] 이승창, 이병해, "鐵筋 콘크리트 構造設計에서의 엔지니어링 데이터베이스 構築", 한국전산구조공학회, 추계논문발표집, 제6권 제11호 1993, pp 163-171

[13] 김홍국, 이병해, "鐵骨 構造物의 統合設計시스템", 한국전산구조공학회 추계논문발표집, 제11권 제2호, 1991,

[14] H.Adeli & D.W.Hawkins, "A Hierarchical Expert System For Design Of Floors In Highrise Buildings.", Computer & Structure vol 41, 1991, pp 773-788

[15] Gregory P. Luth et al, "A Formal Approach to Automating Conceptual Structural Design", Engineering with Computers, 1991, pp 79-89