

意思決定支援시스템과 專門家시스템의 統合的 設計에 關한 研究

An Integrated System Design Approach for Decision Support System and Expert System

權 寧 植*

ABSTRACT

Decision support system(DSS) has been expected to be a powerful tool for aiding the decision making processes in business organizations. But it's contribution has turned to be somewhat doubtful.

In this paper, an intergrated systems design apporach is suggested, which integrates DSS and expert system(ES) for the enhancement of performance of DSS, after carefully reviewing both DSS and ES.

1. 序 論

컴퓨터의 發展과 더불어 이를 經營意思決定에 利用하려는 研究가 많이 있었으며 특히 1970 년 대 이래로 意思決定過程의 效果(effectiveness) 를 提高하기 위하여 意思決定者의 意思決定過程과 經營判斷(managerial judgement)을 支援(support)하기 위한 概念에서의 意思決定支援

시스템에 관한 研究가 활발하게 進行되어 오고 있다.

그러나 미국기업의 使用者 그룹(經營者 또는 非電算 부문 管理者)의 2~3% 정도만이 開發된 意思決定支援시스템을 사용하고 있다는 한 調查報考의 結果에서도 알 수 있듯이 아직까지 意思決定支援시스템이 經營活動에 적절히 利用되지 못하고 있는 실정이다(Scott Morton, 1984).

* 東國大學校 工科大學 産業工學科 助教授

따라서 本 研究은 企業組織에서 제대로 활용되어 그 目的과 機能을 다할 수 있는 意思決定支援시스템 設計를 위하여 專門家시스템(expert system)과의 補完統合을 모색하는 것을 目的으로 한다.

本 研究은 意思決定支援시스템과 專門家시스템의 考察을 통하여 새로운 시스템 設計方法을 제시하는 順序로 전개된다.

2. 意思決定支援시스템의 考察

2.1 定義와 特性

意思決定支援시스템(decision support system : 이하 DSS)은 1970년대초 Scott Morton (1971)의 經營意思決定시스템(management decision system) 概念으로부터 시작되어 그 후 많은 學者들에 의해 研究되어 왔으며 일반적으로 非構造的(unstructured) 또는 半構造的(semi-structured) 問題를 풀기 위하여 資料(data)와 模型(model)을 적절히 활용하는 상호작용하는 컴퓨터시스템(interactive computer based system)으로 定義되고 있다.

意思決定支援시스템과 既存의 經營情報시스템(MIS), 經營科學(OR/MS)과는 根本的인 차이점을 갖고 있는데 Keen-Scott Morton(1978)의 定義는 이들의 차이점을 잘 설명하고 있다.

“經營情報시스템은 構造的인 업무(structured task)의 效率(efficiency)을 증대시키는 것이 目的이며 經營科學(OR/MS)은 構造的인 업무에 대한 보다 效果的인 解를 구하는데 利用되며 반면에 意思決定支援시스템은 컴퓨터를 利用하기에 적절하지만 意思決定者의 판단(judgement)이 필수불가결한 종류의 업무(이를 半構造的인 업무(semi-structured task)라 定義함)에 대한 意思決定問題의 效果(effectiveness)를 제고하는 것이 目的이다.”

한편 많은 學者들이 여러 측면에서 意思決定支援시스템에 대한 定義를 내리고 있는데 Bonczek-Holsapple-Whinston(1980) 등은 意思決定支援시스템의 目的이 어떻게 달성될 수 있는가 하는 시스템 구성요소의 측면에서 “言語(language)시스템, 知識(knowledge)시스템, 問題處理(problem-processing)시스템”으로 定義했다. 여기서 言語시스템은 使用者와 知識시스템, 問題處理시스템과의 意思소통 수단을 제공하게 되며 知識시스템은 意思決定支援시스템내에 구현되어 있는 問題領域에 관한 知識베이스이며 問題處理시스템은 言語시스템과 知識시스템을 연계시키며 문제해결과정을 포함하고 있다.

Keen(1980)은 開發過程의 측면에서 “意思決定支援시스템의 使用者, 設計者, 시스템 자체의 세 요소가 상호작용하여 학습과 발전을 통한 시스템과 그 사용형태의 적응과정(adaptive process)의 結果”로 意思決定支援시스템을 定義하였다.

또한 Keen-Scott Morton(1978)은 問題형태와 시스템 機能의 측면에서 意思決定支援시스템을 “經營者(또는 意思決定者)가 半構造的(semi-structured) 意思決定問題를 해결하기 위하여 보조수단으로 利用하는 컴퓨터기술의 집합체”로 定義하였다.

이들의 定義에 입각하여 볼 때, 意思決定支援시스템은 다음과 같은 特性을 갖고 있다.

1) 意思決定支援시스템(DSS)은 目的함수와 제약조건의 일부는 계량화 할 수 있으나 일부는 계량화 할 수 없고 意思決定者의 판단(judgement)이 필수불가결한 半構造的(semi-structured) 意思決定問題를 대상으로 하며,

2) 意思決定者의 판단을 대체(replace)하기 보다는 支援(support)하는 것이 目的이며,

3) 단순히 意思決定過程의 效率(efficiency) 증대 보다는 效果(effectiveness) 증대에 역점을

두고 있다.

이 외에도 Sprague-Carson(1982)이 제안한 바와같은 다음의 特性도 들 수 있다.

- 1) 意思決定支援시스템은 최고經營者로부터 일선管理者에 이르기까지 모든 段階의 意思決定過程을 도와주어야 하며,
- 2) 다수의 상호관련된 意思決定뿐 아니라 서로 독립적인 意思決定問題도 支援해야 하고,
- 3) H. Simon이 定義한 지능(intelligence),

設計(design), 선택(choice)의 3段階 意思決定過程 모두를 支援해야 하며,

- 4) 시스템 구축(construct)이 용이하고 사용하기 편리해야 한다.

2.2 構造와 분류

일반적으로 意思決定支援시스템은 (그림 1)과 같이 4개의 下位시스템으로 구성된다.

模型베이스管理시스템(MBMS)

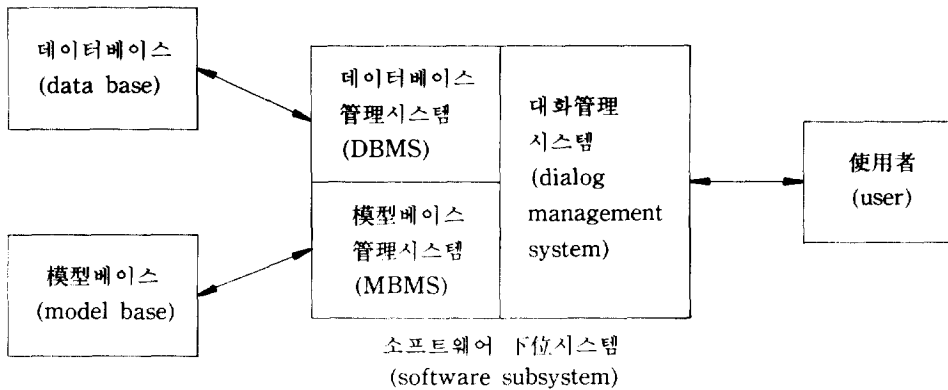


그림 1. 意思決定支援시스템의 構造

- 데이터베이스 下位시스템(database subsystem)은 經營問題와 관련된 資料와 화일의 데이터베이스를 포함하며 데이터베이스 管理시스템(DBMS)과 대화管理시스템(dialog management system)에 의하여 통제된다.

- 모델베이스 下位시스템(model base subsystem)은 意思決定支援시스템의 分析的 능력을 제고시키는 經營科學의 여러가지 模型과 財務的, 統計的 模型을 포함하고 있으며 모델베이스 管理시스템과 대화管理시스템에 의하여 통제된다.

- 소프트웨어 下位시스템(software subsystem)은 데이터베이스 管理시스템, 모델베이스 管理시스템, 대화管理시스템으로 구성될 수 있으며 使用者로 하여금 意思決定支援시스템을 統制할 수 있도록 한다.

Alter(1977)는 이러한 意思決定支援시스템을 資料지향형(data-oriented)과 模型지향형(model-oriented)으로 구분하였다. (그림 2와 표 1 참조) 資料지향형 시스템은 주로 資料處理 기사에 의해 開發되어 資料의 檢索과 分析機能을 제공하며 模型지향형 시스템은 意思決定을

支援하기 위한 회계模型, 모의실험, 최적화模型을 제공하며 주로 經營科學 專門家에 의해 開發된다.

(표 1)은 資料지향형과 模型지향형의 意思決定支援시스템을 그 特性和 적용별로 구분하였다.

그러나 대부분의 意思決定支援시스템은 그 성격의 구분이 모호하며 서로 혼합적인 성격을 갖는 경우가 많아 어떤 意思決定支援시스템이 어떤 분류에 속한다고 명확히 定義하기는 어렵다.

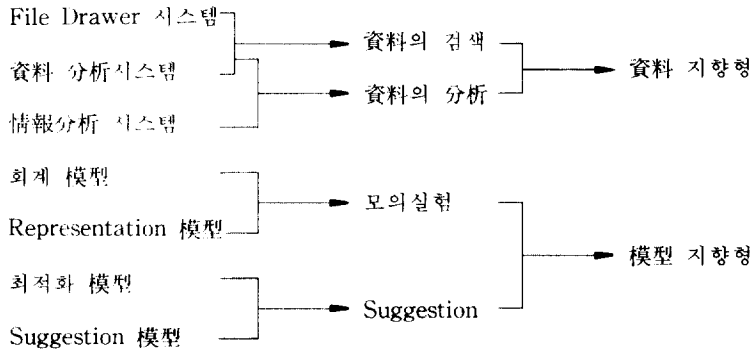


그림 2. 資料지향형과 模型지향형의 意思決定支援시스템

표 1. 意思決定支援시스템의 分類

	특 징	적 용
File Drawer 시스템	특정한 資料를 검색하고 分析하기, 分析하기 위한 on-line 시스템, 수작업에 의한 장부管理의 전산화	생산계획, 재고管理, 항공사예약 시스템
資料分析시스템	현재와 과거의 資料파일의 分析, 經營자가 아닌 스탭인력의 利用, 다양한 데이터베이스로 부터의 情報검색	재무分析, 통계分析, 예측
情報分析시스템	意思決定지향의 데이터베이스와 模型을 利用한 經營情報제공, 經營자에게 적절한 情報를 제공가능	마케팅, 판매전략
회계模型	미리 定義된 회계학적 관계에 따라 어떠한 活動의 結果추정, 經營자와 分析家가 계획업무에 利用	예산決定, 이익추정
表現(representation) 模型	미리 定義되지 않은 模型에 입각하여 어떠한 活動의 結果를 추정, 模型은 현 상황을 적절히 나타내는 것이어야 함	모의실험模型, 마케팅 조사, 생산계획
최적화 模型	일련의 제약조건 하에서 최적解를 찾으므로써 어떠한 活動에 대한 지침을 제시함, 分析기법으로 利用	교통統制模型, 계량經營模型
제시(suggestion) 模型	構造的인 업무에 대한 특정한 意思決定대안 제시, 최적화 模型의 경우보다 構造的인	컴퓨터 고장진단, Business Suggestion

2.3 시스템 設計의 接近方法

意思決定支援시스템 設計의 接近方法은 여러 學者들에 의해 研究되어 왔으며 대표적으로 Carlson, Keen 그리고 Sprague-Carlson의 接近方法을 들 수 있다.

2.3.1 Carlson의 接近方法(1979)

Carlson은 1970년대 意思決定支援시스템이 크게 성공적이지 못한 이유를 意思決定支援시스템의 設計 또는 機能이 意思決定者 또는 意思決定過程의 요구사항과 일치되지 못하는데서 기인한다고 보고 이러한 불일치의 원인을 다음과 같이 들었다.

(1) 실제로 意思決定者는 H. Simon의 3段階 意思決定過程인 知能(intelligence), 設計(design), 선택(choice) 段階를 統合하여 意思決定을 하려하지만 既存의 意思決定支援시스템은 이들 3段階를 분리하여 支援하거나 그 일부만을 支援하도록 設計되어 있다.

(2) 既存의 意思決定支援시스템은 중개자(intermediaries)로 하여금 意思決定支援시스템의 성능과 結果를 分析利用하게 함으로써 결국 意思決定者 자신의 意思決定過程에 대한 직접적 統制를 간접적 統制로 바꾸어 놓았다.

(3) 既存의 意思決定支援시스템은 장기적 기억보조 機能(데이터베이스 같은 역할의 long-term memory aid)을 제공하지만 단기적 기억보조수단(short-term memory aid)으로써의 역할을 잘 수행하지 못하고 있다.

Carlson은 시스템設計者가 모든 가능한 意思決定過程을 완벽하게 파악하기 보다는 意思決定過程에서 利用되는 중요한 대표機能(representation)과 그 연산을 파악하는 것이 용이하다고 보아 이러한 問題點을 해결하기 위하여 意思決定支援시스템은

- (1) 使用者가 시스템의 結果를 해석하고 시스템운영을 統制할 수 있는 대표機能(representation)
- (2) 3段階 意思決定過程인 지능, 設計, 선택 과정을 支援하기 위한 대표機能상의 연산(operation)
- (3) 대표機能과 그 연산을 支援하는 여러 가지의 기억보조장치등이 포함 되도록 設計되어야 한다고 제안하였다.

2.3.2 Keen의 接近方法(1980)

Keen의 方法은 최종적인 意思決定支援시스템은 학습과 발전(evolution)의 적응과정(adaptive process)를 통해서만 開發될 수 있다는 생각에서 출발했는데, Keen은 이러한 적응적 과정이 필요한 이유를 다음과 같이 제시하였다.

(1) 시스템 設計者와 使用者는 半構造的인 意思決定問題의 경우에 시스템 設計에 필요한 知識이 부족한 경우가 많으므로 시스템에 관한 적절한 機能的明細(specification)를 제시하기가 어렵다.

(2) 使用者는 자신들이 무엇을 원하는지 알지 못하며 시스템設計者도 使用者들이 필요로 하는 것이 무엇인지 확실히 모를 경우가 많으므로 초기에 開發되는 시스템은 使用者가 시스템과의 상호작용을 통하여 필요한 사항이 무엇인지 파악해 나갈 수 있도록 設計되어야 한다.

(3) 使用者가 갖고 있는 意思決定 概念은 그가 사용하는 意思決定支援시스템에 의하여 영향을 받으므로 그 시스템은 使用者에게 학습과 새로운 概念형성을 도와줄 수 있어야 한다.

이러한 Keen의 적응적 設計方式은 1) 使用者와 시스템 設計者와의 상호작용 2) 學習過程 3) 個別화된 使用性 4) 새로운 機能의 추가 등의 기능을 통하여 시스템修正과 擴張이 용이하

도록柔軟한構造를 가지게 되며 段階別 施行過程을 거치도록 되어 있다. 이러한 接近方式으로 意思決定支援시스템은 그 능력과 취급領域을 擴張시켜 나가며 결국에는 다양한 意思決定스타일까지도 포함시킬 수 있는 시스템으로 발전하게 된다.

2.3.3 Sprague-Carlson의 接近方法(1982)

Sprague-Carlson의 接近方法은 意思決定支援시스템이 비교적 용이하고 빨리 開發될 수 있는 應用分野를 선택함으로써 使用者가 自信感을 갖고 시스템設計에 참여하게 하여 시스템의 利用度를 높이려는 시도에서 출발하였다. 이들은 이렇게 設計된 시스템은 단기적으로 볼 때 시스템 設計에 따른 危險도 줄일 수 있으며 시스템 運營에 따른 이익도 保障된다고 하였다. 이러한 接近方法은 시스템의 統制可能한 중요한 변수들을 파악하므로써 단기기간내에 매우 유용한 意思決定支援시스템을 設計, 운영하려는 것이다.

앞에서 살펴본 Carlson, Keen, Sprague-Carlson의 接近方法 외에도 Bennet(1976), Senn(1984), Davis(1983), Gore-stubbel(1979) 등에 의하여 여러가지 近接方法이 제안되고 있는데 이들의 研究는 공통적으로 시스템 開發 生命週期(system development life cycle) 概念에 기초를 두고 있다. 다시말하면 1) 問題의 定義段階 2) 시스템 分析段階 3) 시스템 設計段階 4) 구체적인 시스템의 設計段階 5) 시스템 시행段階 6) 시스템의 유지보수段階 등의 諸段階를 거치며 각 段階에서의 問題點을 分析하는데 초점이 맞춰져 있다.

3. 專門家 시스템의 考察

3.1 定義와 特性

1960년대 이래로 人工知能研究家들에 의하여

인간의 사고(thinking) 過程을 研究分析하여 여러 종류의 問題를 풀기 위한 汎用 프로그램(general purpose programs)을 開發하려는 시도는 큰 진전을 보지 못하였다.

1970년대에는 問題가 쉽게 풀릴 수 있도록 問題를 表現(representation)하고 問題를 푸는데 있어 시간이 적게 걸리고 컴퓨터 기억용량을 너무 많이 차지않도록 하기 위하여 解의 探索方法에 관심을 갖게 되었다. 그러나 問題表現과 解의 探索方法 자체에 대한 研究에는 성과가 있었으나 이를 利用한 問題解決에는 많은 발전을 보지 못하였다. 1970년대 후반에 이르러 人工知能研究家들은 컴퓨터프로그램의 問題풀이능력은 프로그램이 갖고 있는 問題表現方法이나 推論方法에 달려 있는 것이 아니라 프로그램이 갖고 있는 知識(knowledge)에 달려 있다는 것을 알기 시작했다. 즉 컴퓨터프로그램이 지능적(intelligent)이기 위하여는 그 프로그램에 어떤 問題領域에 관한 양질의 많은 知識을 갖게 하는 것이 중요하다는 것을 알게 된 것이다. 이로 부터 제한된 問題領域(narrow problem domain)에서 專門家(expert) 역할을 할 수 있는 특수 目的의 컴퓨터 프로그램 開發이 시작되었으며 이러한 종류의 프로그램을 專門家 시스템(expert system)이라 부르게 되었다(Watman, 1986).

Feigenbaum(1982)은 專門家시스템을 다음과 같이 定義하였다. "專門家시스템(expert system)이란 상당한 수준의 인간의 專門知識이 필요한 어떠한 問題의 解를 구하기 위하여 知識(knowledge)과 推論過程(inference procedure)을 利用하는 知能을 갖춘 컴퓨터 프로그램이다."

專門家 시스템을 定義하는 다른 方法은 종래의 컴퓨터 프로그램과 비교하는 方法인데 가장 큰 차이 중 하나는 專門家시스템은 知識을 처리하는 것이며 既存의 컴퓨터 프로그램은 資料를 處理하는 것이다.

(표 2)는 既存의 컴퓨터 프로그램과 專門家 시스템의 차이점을 보이고 있다.

표 2. 既存의 컴퓨터 프로그램과 專門家 시스템의 차이점

既存의 컴퓨터프로그램	專門家 시스템
資料의 表現과 利用	知識의 表現과 利用
計量化된 資料 處理에 편리함	定性的인 資料處理에 편리함
대형의 데이터베이스의 效果的인 處理	대형의 知識베이스의 效果的인 處理
완벽한 情報만 요구됨	불완전하거나 不確實한 情報處理可能
알고리즘方式의 수행 절차	휴리스틱하거나 불확실한 情報處理 가능
시스템수행과 結果에 대한 說明機能 없음	시스템수행과 結果에 대한 說明機能 있음
知識과 處理 절차가 混合되어 있음	知識베이스와 統制構造(推論機關)가 명백히 분리되어 작용

이러한 專門家시스템은 다음과 같은 特性이 있다.

(1) 專門家を 대신하여 問題해결(또는 意思決定)을 하는 것이 目的이다.

(2) 專門家처럼 우수한(또는 적절한) 結果를 얻을 수 있도록 전문지식(expertise)를 갖는다.

(3) 專門家시스템은 대개 規則構造(rule format)을 知識을 表現하므로써 既存의 다른 規則에 영향을 크게 미치지 않고 知識베이스의 擴張 및 변경이 용이하여 따라서 既存 컴퓨터 프로그램보다 補修維持가 유리하다.

(4) 시스템으로부터 推論을 하여 얻어진 결론에 대한 說明(explanation) 機能을 利用하여 問題의 概念을 表現하며 이렇게 表現된 知識을 휴리스틱한 方法을 利用하여 推論하는데 專門家시스템은 이러한 심볼적인 推論(symbolic reasoning) 機能을 갖는다.

3.2 構造와 분류

專門家시스템은 知識 / 規則베이스(know-

ledge/rule base)와 統制構造(control structure)로 구성되어 있다. 知識/規則베이스는 특정한 問題와 관련된 知識의 집합체로서 事實(facts)과 規則(rule)으로 이루어져 있다. 統制構造는 推論기관(inference engine)이라고도 하며 知識/規則베이스와는 달리 일반적인 問題풀이에 관한 知識을 갖는데 인터프리터(interpreter)와 스케줄러(scheduler)로 구성되어 있다.

인터프리터는 知識을 推論하기 위해 規則을 어떻게 적용시킬 것인가 決定하며 스케줄러는 決定된 規則의 적용순서를 決定한다.

이러한 構造를 갖는 대부분의 專門家시스템은 해석(interpreting), 예측(prediction), 진단(diagnosing), 設計(designing), 계획(planning) 감지(monitors), 교정(debugging), 수리(repairing), 지도(instructing), 統制(controlling) 등의 범주로 구분할 수 있다.

專門家시스템을 유형별, 적용예별로 분류하면 (표 3)과 같다.

표 3. 專門家시스템의 분류

구분	특성	적용분야	적용사례
해석(interpreting)	관찰을 통한 현상묘사 推論	영상分析 음성인식 신호해석	Dipmeter Advisor - 지질構造推論 Dendral - 분자構造推論 Prospector - 광물탐사 Hersay - 음성인식
예측(predicting)	주어진 상황으로 부터 가능한 結果 推論	일기예보 교통량예측 농작물수확 예측 군사적 利用	SPERIL - 지진으로 인한 피해추정 Analyst - 군사적 상황예측
진단(diagonosing)	관측을 통해 시스템의 이상유무推論	의학진단 전자, 기계공학 소프트웨어	CADUCEUS - 내과진단 ONCOCIN - 암환자치료 MYCIN - 의학 DART - 컴퓨터고장진단 PUFF - 폐질환진단
계획(planning)	행동의 設計	자동프로그래밍 로보틱스 통신, 군사계획	NOAH - 로보틱스 MOLEGEN - 분자유전학 실험계획 ISIS - 생산 스케줄링
감지(monitring)	관측치와 기대치의 비교	원자력발전소管理 항공기統制 질병管理	CALLISTO - 프로젝트管理 IMS - 자동화 공장管理 VM - 수술후 환자상태의 감지
교정(debugging)	고장에 대한 처방 제시	텍스트에더터시스템 질의응답시스템 컴퓨터보조에 의한 학습 시스템	INTELLECT - 데이터베이스 ALPS - 통역
수리(repairing)	고장에 대해 제시 된 처방	컴퓨터 보수管理 자동차 진단 시스템	CRIB - 컴퓨터 고장수리
지도(instructing)	학생행동의 진단 및 교정	학생지도 시스템 동물행동의 설명 시스템	WEST - 게임지도 WHY - 강우현상의 원인지도 SOPHIE - 전자회로의 고장 진단과 定義 지도 GUIDON - 질병 치료 方法 선택의 지도
統制	전반적인 시스템 의 제어	항공기統制 經營統制 전쟁管理	ADEPT - 전쟁상황제어

(보다 광범위한 관계資料 및 문헌은 D. Waterman 1986)의 "A Guide to Expert System", pp. 239~338 에 자세히 기술됨)

3.3 專門家시스템의 設計時 고려할 사항

專門家시스템이 知識/規則베이스와 統制構造의 2개의 기본요소로 이루어지기 때문에 이들의 設計方式와 問題領域의 特性에 따라 知識/規則 表現(knowledge/rule representation), 探索(search), 知識 획득(knowledge acquisition) 등을 고려하여야 한다.

— 知識/規則表現기법(knowledge/rule representation method) : 專門家시스템의 意思決定問題에 필요한 情報나 知識을 적절히 활용할 수 있도록 知識/規則의 表現이 가능해야 하는데 이들의 表現 方法으로 일차서술논리(first-order predicate logic), 프레임(frame), 세만틱네트(semantic net), 프로퍼티 리스트(property list), 생성시스템(production system) 등의 方法이 있으며 問題의 성격에 따라 적합한 表現方法을 利用해야 한다.

— 探索기법(search method) : 探索과정(search process)은 초기상태에서 시작하여 目的을 만족하는 解를 찾을 때까지 가능한 모든 解의 공간을 조사하는 過程으로써 블라인드 探索(blind search), 휴리스틱 探索(heuristic search), 解의 공간의 추출(abstracting the solution space)의 方法 등이 利用된다.

— 知識획득(knowledge acquisition) : 知識베이스를 구성하는 知識의 획득은 주로 知識工學者(knowledge engineer)가 특정問題領域의 專門家와의 面談 혹은 學習을 통하여 획득되는데 TEIRESIAS 같은 專門家시스템에서처럼 知識工學者와 專門家사이의 意思소통에 따른 問題를 해결하기 위하여 지능을 갖춘 편집프로그램(intelligent editing program)을 利用하여 專門家로부터 知識을 획득할 수도 있는데 이 경우 편집프로그램은 高度의 對話能力(dialog capabilities)과 知識베이스구조에 관한 知識을 가져야 한다.

또한 專門家가 아닌 既存의 資料(data)나 서적으로부터 知識을 추출하기 위해서는 귀납(induction) 프로그램이나 서적이해(text understanding) 프로그램을 利用할 수 있다.

이외에도 적절한 專門家시스템 設計言語를 선택해야 하는데 LISP, PROLOG 같은 專門家시스템 전용言語를 사용할 경우 知識表現의 정교성을 높일 수 있으며 초기시스템의 開發시간을 줄일 수 있으나 專門家시스템 전용컴퓨터가 아닐 경우 시행속도가 느리며 기억용량이 부족할 수 있다.

반면에 既存의 컴퓨터言語인 FORTRAN, PASCAL, C같은 범용言語를 利用할 경우 開發시간이 많이 걸리고 知識表現에는 적절하지 못하나 한번 開發되면 시행속도가 빠르며 既存의 다른 컴퓨터소프트웨어와 호환성이 있으므로 시스템의 초기 開發시에는 專門家시스템 전용言語로 開發한 후 테스트를 거쳐 既存의 컴퓨터言語로 변환하는 方法을 利用하기도 한다.

4. 意思決定支援시스템과 專門家시스템의 統合的 시스템設計를 위한 接近

4.1 意思決定支援시스템과 專門家시스템의 統合

앞장에서 살펴본 바와 같이 意思決定支援시스템과 專門家시스템은 각기 接近方法과 그 目的이 다르며 여러가지 차이점을 가지고 있다. 意思決定支援시스템과 專門家시스템을 적절히 統合補完함으로써 經營意思決定過程의 效率과 效果를 제고할 수 있다.

意思決定支援시스템의 目的을 달성하고 취약점을 補完하기 위하여 전문가시스템이 어떻게 利用될 수 있는가 살펴보자.

첫째로, 意思決定支援시스템은 주로 非構造

的(unstructured) 또는 半構造的(semi-structured) 問題를 대상으로 하는데 컴퓨터를 활용하거나 經營科學 模型을 利用하기 위하여는 이러한 非構造的이거나 半構造的인 問題를 構造的인 問題로 바꾸는 것이 效率的이다. 이러한 變換 過程에는 知識 획득 機能과 推論 機能을 갖는 專門家시스템의 利用이 바람직하다.

둘째로, 불확실한 상황에서 중복될 수 있고 많은 量의 資料와 정보를 處理하기 위하여는 意思決定 資料를 規則(rule) 형태의 知識으로 가공하여 統制 構造와 분리시킨 專門家시스템 接近方式을 채택함으로써 자료 및 情報의 중복성을 줄이고 시간에 따라 변하는 意思決定 資料와 規則의 變化를 적절하게 시스템에 반영시킬 수 있다. 이러한 規則 형태의 知識 베이스는 판단(judgment) 過程을 支援하도록 模型 베이스 관리 시스템과 資料 베이스 관리 시스템과 專門家시스템의 統合을 가능케 할 수 있으며(Waterman, 1986), 또한 유지보수가 용이하다는 장점이 있다.

세째로, 既存의 意思決定 支援 시스템에서 利用되는 버블 探索(bubble search), 순차적 探索(sequential search) 같은 단순한 探索 方法으로는 經營 意思決定에 필요한 대규모의 資料 및 情報을 處理하는데 적합하지 못한 바 專門家시스템에서 利用하는 휴리스틱 探索(heuristic search) 方法의 利用이 바람직하다.

네째로, 經營 意思決定에는 不確實性和 複雜性으로 인하여 精確한 사실에 근거한 情報을 얻기가 어렵다. 따라서 사실이 아닐 수 있는 여론, 주관적 판단, 루머 등과 같은 비사실적(non-factual) 資料를 利用한 推論(reasoning)의 역할이 크다고 볼 수 있다. 따라서 意思決定 支援 시스템에 推論 機能을 補完하는 것이 바람직하다.

다섯째, 既存의 意思決定 支援 시스템은 주로 計量的이고 數學的인 分析 중심이기 때문에 패턴 인식(pattern recognition)이나 유사 推論(ana-

logical reasoning) 같은 定性的인(qualitative) 分析을 支援 할 수 있도록 補完될 필요가 있다. 專門家시스템은 이러한 定性的인 分析을 補完할 수 있는 좋은 方法이 될 수 있다.

여섯째, 意思決定 支援 시스템은 情報 획득 평가 를 통하여 최종 意思決定을 도와주게 된다. 이러한 過程에서 意思決定者의 判斷 誤差(judgemental biases)가 발생할 수 있는데 專門家시스템에서는 이러한 판단오차를 제거할 수 있다. 따라서 이 두 시스템으로 부터의 결론을 평가 補完함으로써 보다 우수한 意思決定을 내릴 수 있으므로 이 두 시스템의 적절한 補完이 바람직하다.

4.2 統合的 시스템 設計의 接近

2장에서 살펴 본 바와 같이 意思決定 支援 시스템 設計의 接近 方法은 基本的으로 시스템 開發 生命週期 概念(system development life-cycle concept)에 기반을 두고 있다.

이 節에서는 이러한 시스템 開發 生命週期 接近 方式을 검토하고 이 方式에 專門家시스템 設計 方式을 이식시키는 새로운 시스템 設計 方式을 제시하고자 한다.

4.2.1 시스템 開發 生命週期 方法(system development life cycle method)

시스템 開發 生命週期 方法에 의한 情報 시스템 設計 段階는 다음과 같이 구성된다.

(1) 問題의 定義 段階(problem definition stage)

分析家는 시스템의 현상과 악을 통하여 既存 시스템의 問題點을 파악하고 意思決定者 또는 使用者의 要求가 무엇인지를 確認한다. 이 段階에서 分析家는 意思決定者 또는 使用者와의 面談을 중심으로 하여 問題點과 시스템의 目的, 要求事項 등을 파악하게 되며 기술적 측면, 비록/

이익 측면에서 요구되는 시스템의 妥當性 검토도 수행 된다.

(2) 시스템 分析段階(system analysis stage)

이 段階는 파악된 問題 자체를 해결하려는 것이 目的이 아니고 그 問題를 해결하기 위하여 무엇을 해야 할 것인가를 決定하는 段階이다. 意思決定者나 使用者는 무엇이 問題이고 무엇을 해야 할 지 알지만 그것을 어떻게 處理해야 할지를 모른다. 여기서 시스템分析家は 資料흐름 圖表(data flow diagram), 資料 事典(data dictionary) 같은 기법을 利用하여 시스템의 論理的 模型(logical model)을 만들게 되는데 이 논리적 模型에는 問題點을 해결하기 위하여 새로운 시스템을 수행해야 할 機能에 관한 사항을 포함하게 된다.

(3) 시스템 設計 段階(system design stage)

전 段階에서 새로운 시스템이 수행해야 할 機能이 파악되면 이 단계에서는 그러한 機能을 어떻게 수행해야 할 것인가가 決定된다. 다시말해 논리적 模型을 물리적 模型(physical model)로 變換시키는 과정이다. 情報시스템의 電算化 정도에 따라 여러가지 代案이 제시되며 비용, 기술의 측면에서 적절한 代案이 선택된다.

(4) 具體的인 시스템의 設計 段階(detailed system design stage)

시스템 분석가는 시스템 시행(implementation)에 필요한 각각의 시스템요소를 定義하게 되는데 각 下位 프로그램과 하드웨어에 대한 구체적인 사양(specification)이 提示되고 그에 따른 費用이 추정되며 각 下位프로그램의 테스트와 施行 일정이 수립되며 각종 文書化도 이루어진다.

(5) 시스템 施行 段階(system implementation stage)

開發된 시스템이 企業經營活動을 위해 설치가 동되어지는 段階로서 필요한 하드웨어와 소프트웨어가 설치, 테스트되며 새로운 시스템의 보

안(security), 操作 및 運營에 필요한 文書도 준비되며 조직의 상황과 諸般事項을 고려하여 既存의 시스템과 새로 開發된 시스템의 대체기 일시에 또는 段階的으로 이루어 진다.

(6) 시스템의 維持補修 段階(system maintenance stage)

시스템이 계속해서 잘 작동되도록 維持補修가 뒤따르게 되는데 이 段階에서는 시스템에 내재되어 있을 수 있는 缺陷을 찾아 修正해야 하며 잘 設計된 시스템 일수록 시간의 경과에 따른 변화 사항을 시스템에 적절하게 반영할 수 있게 된다.

4.2.2 통합적 시스템 設計의 接近方法

전통적인 意思決定支援시스템의 設計方式에 專門家시스템을 이식함으로써 補完的인 시스템 設計의 接近方法은 다음의 段階로 구성된다.

(1) 問題 確認 段階(problem identification stage)

問題確認 段階는 참여자確認(participant identification), 問題確認(problem identification), 자원 確認(resource identification), 목표 確認(goal identification) 과정으로 구성된다.

참여자 確認과정에서는 시스템(專門家시스템)이 補完된 意思決定支援시스템) 구축에 필요한 情報 知識을 얻기 위하여 意思決定者, 특정 問題領域의 專門家, 知識工學者, 經營科學者 등으로 必要인원이 決定되며 역할이 定義된다.

問題確認과정은 전통적인 意思決定支援 시스템의 問題點의 段階와 유사한데 차이점으로는 知識베이스와 資料베이스와의 關係에서 概念이 定義되고 特性이 파악된다.

자원確認 過程에서는 問題의 타당성 검토의 입장에서 知識, 情報, 시간, 자금, 컴퓨터 능력 등의 자원을 확인하는데 특정問題領域의 專門家

는 과거의 문제풀이 경험 등을 검토하여 知識工學者는 과거의 유사한 問題, 知識表現 方法, 컴퓨터소프트웨어 등을 검토한다. 목표確認과정에서는 問題와 모순되지 않도록 적절한 목표를 설정하며 관련제약조건을 파악한다. 목표를 여러개의 하부목표(subgoal)로 분리할 경우 적절한 問題풀이 方式도 고려한다.

(2) 概念化 段階(conceptualization stage)

앞 段階에서 파악된 問題에 대한 知識, 情報를 적절히 表現하기 위한 概念을 정립하는 段階로서 이들 概念을 도표화(diagram) 형태로 나타내는 것이 바람직하다. 이 段階에서는 問題領域의 知識과 統制構造(또는 推論기관)가 분리되어 知識이 쉽게 修正될 수 있도록 고려해야 한다.

(3) 定型化 段階(formalization stage)

知識工學者가 특정 問題領域의 專門家(또는 意思決定者)에게 현재의 問題點을 해결하기 위한 적절한 知識表現기법에 대하여 이해시킴으로써 전 段階에서 얻어진 중요한 概念과 資料흐름(data flow)의 特性을 적절히 表現하는 段階이다.

이 段階에서는 적절한 휴리스틱이나 應用소프트웨어를 利用하여 특정 問題領域을 좁히거나 解의 공간(solution space)을 推出한다. 특정 問題領域의 專門家が 어떤 問題에 대한 解를 얻는 過程을 어떻게 模型化 하는가를 파악하는 것은 知識을 정형화하는데 있어서 중요한 역할을 한다.

또한 資料가 풍부한지 또는 重複되는지, 불확실한 資料인지, 資料획득의 方法과 비용, 問題풀이에 적절한 資料인가 등을 選擇해야 한다.

(4) 시스템 設計 段階(system design stage)

앞의 段階들로부터 파악된 시스템의 要求事項을 만족시킬 수 있도록 設計되는 段階으로써 意思決定者 또는 使用者가 設計를 용이하게 검토하여 原始(prototype) 시스템으로부터 最終시

스템으로 발전되도록 유저프렌들리(user-friendly)해야 한다.

이 段階는 傳統的인 意思決定支援시스템의 設計 段階와 類似하나 심볼적인 知識의 表現과 處理를 고려하여 規則(rule) 構造의 處理를 위한 設計가 고려되는 것에서 차이가 있다.

(5) 시스템 開發 段階(system development stage)

意思決定支援시스템의 구체적 設計段階와 유사하나 초기에는 중요한 부분을 먼저 開發한 후 修正 擴張시켜 나가는 原始(prototype) 시스템 開發 方式이 바람직하다.

(6) 시스템 테스트 段階(system test stage)

시스템의 開發 후에는 原始模型의 테스트 機能이 既存의 設計方法에서 보다 더 강화 되어야 한다. 專門家시스템의 취약점은 部分解를 評價하기가 어렵고 신뢰성이 적은 資料나 知識으로 推論 할 경우도 있기 때문에 專門家시스템으로 補完되는 意思決定支援시스템은 보다 강력한 테스트 機能이 필수적이다.

既存의 意思決定支援시스템의 경우에는 시스템 使用者의 판단착오, 使用者와 시스템 設計者와의 의사소통 부족으로 인한 시스템 機能의 취약성 등으로 인하여 시스템 사용에 따른 오류(error)를 찾아내 修正하기가 수월하지 않으나 專門家시스템 方式으로 補完되는 意思決定支援시스템의 경우 테스트過程에서 確認되는 오류(error)는 입출력 設計過程, 推論과정, 統制構造 등에서 발생하는 것으로 비교적 용이하게 찾아내어 修正될 수 있다.

(7) 原始(prototype)시스템의 修正擴張 段階(prototype revision stage)

原始시스템을 운영, 평가 하면서 原始시스템을 修正擴張에 나간다. 이 段階에서는 적절히 資料와 知識베이스에 修正을 가할 수 있으며 필요한 경우에 再概念化(reconceptualization), 再定型化(reformalization), 再設計(redesign)

段階로 다시 돌아가 修正이 가해 진다.

이와 같은 統合的 시스템 設計方式은 다음과 같은 특징을 갖는다.

(1) 原始시스템을 開發하여 최종적인 意思決定支援시스템으로의 修正擴張이 용이하다.

(2) 計量化 하기 어려운 定性的인(qualitative) 情報를 시스템에 포함시킬 수 있다.

(3) 規則構造를 가질 수 있으며 問題의 知識과 統制構造를 분리 시킬 수 있는 構造를 가지므로 補修維持가 용이하다.

(4) 專門家시스템은 좁은領域의 專門知識을 提供하므로 그 特性상 모든 수준의 經營活動을 支援하기 어렵지만 意思決定支援시스템과의 補完으로 업무적, 管理적, 전략적 意思決定 모두를 支援가능하다.

(5) 原始시스템 開發 方式으로 시스템의 신뢰성을 보다 높일 수 있다.

(6) 專門家시스템의 效率的인 探索과 推論 機能의 補完으로 既存의 意思決定支援시스템이 동

적인 環境변화에 신속정확하게 대응할 수 있게 되었다.

5. 結 論

최근에 와서 專門家시스템의 經營活動에의 應用에 관한 研究가 활발하게 進行되어 오고 있지만 既存의 意思決定支援시스템과는 독립적으로 研究되는 상황에서 두 시스템의 補完을 위한 統合시스템에 관한 研究는 컴퓨터를 利用한 經營 意思決定분야에 새로운 研究 領域이 되고 있다.

本 研究에서는 意思決定支援시스템과 專門家 시스템을 考察하고 두 시스템의 補完을 통하여 보다 效率的이고 效果的인 意思決定支援시스템 設計의 接近方法을 제시하였다. 意思決定支援 시스템과 專門家시스템의 效果적인 統合을 위해서는 두시스템의 技術적인 측면에서의 하드웨어 인터페이스 문제도 先行연구되어야 할 것이다.

参 考 文 献

1. Alter, Steven L., (Fall 1977), "A Taxonomy of Decision Support Systems," Sloan Management Review, Vol.19, No.1, pp.39~56.
2. Bennet, J.L., (July 1976), "Integrating Users and DSS," Proceedings of the 6th and 7th Annual Conference of the Society of MIS, Ann Arbor, Univ. of Michigan, pp. 76~86.
3. Bonczek, R.H., C.M. Holsapple, and A. B. Winston(1980), "The Evolving Roles of Models in DSS," Decision Sciences, Vol.11, No.2, pp.337~356.
4. Carlson, E.D., (Winter 1979), "An Approach for Designing Decision Support Systems," Data Base, pp.3~15.
5. Davis, W.S., (1983), Systems Analysis and Design: A Structured Approach, Addison-Wesley Pub. Co.,
6. Dutta A. and Amit Basu. (Sept 1984), "An Artificial Intelligence Approach to Model Management in DSS." IEEE Transactions on Computer, Vol.17, No.9, pp.90~97.
7. Feigenbaum, E.A. et al, (1982), The Handbook of Artificial Intelligence, William Kaufmann, Inc.
8. Gevarter, W.B. (1984), Artificial Intelligence, Expert System, Computer Vision and Natural Language Processing, Noyes Publishing Co.
9. Gore, M. and Stubbel, J. (1979), Element of Systems Analysis for Business Data Processing, 2nd Ed, Wm. C. Brown Pub. Co.
10. Gottinger, H.W. (1985), "SMARTD: An Intelligent Decision Support Tool for Strategic Management on Technology Diffusion," Strategische Planning, Band 1, S.261~275.
11. Hayse-Roth, F., Waterman, D.A., and Lenat, D. B. (1983), Building Expert Systems, Addison-Wesley Pub. Co. Inc.
12. Keen, P.G.W., (Fall 1980) "Adaptive Design for Decision Support Systems," Data Base, Vol.12, No.1 & No.2, pp.15~25.
13. Keen, P.G.W. and M.S. Scott Morton, (1978), Decision Support Systems: An Organizational Perspective, Addison-Wesley Pub. Co. Inc., Reading, Mass.
14. Sage, A.P. and W.B. Rouse, (July/August 1986), "Aiding the Human Decisionmaker Through the Knowledge-Based Sciences" IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-16, No.4, pp. 511~521.
15. Scott Morton, M.S. (1971), Management Decision Systems: Computer Based Support for Decision Making, Division of Research, Harvard University, Cambridge, Mass.
16. Scott, Morton, M.S. (May 1984), "Expert Decision Support Systems," a paper presented at the DSS conference, Planning Executive Inst. and Information Tech. Inst., New York.
17. Senn, J.A. (1984), Analysis and Design of Information Systems, New York, New York: McGraw-Hill Co.
18. Sprague, Ralph H. Jr, and E.D. Carson (1982), Building Effective Decision Support Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J.
19. Waterman, D.A. (1986), A Guide to Expert Systems, Addison Wesley Pub. Company.