

분산객체 기술을 이용한 금융분석모형과 자료의 통합

허순영¹, 이근우^{1*}, 김형민²

한국과학기술원 테크노경영대학원¹

130-012 서울 동대문구 청량리동 207-43

PricewaterhouseCoopers²

150-010 서울 영등포구 여의도동 23-6

요약

본 논문은 금융상품을 나타내는 자료 객체로부터 분석 모형을 분리해 내고 사용자의 필요에 따라 이들을 자유롭게 통합할 수 있는 객체 지향 구조를 제안한다. 이를 위해 자료와 모형을 효과적으로 다루는 구체적인 객체 지향 구성체를 제시한다. 또한, CORBA나 DCOM과 같은 분산 객체 기술을 이용하여 분산 시스템 환경에도 적용될 수 있도록 설계한다.

1. 서론

의사결정지원시스템은 조직의 업무 논리를 정의하고 있는 모형과 실제적인 자료를 분리하여 관리하기 때문에 모형이 실제 문제에 적용되어 실행되기 위해서는 자료와 모형을 적절히 연결할 수 있는 통합 구조가 필요하다. 의사결정지원시스템의 전통적인 연구 분야인 MS/OR 분야의 경우, 자료 형태나 모형 구조가 거의 변하지 않기 때문에 대부분의 자료와 모형의 통합에 관한 연구는 자료나 모형 변화에 대한 고려가 적었다. 그러나, 금융상품에 대한 분석을 그 목적으로 하는 금융분석모형의 경우, 금융시장의 변화에 따라 새로운 상품이 지속적으로 만들어지며 이를 분석하는 모형 역시 꾸준히 발전하고 있기 때문에, 자료나 모형 구조에 많은

변화가 발생하게 된다. 따라서, 이러한 변화를 반영하기 위해 자료와 모형의 통합 구조가 계속적으로 변경되어야 하는 문제가 발생하며, 이때 고려해야 하는 통합의 이론적 경우의 수는 상품의 종류와 모형의 개수를 곱한 수 만큼 많기 때문에, 자료와 모형 통합 구조의 유지는 전체 시스템 유지 및 보수에 있어 매우 큰 비용이 된다. 따라서, 이 같은 특성을 반영한 자료와 모형의 통합 구조가 필요하다. 이러한 구조에게 요구되어지는 요구사항은 다음과 같다.

첫째, 자료와 모형의 통합 구조가 만족시켜야 할 가장 기본적인 사항으로서 자료와 모형의 연결은 사용자가 원하는 임의의 쌍에 대해 이루어질 수 있어야 한다. 둘째, 일반적으로 많은 의사결정지원 시스템은 조직의 인트라넷을 기반으로 시스템을 구

성하는 각종 프로세스들이 분산되어 있는 분산 시스템의 형태로 구축된다. 또한, 특정 부서에서 필요로 하는 모형은 해당 부서의 사이트에 저장되는 경우가 많으며, 그 사이트 상의 프로세스들의 관리를 받는다. 따라서 자료와 모형의 통합 구조는 이러한 분산 시스템 환경을 지원할 수 있어야 하며, 모형의 위치나 자료의 위치에 관계없이 이들간의 통합을 보장하는 위치 투명성(location transparency)을 지원해야 한다.

이상의 요구 사항들을 만족시키기 위해, 본 논문에서는 시스템 내의 모든 자료와 모형을 각각의 객체로 파악하여 구조화 하고 이들간의 통합 메커니즘을 제공하는 객체 지향 구조를 제시한다. 이 구조의 근간으로는 객체들을 콤포넌트화 하여 이들간의 독립성과 재사용성을 극대화한 분산객체 기술[9]을 이용했는데, 이는 자료와 모형의 동적인 연결에 대한 요구와 분산 시스템 환경에 대한 요구를 만족시키는 기반 구조가 된다.

2. 자료와 모형의 통합에 관한 문헌 연구

모형 관리 연구 분야에서 자료와 모형의 통합에 관한 연구는 대부분 통합을 위한 모형 구조를 정의하는 것[4,5]과 모형이 필요한 자료를 추출하고 사용하는 과정[5,8]에 대한 것이다. 자료와 모형 간의 관계에 중점을 두어 수행된 연구로는 Zhang과 Sternbach의 연구와 Rizzoli at al.의 연구를 들 수 있다.

Zhang과 Sternbach[11]는 자료와 분석 모형간 인터페이스를 정의하는 추상 클래스와 실제 구현 클래스를 분리함으로써, 자료나 모형의 구현 내용이 변하더라도 일관된 통합이 가능한 구조를 제시하였다. 그러나, 이들이 제시한 구조는 클래스 디자인 패턴[3]을 이용하여 설계된 프로그램 소스 수준의 구조로서, 소스 수준에서 일관성을 유지하더라

도 시스템은 언제나 다시 컴파일되어야 한다.

Rizzoli at al.[10]은 환경 문제를 도메인으로 하여, 자료와 모형을 각각 도메인 클래스와 모형 클래스로 형상화하고 그들 간의 자료값 교환 및 자료 형태 변환 등의 연동 구조를 제시하였으며, 이를 통해 보다 효과적인 모형화 환경을 제공하고자 하였다. 이들의 연구는 모형 뿐만이 아니라 자료에 대해서도 그 구조와 형식을 정의함으로써 자료 자체의 재사용성도 높였으며, 더욱 효과적인 자료와 모형의 통합 절차를 제시하였다는 데에 큰 의의가 있다. 그러나, 개념적 수준의 설계에 그침으로써 각 클래스간의 자료 전달 과정이나 자료와 모형의 통합 및 모형 실행 과정 등의 구현 방법에 대한 언급이 부족하다.

3. 분산객체를 이용한 모형의 설계

모형 관리 분야의 연구들이 모형을 정의하는 방법은 정형화된 서술문으로 정의하는 방법, 데이터로 정의하는 방법, 그리고 모형을 하나의 서브루틴으로 정의하는 방법의 세가지[1]로 구분되어 진다. 이중 서브루틴으로 정의하는 방법은 모형관리 초기 연구들이 주로 선택한 방법이었으나, 모형의 적용 범위가 가장 광범위하다는 장점에도 불구하고, 표현방법이 다소 어렵고 모형에 변화가 발생할 때마다 시스템을 다시 컴파일 해야 하며 사용자들간 모형의 공유도 어려웠기 때문에 최근의 연구에서는 그다지 사용되지 않았다. 그러나, 프로그래밍 분야에서 바이너리 코드 수준에서의 객체간 연동 표준을 제시하는 분산객체 기술[9]이 제안되면서, 서브루틴을 이용한 모형 정의 방법의 단점이 보완되었기 때문에 본 논문에서는 모형을 분산객체를 기반으로 한 서브루틴으로서 정의한다.

사용자가 원하는 임의의 자료와 모형의 조합에 대해 통합이 가능하도록 하기 위해서는 모형의 내

부적인 구현이 어떠한 방식으로 되어 있는가와 관계없이 자료와 연동할 수 있는 구조가 되어야 한다. 이를 위하여 모형이 자료와 접촉하게 되는 모든 입출력 항목들을 모형의 인터페이스로서 정의하고 이를 모형의 실제 구현과 분리하여 모형의 구조를 정의한다. 그림1.은 대표적인 파생상품의 하나인 옵션[2,7]의 이론가를 구하는 문제를 예로 들어 이러한 구조를 나타낸다. 그림1.에서 옵션이 알 수 있는 모형의 모습은 만기일, 행사가격, 그리고 이론가 등을 가지는 모형의 인터페이스 뿐이며 따라서 이와 동일한 인터페이스를 가진 다른 어떠한 모형도 이 옵션의 분석에 자유롭게 이용될 수 있다.

이와 같이 모형을 분산객체를 이용하여 정의할 경우 얻을 수 있는 장점은 다음과 같다. 첫째, 모든 모형들이 표준화된 인터페이스를 갖게 됨에 따라 동일 자료에 대해 여러 모형을 그대로 적용할 수 있다. 둘째, 분산객체 기술은 네트워크 상에 분산되어 존재하는 객체들이 그들의 위치에 관계없이 연동할 수 있는 객체간 위치 투명성을 지원하므로, 자료와 모형의 통합도 각각의 위치에 관계없이 이루어질 수 있다. 따라서, 상품 자료와 분석 모형들이 필요에 따라 분산되어 존재하는 분산 시스템 환경을 지원할 수 있다. 마지막으로, 분산객체 기술은 플랫폼과 프로그래밍 언어에 독립적이기 때문에 시스템 사용자는 자신이 쉽게 사용할 수 있는 플랫폼과 언어를 이용하여 모형을 개발할 수 있으며, 이러한 상이한 종류의 모형 및 자료들도 모두 일관된 방식으로 통합될 수 있는 구현 투명성(implementation transparency)을 얻을 수 있다.

4. 결론

본 논문에서 제시하는 자료와 모형의 통합 구조를 통해 의사결정지원시스템이 얻을 수 있는 장점은 다음과 같다. 첫째, 자료와 모형을 자유롭게 연결하여 실행할 수 있고, 둘째, 자료나 모형의 내부 구현의 변화에 관계없이 일관된 통합 메커니즘을 얻을 수 있으며, 마지막으로, 분산 시스템 환경에서 자료와 모형의 위치 투명성 및 구현 투명성을 얻을 수 있다. 이러한 장점은 자료나 모형의 변화가 비교적 많은 금융분석모형과 같은 경우에 큰 이익을 가져올 것이며, 기타 다른 의사결정지원시스템에도 적용될 수 있을 것이다.

그러나, 본 논문에서 제시한 통합 구조는 모형이 받아들이는 입력항목이나 출력항목의 개수가 달라지는 것과 같은 인터페이스의 변화가 발생할 경우 이들의 통합을 보장할 수 없기 때문에, 전체적인 수정이 필요하게 된다. 따라서, 모형 인터페이스의 변화에 대해서도 일관된 통합 과정을 제공할 수 있도록 통합 구조의 개선이 필요하다. 그리고, 모형과 모형을 연결하여 만들어지는 복합 모형에 대한 통합도 고려할 수 있는 구조에 대한 연구도 필요할 것이다.

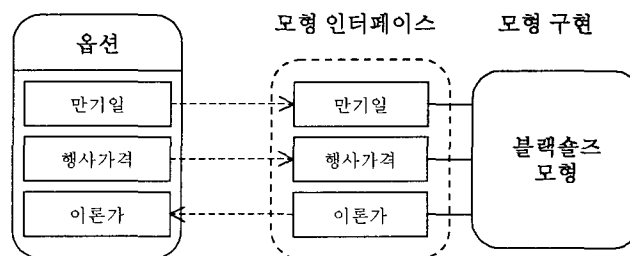


그림 1. 모형 인터페이스와 모형 구현

참고문헌

- [1] Chang, A. M., C. H. Holsapple, and A. B. Whinston, "Model management issues and directions", *Decision Support Systems*, Vol.9 (1993), pp.19-37.
- [2] Copeland, T. E. and J. F. Weston, *Financial theory and corporate policy*, 3rd ed., Addison-Wesley, Reading, MA, 1992.
- [3] Gamma, E., R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*, Addison-Wesley, Reading, MA, 1995.
- [4] Geoffrion, A. M., "The formal aspects of structured modeling", *Operations Research*, Vol.37, No.1 (1989), pp.30-51.
- [5] Huh, S. Y., "Modelbase construction with object-oriented constructs", *Decision Science*, Vol.24, No.2 (1993), pp.409-434.
- [6] Huh, S. Y. and Q. B. Chung, "A model management framework for heterogeneous algebraic models: object-oriented database management systems approach", *Omega, Int. J. Mgmt Sci.*, Vol.23, No.3 (1995), pp.235-256.
- [7] Hull, J. C., *Options, futures, and other derivatives*, 3rd ed., Prentice Hall, NJ, 1997.
- [8] Liang, T. P., "Integrating model management with data management in decision support systems", *Decision Support Systems*, Vol.1 (1985), pp.221-232.
- [9] Pritchard, J., *COM and CORBA® side by side: architectures, strategies, and implementations*, Addison-Wesley, Reading, MA, 1999.
- [10] Rizzoli, A. E., J. R. Davis, and D. J. Abel, "Model and data integration and re-use in environmental decision support systems", *Decision Support Systems*, Vol.24, No.2 (1998), pp.127-144.
- [11] Zhang, J. Q. and E. J. Sternbach, "Financial software design patterns", *Journal of Object-Oriented Programming*, Feb. (1996), pp.6-12.