

## 장외파생상품거래를 위한 유연한 의사결정지원시스템 아키텍처 설계에 관한 연구: A은행 사례

이근우\* · 양근우\*\*

### 〈 목 차 〉

I. 서론	4.2 시스템 구축 목표
II. 선행연구	4.3 시스템 구성 방안
2.1 모델기반 의사결정지원시스템	4.4 시스템 아키텍처의 구현
2.2 장외파생상품 거래지원시스템	4.5 시스템 구축 결과
III. 장외파생상품 가치평가모델기반 DSS	V. 결론
3.1 제안시스템 요구사항	참고문헌
3.2 제안시스템 아키텍처	<Abstract>
IV. 모델기반거래지원시스템(MTSS) 구축사례	
4.1 추진배경	

### I. 서론

의사결정지원시스템(DSS)은 기업 활동에서 발생하는 다양한 의사결정 문제를 정량적이고 과학적인 방법으로 해결하도록 지원함으로써 기업의 경쟁력 강화에 중요한 역할을 담당해왔다(Shim et al., 2002; Zeleznikow and Nolan, 2001). 최근에는 정보 인프라의 급격한 발달로 가용 정보의 양과 활용 방법이 더욱 다양해지면서, DSS 역시 과거에 비해 매우 광범위한 분야에서 세분화, 전문화되어 연구, 개발되고 있다(박

정희, 2006; Power, 2001).

이러한 DSS는 금융상품의 가치를 정교하게 모델링하고 이를 토대로 거래의사결정을 내림으로써 수익을 창출하는 금융시장 참가자들에게도 매우 중요한 역할을 담당한다. 일반적으로 거래 의사결정은 거래 대상 금융상품의 미래 수익성과 변동 위험성 등을 고려하여 이루어지는데, 이를 모델링하고 분석하는 것은 고난도 수학적 이론과 고밀도 계산을 필요로 하는 매우 전문적인 작업이다. 특히, 장외파생상품은 다른 상품의 가격에 따라 종속적으로 가치가 결정되는 이중적 가

\* 우리은행 트레이딩부(주저자), 경영공학박사, keunwoo.lee@wooribank.com

\*\* 계명대학교 사회과학대학 전자무역학과 조교수(교신저자), kwyang@kmu.ac.kr

치구조를 가질 뿐만 아니라 거래당사자간 이해 관계에 따라 세부거래 조건이 임의로 조정될 수 있는 형태이기(Hull, 1997) 때문에, 더욱 복잡한 모델링이 요구되는 경우가 많다. 그러나 일반적인 거래담당자가 모든 전문성을 갖추고 거래에 임하기는 어렵기 때문에, 금융공학 전문가에 의해 개발, 구현된 모델을 이론에 맞춰 적절히 활용할 수 있도록 지원하는 DSS의 역할은 성공적인 거래의사결정을 위해 반드시 필요한 요소 중 하나이다.

그럼에도 불구하고, 아직까지 많은 금융기관의 거래지원시스템은 금융상품 모델링이나 거래담당자의 모델 활용 등을 지원하기 위한 사용자 편의성 증대보다는 거래업무 프로세스의 자동화 또는 대용량 거래 데이터의 효과적인 관리 등 데이터 처리에 대한 이슈를 중심으로(김준석, 2008) 구축되고 있다. 이러한 거래의사결정 지원에 대한 사용자 편의성 부족은 거래담당자가 적절한 모델을 선택하고 적용하는데 어려움을 느끼게 함으로써 결과적으로 시스템에 대한 낮은 만족도로 이어지게 된다. 또한, 안정적이고 효율적인 데이터 처리를 위해 시스템 내부 프로세스가 거의 공개되지 않는 닫힌시스템(closed system)으로(Wikipedia, 2010a) 구축되는 경우가 많기 때문에, 예상되는 시장상황 변화에 따라 사전에 다양한 설정으로 모델을 적용해 보는 what-if 분석(Wikipedia, 2011a)이나 모델을 특정 상황에 최적화하여 적용해보는 등의 사용자 커스터마이징이 어려운 경우가 많다. 그러나 장외파생상품 시장은 신상품의 출시나 금융당국 규제의 변동 등으로 시스템 환경에 변화가 많은 곳으로, 이를 금융상품 모델링이나 모델 분석 절차 및 결과 도출 방식 등에 시의 적절히 반영하여

거래의사결정을 지원하는 것이 매우 중요함에도 닫힌시스템 구조는 근본적인 한계를 가질 수밖에 없다.

본 논문은 기존 데이터 처리 중심 거래지원시스템이 갖는 이러한 한계를 극복하기 위해, 금융상품의 모델링 및 효과적인 활용을 지원하고 시장 환경 변화에 유연하게 대처할 수 있는 DSS 프레임워크를 제안한다. 특히, 국내의 대표적 시중 은행에서 수행된 장외파생상품 거래지원시스템 구축 사례를 통해 실무적 관점에서 고려해야 할 사항들을 중심으로 제안 프레임워크의 장점과 한계점 등을 분석한다. 본 논문의 구성은 다음과 같다. II장에서는 모델기반 DSS와 관련한 기존 연구를 살펴보고, 기존 장외파생상품 거래지원시스템의 형태 및 장단점 등에 대해 논의한다. III장에서는 장외파생상품 거래를 위한 모델기반 DSS가 갖춰야 할 시스템 요구사항을 정리하고 이를 충족하기 위해 본 논문에서 제안하는 시스템 아키텍처를 제시한다. IV장에서는 국내 A은행의 장외파생상품 지원시스템 구축 사례를 통해 제안 프레임워크의 실무적 관점에서 적용 방안과 장단점 등을 소개한다. 마지막으로 V장에서 결론 및 향후 연구 방향을 정리하고자 한다.

## II. 선행연구

### 2.1 모델기반 의사결정지원시스템

의사결정지원시스템(DSS)은 구조화되고 반복적인 작업을 주로 지원하는 초기 정보시스템과 달리 구조화되지 않은 (semi-structured or unstructured) 문제 해결을 지원하는 시스템으로

분리 주장되었다(Gorry and Scott-Morton, 1971). 이에 따라 많은 연구에서 전형적인 DSS의 구조는 기존 경영정보시스템(MIS)과 다른 형태를 갖도록 구성되었으며, 주로 모델(베이스), 데이터베이스 그리고 대화 시스템(dialog management)의 세가지 요소로 정의되었다(Sprague and Carlson, 1982; Turban, 1995). 이후 시스템을 활용하는 업무 형태에 맞춰 의사결정을 지원하는 주요 시스템 요소가 무엇인가에 따라 보다 세분화되어 연구되고 있다. Power and Sharda(2007)는 DSS의 유형을 의사결정을 지원하는데 가장 주요한 시스템 구성요소가 무엇인가에 따라 모델기반(model-driven), 대화기반(communication-driven), 데이터기반(data-driven), 문서기반(document-driven), 지식기반(knowledge-driven) DSS의 다섯 가지 형태로 구분하였다.

이 중 모델기반 DSS는 가장 오랫동안 연구되어 온 시스템 형태로서, 금융상품 가치평가모델과 같이 수리적인 모델을 활용하는 분야에 가장 적합한 형태이다. 이 때, 모델이란 실제 세계의 문제를 정형화되고 수리적인 형태로 표현한 것으로(Goffrion, 1987), 정량적이고 과학적인 방법에 따라 기업의 영업의사결정을 지원하기 위해 활용된다. DSS에서 모델 관리 기능은 조직화된 모델의 집합인 모델베이스로 정의되어, 모델의 생성, 저장, 추출, 갱신 및 실행 등의 모델 조작 기능이 제공된다(Applegate et al. 1986). 초창기 모델베이스에 대한 연구는 데이터베이스와 대칭되는 관점에서 관계형(relation) 모델을 바탕으로 이루어졌다. Blanning(1985)은 여러 종류 모델의 입력과 출력을 서로 일치시켜 실행할 수 있는 범용의 관계형 모델을 이용하였으며, Lenard(1986)는 데이터베이스 시스템에서 모델

을 표현하기 위한 관계형 스키마를 제안하였다. 한편, Muhanna and Pick(1988)은 모델베이스로서의 상용 데이터베이스관리시스템(DBMS)들의 한계를 제시하고 자체적인 파일 시스템 기반의 프레임워크를 제안하였으며, Desai(1991)는 모델 관리에 있어 관계형 접근방식의 한계에 대해 논의하였다.

1980년대 이후 객체지향(object-oriented) 방법론의 발전과 함께, 객체지향모델링에 입각하여 모델베이스를 구현하고자 하는 연구가 이어졌다. Lenard(1987), Le Claire and Sharda(1990) 그리고 Muhanna(1990)는 객체지향 프로그래밍 접근방식을 제안하였는데, Geoffrion(1987)의 구조화 모델링 방법론에 기반하여 복잡한 모델 관리 환경을 구축하기 위해 객체지향 프로그래밍 언어의 개념을 정교하게 적용하였다. Ma(1995,1997)는 모델베이스를 구축하는데 있어 객체지향 방법론의 주요 개념인 데이터추상화(data abstraction), 정보은닉(information hiding), 계승(inheritance) 등을 적용하였다. 이러한 객체지향 관점의 모델베이스 구축은 관계형 모델이 가진 정형화된 형태에서 탈피하여 보다 직관적으로 모델을 생성, 관리 및 실행할 수 있도록 하였다. 또한 독립된 객체간 느슨한 관계로서 정의되는 시스템 구조는 환경 변화에 따른 모델의 추가 및 변경이 보다 쉽게 이루어질 수 있는 유연성을 제공한다(Lee, 2006).

모델을 활용한 의사결정지원 과정은 특정 상황에서 모델이 제시하는 해를 그대로 따르는 정적인 형태가 아니라, what-if 분석 등과 같이 모델과 사용자간 지속적인 상호작용을 통해 최적의 해를 찾아가는 동적인 과정이다. 따라서 이를 효과적으로 지원할 수 있는 사용자 인터페이스

스의 구현은 모델 기반 DSS의 성공적인 활용을 위해 매우 중요한 요인이다(Power, 2007). 사용자 인터페이스 구현 도구로서, 마이크로소프트 엑셀과 같은 스프레드시트 프로그램은 테이블 형태의 직관적인 데이터 처리 방식과 다양한 자료 분석 기능 및 그래프 작성 기능 등 매우 효과적인 분석환경을 제공한다. LeBlanc et al.(2000)은 스프레드시트를 이용하여, 건설 프로젝트에 관리자를 배정할 때 최소의 비용으로 관리자들의 업무 부담을 균형 있게 유지할 수 있는 최적의 방법을 찾는 모델을 구현하였다. Raden(2005)는 기업의 IT 인프라에서 스프레드시트 프로그램이 갖는 유용성 및 과급효과와 그로인한 문제점 등을 논의하고, 전사적 IT 관리정책에 부합할 수 있도록 제어된 형태로 활용하는 방안에 대해 제시하였다. 이러한 스프레드시트 기반 DSS는 특히 파생상품 평가 모델 등과 같이 수리적 계산을 바탕으로 해를 구하는 분야에서 그 유용성이 더욱 높다 하겠다. 또한, 스프레드시트 상에 출력되는 시스템 분석결과는 이를 사용자가 다시 가공하여 원하는 형태로 재정리하는 것이 매우 용이하기 때문에, 시스템 분석 결과의 유용성이 크게 높아질 수 있다. 스프레드시트 기반 DSS의 이와 같은 장점을 활용하여 본 논문에서 제시하는 DSS 프레임워크 역시 엑셀을 기반으로 데이터의 가공 및 결과 출력 등을 구현한다.

## 2.2 장외파생상품 거래지원시스템

장외파생상품 거래지원시스템은 주된 지원 대상 업무 범위에 따라 크게 거래 워크플로우(workflow) 시스템과 상품가치평가 시스템의 두 가지 유형으로 분류할 수 있다.

먼저, 거래 워크플로우 시스템이란 최초 거래의 투입에서부터 최종 회계처리 및 결제처리까지 일련의 반복적인 거래 워크플로우의 안정적인 지원을 목표로 구축된 시스템으로서, 대표적인 상용 시스템들로는 Kondor(Thomson Reuters, 2010), Murex(Murex, 2009), Summit(Misys, 2008), Calypso(Calypso, 2010) 등이 있다. 장외 파생상품 거래는 중앙집중식 중계기관을 통하는 거래소 상품과 달리, 거래상대방과 일대일 계약을 체결하기 때문에 거래정보의 확인 및 그에 따른 대금 지급까지 모든 작업을 거래당사자가 직접 수행해야 하며, 특히 금융상품의 특성상 업무 처리과정에서의 실수는 막대한 금전적 손실로 직결될 수 있는 만큼 업무절차에 따라 정확하고 빠르게 거래 정보를 처리하는 것이 매우 중요하다. 이처럼 거래데이터의 안정적이고 신속한 처리를 위해 거래 워크플로우 시스템은 대부분 내부구조를 공개하는 열린시스템(open system)보다는 내부적으로 성능을 최적화시킨 닫힌시스템(closed system)을 선호하며, 이로 인해 장외파생상품에서 빈번히 발생하는 신규 상품의 출시나 감독당국의 규제 변경, 또는 상품 분석 방식의 변경 등의 변화를 반영하기 위해 시스템을 개선하고 보수하는 것이 매우 어려운 단점이 있다.

한편, 상품가치평가 시스템은 사용자가 파생상품의 가치를 정확히 평가할 수 있도록 매우 다양하고 전문적인 형태의 모델링 및 분석 기능을 제공한다. 대표적인 상용 시스템들로는 FinCad(FinCad, 2010), Numerix(Numerix, 2010), MatLab(MathWorks, 2010) 등이 있으며 공개 도구로 QuantLib(QuantLib, 2010)이 있다. 이들 시스템은 엑셀과 같은 스프레드시트 프로그램에 add-in 형태로 추가되어 모델링 도구를 제공하며

나 Visual Basic이나 C++와 같은 프로그래밍 언어 내에서 사용할 수 있는 라이브러리 형태로 제공된다. 사용자는 이들이 제공하는 함수들을 조합하여 비교적 이론적으로 복잡한 이색옵션(exotic options)(Hull, 1997)형태의 파생상품까지도 모델링하고 그 해를 계산해 낼 수 있다. 이와 같은 상품가치평가 시스템은 메인 거래업무를 전체적으로 지원하는 거래 워크플로우 시스템의 보조역할로서 상품에 대한 모델링 기능을 보완하는 용도로 많이 사용된다. 그러나, 주로 일회성 계산위주의 프로그램으로서 전체 거래포지션의 유지 및 관리의 용도로 사용하기에는 어렵다는 한계가 존재한다.

이처럼 거래 워크플로우 시스템과 상품가치평가 시스템 개별로는 장외파생상품 거래업무 전체를 지원하는데 한계가 존재하며, 상호보완적인 위치에서 유기적으로 통합 운영되어야 한다. 따라서, 본 논문에서는 각 시스템이 가진 강점을 활용하고 한계를 보완하여 통합함으로써 효과적인 지원시스템 환경을 제공할 수 있는 DSS 시스템 프레임워크를 제시하고자 한다.

### Ⅲ. 장외파생상품 가치평가모델기반 DSS

#### 3.1 제안시스템의 요구사항

앞서 설명한 바와 같이 장외파생상품에 대한 거래의사결정은 해당 상품에 대한 가치평가 모델의 분석결과를 토대로 이루어진다. 이를 지원하기 위한 DSS는 다른 분야의 시스템에 비해 다음과 같은 특징을 갖는다.

첫째, 가치평가 모델을 활용한 금융상품 거래 업무는 모델이 기반을 두는 금융공학 이론에 대한 이해뿐만 아니라 시장 상황에 따라 이를 적절히 적용할 수 있는 경험과 통찰력도 필요한 어려운 과정이다. 그러나 모든 거래담당자가 그러한 지식과 경험을 완벽히 갖추기는 어렵기 때문에 기반 지식이 충분하지 않은 사용자라도 큰 어려움 없이 모델을 이해하고 적용할 수 있도록 지원하는 사용자 인터페이스의 역할이 매우 중요하다. 즉, DSS의 사용자 인터페이스는 사용자가 모델의 논리를 쉽게 파악할 수 있고 시장 상황 변화에 따라 모델 적용 결과의 변화를 분석할 수 있게 하는 등 높은 사용자 편의성을 제공해야 한다.

둘째, 금융시장은 매우 역동적으로 변화하는 시스템 환경으로서 이에 신속하고 효과적으로 대응할 수 있는 유연한 시스템 구조를 갖추어야 한다. 예를 들어, 장외파생상품 시장에서 시장참가자들은 보다 좋은 수익 구조의 신상품 개발을 통해 이익을 확대해가는 경우가 많으며 따라서 신상품에 대한 가치평가모델을 빠른 시일 내에 추가하고 실행할 수 있도록 하는 것이 매우 중요하다. 또한, 거시경제 상황의 변화 등에 의해 감독당국의 규제가 바뀌는 경우도 많기 때문에 상품의 가치평가 방법을 변경하거나 분석결과에 대한 보고서 형태를 변경하는 등 규제변화에 따른 임의적인 수정에 적절히 대응할 수 있어야 한다.

셋째, 금융상품 거래에 대한 의사결정은 언제나 과거 거래를 누적한 현재 포지션을 근거로 이루어지므로 주기적으로 대용량의 거래 데이터를 모델에 적용하여 실행하고 결과를 분석하는 작업이 필요하다. 또한, 각 거래상대방 별로 사전 설정된 거래가능 한도금액 등의 거래위험관리를

위한 사전규제의 충족여부 등도 확인할 수 있어야 한다. 이를 위해, 모델기반 DSS는 거래 데이터를 추적하고 통제, 관리하는 조직 내 트랜잭션 시스템과 유기적으로 통합되어야 하며 모델 실행시 필요에 따라 거래데이터의 자유로운 접근 및 활용을 보장할 수 있어야 한다.

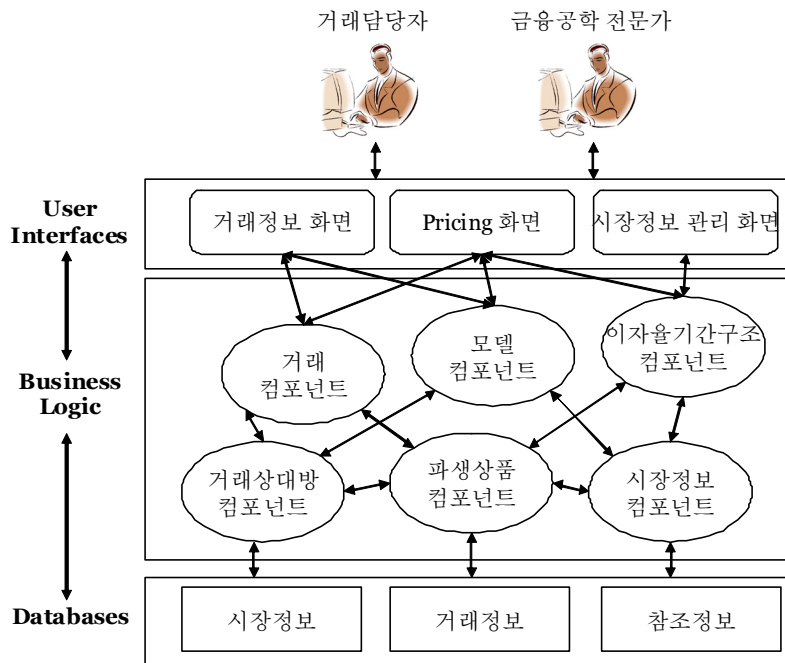
### 3.2 제안시스템 아키텍처

앞 절에서 살펴본 시스템 요구 사항 달성을 위해, 본 논문에서는 주요 시스템 구성 요인을 계층화한 구조의 DSS 아키텍처를 제안한다. <그림 1>은 제안시스템의 3계층(3-tier) 구조를 보여준다. 가장 하위에 거래정보 및 시장정보 등을 저장하는 데이터베이스 계층을 토대로 파생상품 가치평가 모델이나 거래업무 로직 등을 구현한 비즈니스 로직 계층, 그리고 최상위의 사용자

인터페이스 계층으로 구성된다. 이러한 계층화 구조는 각 시스템 요인 별 기능의 독립성을 높임으로써 각 계층에서 발생하는 요건변경의 수정이 다른 계층에 미치는 영향을 최소화하여 시스템 유연성을 높이고 유지보수 비용을 감소시킨다. 또한, 계층별 구현을 통해 개발의 복잡성을 감소시키고 구현결과의 품질도 향상시킬 수 있다. 제안시스템의 3계층 구조 중 금융상품 가치평가모델 기반 DSS의 특성을 구현하는데 중요한 역할을 담당하는 사용자 인터페이스와 비즈니스 로직의 두 계층에 대해 좀 더 자세히 설명하고자 한다.

#### 3.2.1 스프레드시트 기반 사용자 인터페이스

제안시스템의 3계층 구조 중 최상위의 사용자 인터페이스는 거래담당자가 가치평가 모델을 쉽



<그림 1> 장외파생상품 가치평가모델기반 DSS의 3계층 구조.

계 이해하고 적용할 수 있도록 지원함으로써 시스템의 활용도에 직접적으로 영향을 미치는 중요한 시스템 요소이다. 본 논문에서는 사용자 편의성을 극대화하기 위해 대다수 거래담당자들에게 매우 친숙한 업무도구인 엑셀 프로그램의 스프레드시트를 이용한 사용자 인터페이스의 구성을 제안한다. 거래담당자는 개별 거래에 대한 입력, 조회, 분석 등을 위한 화면과 여러 거래를 전체적으로 분석하기 위한 보고서 화면, 그리고 기간별 이자율 분포를 나타내는 이자율 기간구조 등과 같은 금융시장 현황정보를 조회하기 위한 화면 등이 구현된 스프레드시트를 통해 시스템 기능을 이용한다.

이러한 스프레드시트 기반 인터페이스의 장점은 다음과 같다. 첫째, 대다수 거래담당자들에게 엑셀은 이미 매우 익숙한 도구이기 때문에, 신규 시스템의 운영에 따른 거부감을 최소화할 수 있고 비교적 짧은 시간과 노력만으로도 시스템의 기능을 습득하도록 할 수 있다. 이러한 스프레드시트에 대한 친숙성은 DSS의 사용자 편의성을 높일 뿐 아니라, 사용자가 쉽게 시스템의 분석결과를 자신이 원하는 형태로 다시 가공할 수 있게 한다. 이를 통해 시스템 상에 구현되기 어려운 특정 사용자만의 고유 요건이나 혹은 시장전망 보고서와 같은 정성적 분석요건 등도 시스템이 제공하는 정량적 결과를 이용해 사용자가 직접 생성할 수 있다. 또한, DSS 분석결과를 자신의 관점에 따라 재가공하는 과정을 통해 거래담당자는 기반모델의 논리적 배경이나 그 적용방식 등에 대한 이해를 높일 수 있다. 둘째, 거래포지션 분석보고서 등의 사용자 인터페이스를 엑셀 스프레드시트 상에 구현하는 것은 본격적인 프로그래밍을 통해 동일한 작업을 하는 것

에 비해 매우 적은 시간과 노력으로도 충분한 비교적 단순한 작업으로, 시스템 구축의 시간과 비용을 크게 줄일 수 있다. 이러한 장점은 감독규제의 변경 등과 같은 요건의 변화를 관련 스프레드시트에 반영하는 것 또한 쉽게 하여 유지보수 비용을 낮추고 사용자 요건의 변화에 시스템이 빠르게 대응할 수 있는 요인이 된다. 셋째, 거래담당자와 시스템개발자 양측에게 모두 친숙한 엑셀 스프레드시트를 매개로 시스템 개발요건이 논의됨에 따라 이들 간 의사소통이 원활해지고 구현 결과에 대한 거래담당자의 만족도 또한 매우 높아진다. 마지막으로, 금융거래 업무를 지원하기 위한 각종 시스템들(예를 들어, 거래입력 및 체결 시스템, 시장정보 조회시스템, 이론가격 계산용 프로그램 등)은 기본적으로 엑셀과의 데이터 연계를 염두에 두고 개발되는 경우가 많기 때문에, 이들 시스템으로부터 필요한 정보를 제공받거나 DSS에서 산출된 결과값을 입력데이터로 제공하는 등의 상호 데이터 교환이 쉽게 이루어질 수 있다.

반면, 엑셀기반 인터페이스의 단점으로는 스프레드시트 상에서 사용자의 자유로운 데이터 입출력이나 조작 명령 등을 세밀하게 제어하기는 어렵기 때문에 사용자의 실수를 사전에 방지하는 형태로 인터페이스를 구현하기가 쉽지 않다는 점이다. 예를 들어, 특정 파생상품을 분석하는 과정에서 해당 상품구조의 논리상으로는 적절하지 않은 범위의 분석 파라미터 값을 입력하는 경우, 이를 인터페이스 수준에서 통제하도록 구현하는 것은 거의 불가능하다. 본 논문에서는 이러한 사용자 동작에 대한 논리적 타당성의 검증은 하위의 비즈니스 로직 계층에서 점검하고 실제 실행 전 사용자에게 오류여부에 대한 피드

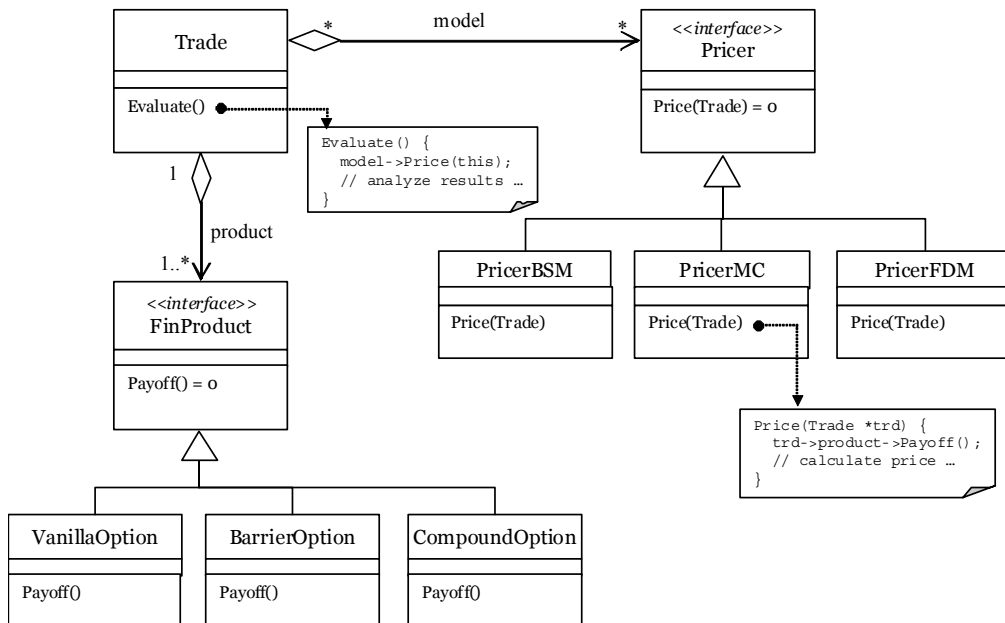
백을 줌으로써 사용자의 논리적 실수에 의한 시스템 오류를 방지하는 형태로 이를 보완하고자 한다.

### 3.2.2 업무로직의 컴포넌트 기반 모듈화

제안시스템의 3계층 구조 중 두 번째 계층은 파생상품 가치평가모델이나 거래 정보 처리 절차 등과 같은 업무로직을 모듈화한 비즈니스 로직 계층이다. 이 계층은 시스템의 유연성을 높이고 업무로직의 변화 등에 따른 유지보수가 쉽게 이루어지도록 컴포넌트 기반 구조(Box, 1998)로 구성된다. 일반적으로, 컴포넌트 기반 구조는 컴포넌트간 기능의 독립성이 매우 높고 상호간의 의존도가 약한 연결(loosely coupled) 관계로 이루어지기 때문에, 개별 컴포넌트의 구현 및 변경이 쉽고 재사용성도 높아지게 된다.

<그림 2>는 비즈니스 로직 계층의 주요 객체

간 개괄적 모듈구조를 UML 클래스 다이어그램(Booch et al., 1999)을 이용하여 보여준다. 모듈 구조의 최상위에는 거래, 파생상품, 가치평가모델을 각각 구조화한 Trade, FinProduct, Pricer 클래스가 정의된다. 이 중, FinProduct와 Pricer는 모든 파생상품 및 가치평가모델의 인터페이스를 정의하는 역할을 한다. 예를 들어, FinProduct는 환율이나 이자율 수준 등의 시장 상황에 따른 파생상품의 현금흐름 내역을 생성하기 위한 Payoff() 함수를 포함하며, Pricer는 가치평가 계산을 위한 Price() 함수를 가진다. 이때, 이들 함수는 단지 이름만을 정의하는 것이며, 함수의 구체적인 동작은 이들 클래스를 상속받아 실제 특정 파생상품이나 평가모델을 구조화하는 하위클래스에서 구현된다. <그림 2>에서는 FinProduct를 상속받은 VanillaOption, BarrierOption, CompoundOption 클래스가 각



<그림 2> 주요 구성 객체간 모듈화 구조.



각 자신들이 구조화한 상품의 특성에 맞도록 Payoff() 함수를 구현하며, Pricer를 상속 받은 PricerBSM, PricerMC, PricerFDM 클래스는 각 평가모델의 알고리즘에 따라 Price() 함수를 구현한다. 한편, Trade 클래스는 오직 FinProduct 나 Pricer 클래스와만 상호작용하는 형태로 정의되어 실제 대상상품이나 평가모델에 관계없이 동일한 형태로 동작하도록 구현된다. 이후 프로그램 실행 시에 실제 지정된 상품 및 모델에 해당하는 하위클래스의 함수가 호출되어 적절한 동작이 실행된다. 예를 들어, Trade의 Evaluate() 함수는 model로서 연결된 Pricer의 Price() 함수를 호출하고 그 결과를 이용하여 해당 거래의 가치를 평가하는데, 프로그램 실행과정에서 Pricer의 하위클래스 중 실제로 어떤 클래스가 지정될 것인지에 관계없이 한가지 형태로 작성된다. 이는 객체지향 프로그램의 클래스 상속(inheritance) 및 함수 호출의 다형성(polymorphism) 메커니즘을 활용한 것으로, 단순하고 직관적인 클래스 구조의 설계를 가능하게 하고 프로그램의 유연성을 높인다(Joshi, 2004). 즉, <그림 2>의 클래스 구조에서 새로운 상품이 개발되거나 신규 모델이 추가될 때, 이를 FinProduct나 Pricer의 하위클래스로서 정의하고 적절한 Payoff() 함수 혹은 Price() 함수를 구현하면 Trade 및 다른 클래스는 아무런 변경 없이도 해당 신상품이나 신규 모델과 연동하여 동작할 수 있게 된다.

#### IV. 모델기반거래지원시스템(MTSS) 구축 사례

이번 장에서는 국내 시중은행의 실제 시스템

구축사례를 통해 본 논문에서 제안하는 DSS 프레임워크가 어떻게 구현되고 앞서 살펴본 시스템 요구사항을 충족할 수 있는지에 대해 논의하고자 한다.

##### 4.1 추진배경

A은행은 대표적 국내 시중은행으로서 금융상품 거래업무 부서 내에 환율 및 이자율 등을 기초 자산으로 하는 장외파생상품 거래담당 팀을 운영하고 있으며, 1990년대 후반부터 본격화된 국내 장외파생상품 시장의 발전에 따라 거래 대상 상품의 종류와 규모를 확대해 나가고 있다. 이를 지원하기 위해, 거래의 공식적인 기록과 손익 산출을 위한 메인 거래시스템을 운영하고 있으며 몇 가지 상품 유형에 대해서는 좀더 다양하고 전문화된 분석지표를 계산하기 위한 모델링 도구를 추가로 보유하고 있다.

A은행이 다루는 장외파생상품의 종류가 늘어감에 따라 메인 거래시스템과 모델링 도구를 통한 거래포지션의 운영은 몇 가지 측면에서 한계에 부딪혔다. 그 중 가장 큰 어려움은 이들 프로그램이 외부제공업체로부터 패키지 형태로 구매하여 설치한 것으로 신규 상품을 추가하거나 기존 상품에 새로운 평가 모델을 반영하는 등의 요건변경에 따른 관련 모듈의 수정이 극히 제한적이라는 점이었다. 또한 종종 메인 거래시스템의 가치평가 결과가 거래담당자들의 예상에서 상당히 벗어나는 경우도 있었으나 프로그램의 내부 로직을 정확히 알 수 없었기 때문에 차이의 원인을 명확하게 규명하기 어려운 경우도 많았다.

이러한 기존 시스템의 한계로 인해, 거래담당자들은 메인 거래시스템에서 다루기 어려운 상

품들을 범용의 스프레드시트 프로그램인 마이크로소프트 엑셀을 이용하여 개별 파일로 저장, 관리하였으며, 가치평가 또한 자체적으로 구현한 모델을 이용하였다. 그러나 이러한 엑셀 파일을 통한 거래 관리는 다음과 같은 문제를 발생시켰다. 첫째, 엑셀로 따로 관리하는 상품의 종류와 수량이 늘어남에 따라, 메인 거래시스템 내 거래 포지션과 엑셀파일로 존재하는 거래포지션을 통합하여 전체 포지션을 파악하고 분석하는 것이 점점 더 어려워지고 있었다. 특히, 엑셀 파일로 저장되어 있는 거래들의 경우 각각을 따로 계산한 후 메인 거래시스템의 포지션과 수기로 합쳐줘야 했기 때문에, 비효율적인 수작업에 많은 시간이 소모되었다. 또한, 일관성 있는 분석을 위해서는 이자율이나 환율 등과 같은 지속적으로 변동하는 시장정보가 동일하게 설정된 상태에서 계산이 이루어져야 하나 메인 거래시스템과 엑셀파일 별로 분석 시점이 차이 나거나 혹은 사용자가 조작 중 실수 등의 오류로 종종 일관성 없는 결과가 얻어지기도 하였다. 둘째, 파일시스템이 가지는 한계로 인해, 거래담당자의 의도하지 않은 조작 오류 혹은 의도적인 부정행위에 의해 거래정보가 왜곡 또는 누락되는 것을 통제하거나 그 발생 여부를 감시하는 것이 매우 어려웠다. 일례로, 거래담당자간 혹은 관리자간 거래정보를 공유하기 위해 관련 엑셀파일을 복사하는 과정에서 파일의 버전관리가 제대로 되지 않아 거래 정보가 손실되거나 신규 정보가 제대로 반영되지 않는 등의 문제가 발생하기도 하였다. 부정확한 거래정보는 거래담당자가 집계하는 손익과 사후처리담당 부서에서 공식적으로 집계하는 손익이 서로 달라지는 등의 심각한 오류를 발생시키기도 하였으며 이를 해결하기 위해 많은 시간

과 인력이 낭비되기도 하였다. 셋째, 엑셀파일을 이용해 파생상품의 가치를 계산하는 작업은 메인 거래시스템과 같은 전용 시스템을 이용하는 경우와 비교해 매우 많은 시간이 소요되었다. 이로 인해, 거래 시점마다 미래 시장상황 시나리오에 따라 미리 결과를 예측해 보는 등의 포지션 분석은 거의 불가능하였으며 일별로 당일 손익을 산정하기에도 버거운 상황이었다.

이처럼 엑셀을 이용한 일부 거래포지션의 관리는 A은행의 메인 거래시스템이 가진 한계를 보완하기 위한 것이었으나, 그에 못지않은 다른 문제들을 발생시켰다. A은행은 이를 해소하기 위해 기존 메인 거래시스템과 연동하면서 메인 거래시스템이 다루기 어려운 상품에 대한 포지션 관리를 지원하고 또한 은행의 자체 가치평가 모델의 구현 및 실행이 가능한 모델기반거래지원시스템 (Model-based Trading Support System; MTSS) 구축 프로젝트를 진행하였다.

#### 4.2 시스템 구축 목표

MTSS는 III장에서 살펴본 장외파생상품 가치평가모델기반 DSS로서의 시스템요구사항을 바탕으로 실무적 관점에서 다음과 같은 목표를 가지고 구축되었다.

첫째, 현재 메인 거래시스템과 엑셀파일로 분리 운영되고 있는 전체 포지션을 통합하여 일괄적으로 분석할 수 있는 기능을 제공해야 한다. 앞 절에서 언급한 바와 같이 포지션의 분리 운영은 그 자체로서 통합, 관리에 상당한 비용을 필요로 하며 나아가 분석의 일관성 결여 및 사용자 조작 오류나 부정행위 등의 위험마저 내포하고 있다. 따라서 MTSS는 전체 포지션의 통합 및

분석 과정을 최대한 자동화하여 최소한의 비용으로 관리될 수 있도록 하며, 동시에 사용자의 인위적 조작 가능성을 최대한 배제하여 운영상의 오류를 원천적으로 막을 수 있어야 한다.

둘째, 장외파생상품 시장의 특성에 따라 은행은 지속적으로 신규 이익 창출을 위한 새로운 형태의 신상품을 출시해야 하므로, MTSS는 출시되는 신상품을 구조화하고 관련 모델을 신속하게 개발, 반영하여 거래담당자가 활용할 수 있도록 지원해야 한다. 기존 메인 거래시스템은 이러한 신상품 추가와 같은 변경에 상당한 시간과 비용을 필요로 하였는데 이는 은행의 시장 내 영업 경쟁력 저해 요인으로서 MTSS 구축이 필요한 가장 중요한 이유 중 하나였다. 따라서 MTSS는 신상품의 구조적 특성과 거래담당자의 분석 요건을 최소한의 수정만으로 시스템 내에 구현할 수 있도록, 기술적으로 변화에 유연하게 대처할 수 있는 시스템 구조를 갖춰야 한다.

셋째, 이자율이나 환율과 같이 은행 포지션 평가의 기준이 되는 시장 지표들은 일중에도 지속적으로 변화하기 때문에 거래담당자는 시장 상황에 따른 보유 포지션 변화를 분석하고 그에 맞는 거래전략을 구사할 수 있어야 한다. 그러나 기존 엑셀파일을 이용한 계산은 파일 기반이라는 기술적 한계로 인해 상당한 시간을 필요로 하여 적시의사결정에 많은 어려움이 있었다. A은행은 이러한 성능부족을 하드웨어 측면에서 보완하기 위해 비교적 고성능의 개인컴퓨터를 지급하는 등 우수한 하드웨어 인프라를 갖추고 있었으나, 거래시스템의 기술적 한계로 인해 이를 충분히 활용하지 못하고 있었다. 따라서 MTSS는 이러한 하드웨어 성능을 충분히 활용하여 전체 포지션을 분석하는데 있어 거래담당자가 시장 상황

에 신속히 대처할 수 있을 정도의 성능을 발휘할 수 있는 구조를 갖춰야 한다.

다음 절부터는 이러한 구축 목표를 달성하기 위한 MTSS의 구성과 내부 아키텍처의 구현 방법 등에 대해 보다 자세히 살펴보고자 한다.

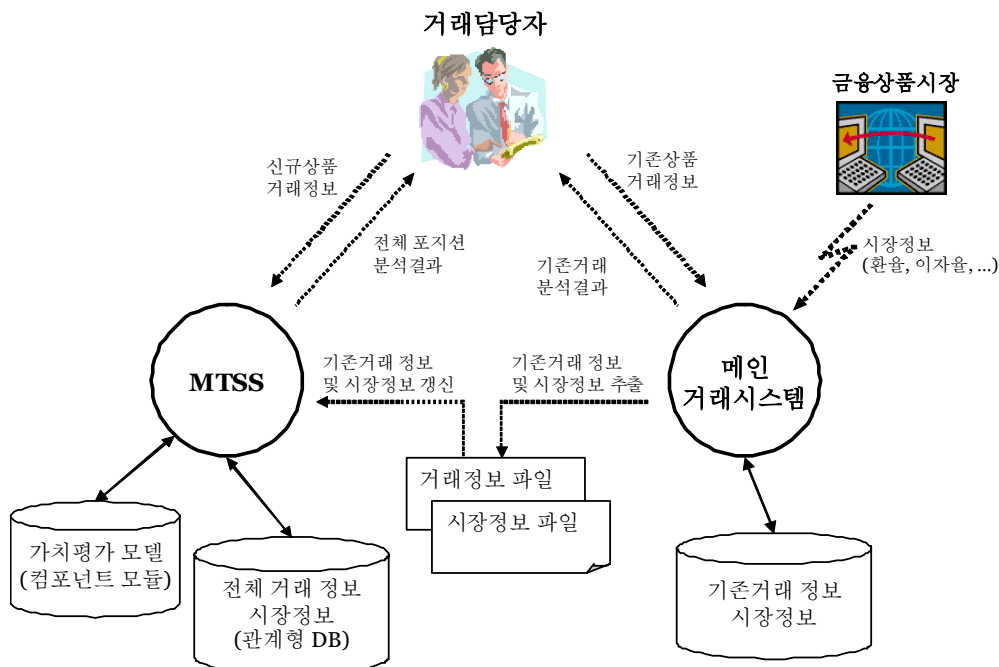
### 4.3 시스템 구성 방안

앞 절에서 살펴본 시스템 구축 목표 달성을 위해 가장 먼저 고려해볼 수 있는 대안은 기존 메인 거래시스템을 완전히 대체하면서 엑셀 파일로 관리했던 모든 형태의 거래에 대한 지원도 가능한 신규 시스템을 도입하는 것이다. 그러나 메인 거래시스템에 요구되는 데이터 안정성과 모델 활용을 위해 요구되는 사용자 편의성 및 시스템 유연성 등은 다소 상호배타적인 특성을 갖는 것으로 이를 모두 충족시킬 수 있는 단일 시스템의 구현은 그에 요구되는 시간과 비용의 제약을 고려했을 때 현실적으로 실행하기 어려운 작업이다. 그에 따라, A은행은 MTSS를 기존 메인 거래시스템의 보조적 역할을 담당하는 시스템으로서 정의하고, 메인 거래시스템 내 포지션의 통합 절차를 가능한 자동화하여 사용자 개입의 여지를 차단함으로써 포지션 분석에 대한 일관성을 최대한 보장하고 사용자 오류를 최소화 할 수 있는 시스템 구축을 추진하게 되었다. 따라서 MTSS는 메인 거래시스템에서 다루기 힘든 상품들은 자체적으로 관리하면서, 그 외 포지션정보나 시장 지표에 대한 정보 등 분석에 필요한 상당부분의 데이터는 메인 거래시스템으로부터 자동으로 제공받아 전체 통합 포지션을 분석하는 형태로 정의되었다. 이때, 분석의 기초자료인 이자율이나 환율과 같은 시장정보도 두 시스템

의 포지션에 대해 정확히 일치되도록 설정하여 일관성 있는 분석결과를 도출해야 한다. 이러한 시스템간 데이터의 통합과 분석은 거래담당자가 전체 포지션의 수익성 및 위험성 분석결과를 토대로 어떤 파생상품을 어느 정도 거래할 것인지 판단하기 위해 반드시 필요한 과정으로 분석이 필요한 시점에 적절한 성능으로 수행될 수 있어야 한다.

<그림 3>은 이러한 MTSS와 메인 거래시스템 사이의 데이터 전송 구조를 보여준다. 먼저, 거래담당자가 수행한 파생상품 거래 정보는 대상 상품이 무엇인가에 따라 메인 거래시스템이나 MTSS 내에 저장된다. 추가적으로, 메인 거래시스템은 로이터(Reuters)나 한국증권전산 등과 같은 외부 시장정보제공 업체로부터 포지션 분석에 필요한 시장정보를 실시간으로 제공받아

저장한다. 이렇게 저장된 메인 거래시스템의 거래정보와 시장정보는 거래담당자의 전체 포지션 분석요구가 있을 때 추출되어 MTSS로 전달된다. 이때, 데이터 전달 방식은 매개파일을 이용하는 형태로 구현되었다. 즉, 메인 거래시스템은 자신이 관리하는 거래정보 및 시장정보를 보고서 파일의 형태로 제공하고 MTSS는 이를 불러들여 해석하고 저장하는 방식이다. 이러한 매개파일을 통한 데이터의 전송은 그 과정이 다소 번거롭고 시간이 걸리기 때문에 실시간으로 데이터를 주고받기에는 적합하지 않다는 단점이 있다. 그러나, 기존 메인 거래시스템 상에 이와 같은 외부시스템의 데이터 추출을 위한 인터페이스가 존재하지 않았기 때문에 실시간 데이터 전송 모듈의 구현은 지나치게 많은 시간과 인력을 필요로 하였다. 또한, 장외파생상품 거래의 경우



<그림 3> MTSS와 메인 거래시스템의 통합 구조.

실시간 데이터 통합이 필요할 만큼 거래 빈도가 높지 않다는 점에서, 매개파일을 통한 데이터 전송 방식을 적용하는데 별다른 문제가 없었다. 이렇게 통합된 데이터를 기반으로 거래담당자는 MTSS가 제공하는 가치평가모형을 이용하여 전체 포지션에 대한 수익성과 위험성을 분석하고 그 결과를 거래에 활용한다.

#### 4.4 시스템 아키텍처의 구현

MTSS의 내부 아키텍처는 III장에서 설명한 장외파생상품 가치평가모형기반 DSS의 3계층 아키텍처에 입각하여 구현되었다. 먼저, 최상위의 사용자 인터페이스는 엑셀 스프레드시트를 이용하여 개발되었으며 <그림 4>는 인터페이스의 예를 보여준다. <그림 4>에서 보는 바와 같이,

사용자는 이미 친숙하게 활용하던 엑셀 프로그램을 사용하는 것과 유사한 형태로 MTSS의 기능과 분석 결과를 이용할 수 있다. 이때, 화면 구성을 위한 데이터의 추출 및 가공 작업은 엑셀 VBA를 이용한 매크로 함수(Wikipedia, 2010b)로 구현되었다. 스프레드시트 기반 사용자 인터페이스는 앞서 언급한 바와 같이 사용자가 시스템을 매우 친숙하게 활용할 수 있다는 장점을 가진다. 또한 사용자 관점에서 엑셀파일로 관리하던 기존 업무방식과 유사하게 보여지기 때문에 어느 정도 업무연계성을 유지하는 효과도 얻을 수 있었다. 시스템 개발 측면에서도 적은 비용으로 빠르게 구현할 수 있다는 장점이 있으며, 업무요건의 변경이나 시장상황의 변동 등에 따른 인터페이스 수정 요구도 쉽게 반영할 수 있다.

MTSS의 3계층 구조 중 두 번째인 비즈니스

The screenshot displays an Excel spreadsheet interface for 'FX Snowball'. It includes several data tables and reports:

- Deal Info Table:** Columns include Deal ID, Status, Product, Maturity, Marketer, Unwind Date, Currency, Data Rule, Settlement, Risk Class, and Product Code.
- Deal Type Table:** Columns include Deal Type, Par/Del, K/D, Window, Strike Reset, Min Strike, Max Strike, SP Start Num, and KO Point.
- Value Table:** Lists metrics like Value (551,635,804), Delta (2,396,792), Vega (224), Rho (36,869), Pp (46,931), and Theta (183,199,922).
- Greek Report Table:** A detailed report for '1. Greek보고' showing option and underlying greeks for various currencies (USD, EUR, JPY, KRW).
- 4. Time Bucket - BPV Table:** A table showing BPV values across different time buckets (1W, 1M, 2M, 3M, 4M, 5M, 6M, 9M, 11M) for various currencies.

<그림 4> MTSS 사용자인터페이스 예시.

로직 계층의 파생상품 가치평가모델이나 거래 정보 처리 절차 등의 업무로직 객체간 구조는 비주얼베이직(Visual Basic)을 이용하여 COM(Box, 1998) 컴포넌트 구조로 구현되었다. 이를 통해, III장에서 언급한 바와 같이, 개별 컴포넌트의 독립성과 높은 재사용성을 기반으로 업무 로직의 구현 및 변경이 쉽게 이루어질 수 있는 유연한 시스템 구조를 갖출 수 있었다. 비록, 비주얼베이직은 C++나 Java 등 본격적인 객체지향 프로그래밍 언어에 비해 컴포넌트 기반 객체지향구조가 갖는 상속성이나 다형성 등과 같은 객체지향 프로그램 메커니즘이나 COM 컴포넌트 기반 구조의 장점을 완벽히 구현하기에는 다소 한계가 있으나, 개발자가 쉽게 사용할 수 있는 형태의 언어로서 개발 기간이 크게 단축되고 유지보수 비용 또한 매우 적다는 장점이 있다.

반면, 파생상품 가치평가 모델의 실행에 필요한 실제 계산 모듈은 상당한 양의 계산을 필요로 하는 경우가 많기 때문에 시스템 구조의 유연성이나 개발의 편의성보다는 실행속도 자체가 매우 중요하다. 그러나 비주얼베이직은 프로그램 실행 성능보다는 개발자 편의성에 중점을 둔 언어로서 많은 계산을 수시로 필요로 하는 파생상품 분석 업무에 사용하기에는 너무 많은 시간을 요하는 경우가 많아 실무적인 활용에 한계를 갖는다. 또한, 시스템 운영체제나 하드웨어에 대해 직접적인 접근이 어렵기 때문에 은행의 전산 인프라가 가진 성능을 충분히 활용하지 못하는 한계도 존재한다. MTSS는 비주얼베이직의 이러한 한계를 보완하기 위해 많은 계산을 요하는 모듈은 C++를 이용하여 절차적(procedural) 프로그램(Wikipedia, 2010c)의 형태로 구현하고 멀티쓰레딩 프로그래밍(Wikipedia, 2011b)과 같은

시스템 프로그래밍 기법을 적용하여 충분한 계산 성능을 확보할 수 있도록 하였다.

#### 4.5 시스템 구축 결과

MTSS의 구축을 통해 A은행은 장외파생상품 거래업무에 있어 다음과 같은 세가지 주요한 개선사항을 얻을 수 있었다.

첫째, 기존 거래지원시스템과 엑셀 파일을 통해 분리 관리하던 거래포지션을 MTSS를 통해 통합하여 관리, 분석할 수 있게 됨으로써, 포지션 운영이 보다 용이해졌으며 과거 수작업을 통해 통합하면서 발생하던 오류들을 크게 줄일 수 있었다. 비록 실시간 통합이 아닌 매개파일을 이용하는 통합 형태이나, 대부분의 과정을 자동화함으로써 사용자 개입에 의한 오류 가능성을 최소화하여 전체 포지션에 대해 일관성 있고 신뢰도 높은 결과를 얻을 수 있게 되었다.

둘째, 새롭게 설계된 신규 파생상품이 출시되거나 기존 상품에 대한 분석 요건이 변경되었을 때, MTSS의 객체지향구조는 최소한의 수정만으로 이를 시스템에 반영할 수 있도록 함으로써 신속한 대응이 가능하도록 하였다. 단적으로 과거 기존 거래지원시스템에서 새로운 상품을 지원할 수 있도록 변경하기 위해서는 시스템 개발사의 도움을 얻는 과정과 함께 수개월 정도의 시간이 필요하였으나 MTSS에서는 수주일 정도의 노력으로도 가능하였다. 이러한 빠른 대응 능력은 신상품에 대해 시장 선점 가능성을 높임으로써 실질적인 거래영업이익의 증대에 기여하였다.

셋째, 엑셀파일을 통해 관리되던 포지션을 데

이터베이스 기반 전용프로그램인 MTSS 내에서 관리함으로써 전체 포지션 분석 성능을 크게 향상시킬 수 있었다. 엑셀파일을 통한 분석은 시시각각 변화하는 시장상황에 따라 전체 포지션의 가치를 평가하고 그에 따른 거래전략을 수립하는데 매우 많은 시간을 필요로 하였으나, MTSS에서는 엑셀 워크시트에서 계산하던 로직을 VBA나 C++와 같은 전문 프로그래밍 언어를 통해 구현하고, 멀티쓰레딩 프로그래밍을 적용하는 등 보다 효율적인 프로그램 구조로 개선하여 시스템 성능을 획기적으로 향상시킬 수 있었다. 예를 들어 환율파생상품 전체에 대해 특정 분석을 수행하기 위해서는 과거 여러 시간이 필요하였으나, MTSS에서는 이를 수분 이내로 단축시킬 수 있었다.

이러한 MTSS 구축을 통해 A은행은 전체 포지션 운영의 수작업을 최소화하고, 유지보수비용을 절감하였으며, 충분한 계산 성능을 확보하는 등의 업무 개선효과를 얻을 수 있었다. 이는 장외파생상품 거래업무의 효율성 증대로 이어져 결과적으로 시장에서 영업경쟁력을 강화할 수 있는 요인이 되었다.

## V. 결론

본 논문은 국내 대표적인 시중은행 중 하나인 A은행에서 수행된 모델기반 장외파생상품 거래 지원시스템(MTSS) 구축 사례를 통해 장외파생상품 거래업무에서 모델기반 DSS에 요구되는 사항들과 이를 충족시키기 위한 시스템 아키텍처의 설계 방법에 대해 논의하였다. 특히, 실제 시스템 구축 사례를 분석함으로써 제안 아키텍

처의 실무적 구현결과 및 실제 업무 개선 효과 등을 살펴보았다. 이로부터 본 논문이 제안하는 DSS 프레임워크의 의미는 다음과 같이 정리될 수 있다.

첫째, DSS 사용자가 쉽게 관련 모델을 실행하고 그 분석 결과를 업무에 적절히 적용할 수 있는 높은 사용자 편의성을 제공하였다. 이를 위해 MTSS에서는 주 사용자인 거래담당자나 모델링 전문가에게 매우 친숙하고 숙련도 높은 도구이자 정량적 분석 업무에서 폭넓은 범용성과 확장성을 제공하는 스프레드시트 프로그램을 기반으로 사용자 인터페이스를 구현하였다. 사용자는 스프레드시트를 통해 제공되는 포지션 분석결과를 쉽게 확인하고 이를 다시 다양한 방법으로 재가공해 활용할 수 있다. 이를 통해 시스템이 충분히 제공하지 못하는 정보나 특정 사용자에게 특화된 정보 등을 사용자가 직접 작성하는 방법으로 시스템 활용도를 확장할 수 있다. 또한 환경변화에 따른 분석방법의 변경이나 신상품 출시 등과 같은 요건 변경에 따른 변화도 범용의 분석도구라는 스프레드시트의 특성상 신속하게 반영할 수 있다. 이러한 높은 편의성은 사용자가 보다 적극적으로 시스템을 활용할 수 있게 하며, 사용자 입장에서는 다양한 관점의 분석을 통해 모델을 가장 효과적으로 적용하는 방법을 찾아갈 수 있는 토대가 됨으로써 최적의 거래의사결정을 내릴 수 있는 바탕이 된다.

둘째, 사용자 인터페이스 계층, 비즈니스 로직 및 해법 알고리즘 계층, 그리고 데이터베이스 계층 등으로 분화된 계층적 구조를 바탕으로 객체지향 개념에 따라 구성된 시스템 아키텍처는 시스템의 유지보수에 있어 매우 유연한 구조를 제공함으로써, 신규 상품의 개발이나 감독 규제의

변화 등 시스템 요건 변화에 대해 시스템이 효과적이고 신속하게 대처할 수 있도록 한다. 이러한 시스템의 빠른 시장적응력은 장외과생상품 거래에 있어 경쟁자에 앞서 시장을 선점할 수 있는 경쟁력으로 이어질 수 있기 때문에 실질적인 이익 창출의 바탕이 될 수 있다.

셋째, 기존 워크플로우 중심의 메인 거래지원 시스템과 통합, 운영되는 형태로 구현됨으로써, 데이터 처리 중심의 메인시스템과 모델 활용 중심의 MTSS가 상호 교류하면서 부서 전체 포지션을 일관성 있게 관리, 분석할 수 있는 시스템 환경을 제공하였다. 이러한 유기적 통합을 통해, MTSS는 안정적인 데이터 처리에 대한 요구와 모델사용자 편의성에 대한 요구라는 다소 상호 대립되는 요건 두 가지를 모두 만족시킬 수 있었다. A은행은 이러한 통합을 통해 과거 수작업 위주의 포지션 취합 과정에서 발생 가능했던 운영상의 오류나 사용자 과실 등의 위험을 크게 줄이고 보다 적은 노력과 시간으로 운영이 가능하도록 함으로써 업무 효율성을 높이고 직간접적인 운영비용의 절감 효과를 거둘 수 있었다. 향후 개선과제로는 현재 객체지향에 근거한 시스템 구조를 보다 확장하여 SOA(신정범 외, 2009; 최형림 외, 2010; Gold et al., 2004)에 기반하여 분산 환경에서도 다중 사용자의 다양한 거래분석 요구에 대응할 수 있는 분산 아키텍처로의 발전을 고려할 수 있다. 또한, 기존 워크플로우 중심 거래지원시스템과의 통합에 있어 현재의 매개과일에 기반한 일괄처리 방식을 보완하여 실시간 통합(Stankovic et al., 1999)이 가능하도록 함으로써 보다 빠르고 편리하게 전체 포지션을 분석할 수 있는 시스템의 구축을 고려해 볼 수 있다.

## 참고문헌

- 김준석, 김우상, "금융기관의 차세대시스템이 금융상품 개발 및 비즈니스 프로세스 단순화에 미친 효과: 기업은행 사례를 중심으로," *Information Systems Review*, Vol. 10, No. 2, 2008. 8, pp. 159-177.
- 박정희, 이기동, "A Framework of the Web-Based Knowledge Management Agent for Financial Decision Support System," *정보시스템연구*, 제15권, 제3호, 2006. 9, pp. 175-186.
- 신정범, 김재균, 장길상, "SOA 및 BPM 기반의 정보시스템 구축 방법론: 고객지향 수주 생산 환경에서의 제품 BOM 관리 적용 사례," *정보시스템연구*, 제18권, 제1호, 2009. 3, pp. 77-95.
- 최형림, 김현수, 홍순구, 박영재, 최성욱, 조민제, 박창현, 강시협, 안병선, "RTE 환경에서의 SOA 기반 협업적 분산 에이전트 개발," *정보시스템연구*, 제19권, 제1호, 2010. 3, pp. 79-96.
- Applegate, L.M., Konsynski, B.R., and Nunamaker, J.F., "Model Management Systems: Design for Decision Support," *Decision Support Systems*, Vol. 2, 1986, pp. 81-91.
- Blanning, R., "A Relational Framework for Join Implementation in Model Management," *Decision Support Systems*, Vol. 1, 1985, pp. 69-82.
- Booch, G., Rumbaugh, J., and Jacobson, I., *The Unified Modeling Language User*



- Guide, Addison-Wesley, Massachusetts, 1999.
- Box, D., Essential COM, Addison-Wesley, Massachusetts, 1998.
- Desai, S., "Are EXDBs better than RDBMSs for Model Management," Working Paper, Anderson Graduate School of Management, University of California, Los Angeles, 1991.
- Geoffrion, A.M., "An Introduction to Structured Modeling," Management Science, Vol. 33, 1987, pp. 547-588.
- Gold, N., Mohan, A., Knight, C., and Munro, M., "Understanding Service-Oriented Software," IEEE Software, March/April 2004, pp. 71-77.
- Gorry, A. and Scott-Morton, M.S., "A framework for information systems," Sloan Management Review, Vol. 13, No. 1, Fall 1971, pp. 56-79.
- Hull, J.C., Options, Futures, and Other Derivatives, Prentice Hall, New Jersey, 1997.
- Joshi, M., C++ Design Patterns and Derivatives Pricing, Cambridge University Press, 2004.
- Lee, K. and Huh, S., "A Model-Solver Integration Framework for Autonomous and Intelligent Model Solution," Decision Support Systems, Vol. 42, 2006, pp. 926-944.
- LeBlack, L.J., Randels, D.Jr., and Swann, T.K., "Herry International's Spreadsheet Optimization Model for Assigning Managers to Construction Projects," Interfaces, Vol. 30, 2000, pp. 95-106.
- Lenard, M., "Representing Models as Data," Journal of Management Information Systems, Vol. 2, 1986, pp. 36-48.
- Little, J.D.C., "Models and managers: The concept of a decision calculus," Management Science, Vol. 16, No. 8, April 1970, pp. 466-485.
- Ma, J., "An Object-Oriented Framework for Model Management," Decision Support Systems, Vol. 13, 1995, pp. 133-139.
- Ma, J., "Type and Inheritance Theory for Model Management," Decision Support Systems, Vol. 19, 1997, pp.53-60.
- Muhanna, W.A and Pick, R., "Composite Models in SYMMS," Proceedings of the 21st Hawaii International Conference on System Science, 1988, pp. 418-427.
- Power, D.J., "Supporting decision-makers: An expanded framework," 2001 Informing Science Conference e-book proceedings, Krakow, Poland, June 19-22, 2001, pp. 431-436.
- Power, D.J., A brief history of spreadsheets, DSSResources.COM, World Wide Web, <http://dssresources.com/history/sshistory.html>, version 3.6, August 2004.
- Power, D.J. and Sharda, R., "Model-driven decision support systems: Concepts and research directions," Decision Support Systems, Vol. 43, 2007, pp. 1044-1061.
- Raden, N., Shedding light on shadow IT: Is Excel

- running your business?, DSSResources.COM, 26 Feb. 2005.
- Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R., and Carlsson, C., "Past, Present, and Future of Decision Support Technology," *Decision Support Systems* Vol. 33, 2002, pp. 111-126.
- Sprague, R. and Carlson, E., *Building Effective Decision Support Systems*, Prentice Hall, 1982.
- Stankovic, J.A., Son, S.H., and Hansson, J., "Misconceptions about Real-Time Databases," *IEEE Computer*, June 1999, pp. 29-36.
- Turban, E., *Decision Support and Expert Systems*, Prentice Hall, 1995.
- Zelevnikow, J. and Nolan, J.R., "Using Soft Computing to Build Real World Intelligent Decision Support Systems in Uncertain Domains," *Decision Support Systems*, Vol. 31, 2001, pp. 263-285.
- Calypso, *Cross-Asset Front-to-Back*, <http://www.calypso.com/solutions/cross-asset-front-to-back.php>, 2010.
- FinCad, *Excel Solutions*, <http://www.fincad.com/derivatives-solutions/analytics/default.aspx>, 2010.
- MathWorks, *MATLAB - The Language for Technical Computing*, <http://www.mathworks.com/products/matlab/>, 2010.
- Misys, *Summit*, [http://www.misys.com/tcm/markets/otc\\_derivative\\_trading\\_solutions-summit.html](http://www.misys.com/tcm/markets/otc_derivative_trading_solutions-summit.html), 2008,
- Murex, *MX III*, <http://www.murex.com/mxIII.php>, 2009
- Numerix, *World-Class Solutions for Today's Challenges*, <http://www.numerix.com/products-and-services>, 2010.
- QuantLib, *A Free/Open-Source Library for Quantitative Finance*, <http://quantlib.org/>, 2010.
- Thomson Reuters, *Kondor Trade Processing*, <http://thomsonreuters.com/content/financial/pdf/enterprise/KondorTradeProcessing.pdf>, 2010.
- Wikipedia, *Proprietary Software*, [http://en.wikipedia.org/wiki/Closed\\_source\\_software](http://en.wikipedia.org/wiki/Closed_source_software), 16 Nov. 2010a.
- Wikipedia, *Visual Basic for Applications*, [http://en.wikipedia.org/wiki/Visual\\_Basic\\_for\\_Applications](http://en.wikipedia.org/wiki/Visual_Basic_for_Applications), 3 Nov. 2010b.
- Wikipedia, *Procedural Programming*, [http://en.wikipedia.org/wiki/Procedural\\_programming](http://en.wikipedia.org/wiki/Procedural_programming), 8 Nov. 2010c.
- Wikipedia, *Sensitivity Analysis*, [http://en.wikipedia.org/wiki/What-if\\_analysis](http://en.wikipedia.org/wiki/What-if_analysis), 14 Jan. 2011a.
- Wikipedia, *Multithreading (Computer Architecture)*, [http://en.wikipedia.org/wiki/Multithreading\\_\(computer\\_architecture\)](http://en.wikipedia.org/wiki/Multithreading_(computer_architecture)), 4 Jan. 2011b.

### 이근우(Keun-Woo Lee)



현재 우리은행 자금시장본부 트레이딩부서에 재직 중이며, 한국과학기술원 경영대학에서 경영공학 박사를 취득하였다. 한국경영과학회지, 인터넷전자상거래연구, Decision Support Systems 등 국내외학술지에 논문을 게재한 바 있다. 주요 관심분야는 SOA, Semantic Web, DSS, 금융상품 위험관리 및 트레이딩 시스템 등이다.

### 양근우(Kun-Woo Yang)



현재 계명대학교 전자무역학과 조교수로 재직 중이며, 한국과학기술원 경영대학에서 경영공학 박사를 취득하였다. 한국경영과학회지, 경영정보학연구, 통상정보연구, 해운물류연구, 인터넷전자상거래연구, 한국지능정보시스템학회논문지, 경영교육연구, Journal of Database Management, Expert Systems with Applications 등 국내외학술지에 논문을 게재한 바 있다. 주요 관심분야는 웹 2.0, 무역 프로세스 자동화, KMS 등이다.

<Abstract>

## **Design of Flexible DSS Architecture for OTC Derivatives Trading: 'A' Bank Case**

Keun-Woo Lee · Kun-Woo Yang

Model-based decision support system (DSS) has acted as a crucial role in strengthening the business competitiveness by providing a way of modeling and solving real-world decision problems in a quantitative and scientific manner. It is even more important for trading OTC derivatives, which requires extensive financial-engineering expertise while actively reacting to the continuously changing financial market. This paper proposes a flexible model-based DSS architecture that can support user-friendly interface for executing and analyzing the models and can adapt to the changes of financial market seamlessly. For user-friendliness, we implement the user-interfaces (UIs) using Microsoft Excel, a very widely used spreadsheet program for its great generality and extensibility. Users can utilize the analysis results of DSS or reprocess them for their special needs through the UIs in the form of familiar spreadsheets easily. For adaptiveness to the markets, the proposed architecture is constructed based on the object-oriented concepts, which enables such changes as release of a new financial product can be updated into the system without any delay at the lowest cost. We investigate the practical benefits and limitations of the proposed architecture by a case study on the construction of Model-based Trading Support System (MTSS), performed by a commercial bank in Korea.

**Keywords:** Model-Based Decision Support Systems, Financial Derivatives Trading Systems, System Architecture Design

\* 이 논문은 2011년 1월 24일 접수되어 1차수정(2011년 3월 2일)을 거쳐 2011년 3월11일 게재 확정되었습니다.