□특집□

Parameter Driven 방식과 금융 정보 기술

박 기 영†

◈ 목 차 ◈

1. 서 론

금융기관의 업무 비즈니스 로직은 금융기관마다 제각기 달라 보이지만 입금, 출금, 해지, 이자계산, 고객 판단기준, 예금 평잔 등 주요 금융 업무 프로시저는 세계적으로 공통적인 사항이다. 이렇게 어느 금융 시스템이든 동일한 골격을 가지고 있음에도 불구하고, 우리 특유의 금융 관행들 때문에 선진 금융 패키지의 도입을 꺼리는 경우가 발생한다. 때로는 금융 패키지에 대한 Localizeing, Customizeing 등의 이름을 붙여가면서 패키지를 도입하기도 하지만, 대체로 이것은 전면적인 재개발을 의미할 정도로 규모가 크다.

우리는 이것을 Parameter Driven 방식으로 이를 해결하고자 한다. 그러나 이것은 Customizing 단계에 앞서, 도입하려는 패키지에 기본적으로 포함되어 있어야 하는 하나의 사상인 것이다. 이 사상에 따라 처음부터 끝까지 일관성 있는 개발이 이루어진다면, 특수 관행이나 특정 계정과목에 얽매이지 않고도 세계적으로 표준화된 선진 금융패키지를 사용할 수 있는 것이다.

이 Parameter Driven이란 새로운 업무 추가에 대하여 프로그램을 매번 하드 코딩하지 않고 Parameter화 함으로써 소스 코드의 일부 수정만으로 즉각 대응할 수 있도록 만드는 방법이다. 더 나가서는 데이터베이스 테이블 내 데이터의 변경만으로도 새로운 업무 추가가 손쉽게 가능해져야 한다. 따라서 이 방법은 개발 설계 당시부터 반드시 고려되어야 할 사항이며, 어쩌면 자신들은 항상 이 방법을 채택해서 개발해왔다고 착각하는 수도 있다.

그럼에도 불구하고 아직도 이 방식은 최신 금융업무 개발기법의 하나로서 Component화와 더불어 차세대 시스템에서 계속 그 중요성을 인정받고 있다. 뿐만 아니라 국내에는 이미 Parameter Driven 방식으로 구현된, 적어도 두 개 이상의 Banking 패키지가 도입 구축된 사례가 있다.

그런데 이렇게 기본적인 사상으로 그다지 큰 의미가 없을 것 같은, 패키지들을 차세대니 신개념이니 하면서 금융기관에서 도입을 서두르는 데는 나름대로의 이유가 있다. 그것은 대부분 현재 사용하고 있는 금융시스템이 종합적으로는 Parameter Driven방식으로 구현되어 있지 않다는 점과, 만일 전 업무에 걸쳐 이 방식으로 Banking 시스템을 구축할 수만 있다면 많은 부분에 있어서 상당히 혁명적인 업무개선 효과를 거둘 수 있다는 사실 때문일 것이다.

† 정 회 원 : IMS시스템 금융시스템 담당 수석연구원

2. Parameter Driven 방식이 주는 개선 효과

우선 Parameter Driven 방식을 통한 Banking 업무 개발의 획기적인 개선효과는 수평적 사고에 의해 발생한다. 지금까지의 업무 개발은 수직적인 방법을 택했다. 수신, 여신 등으로 나누고, 이를 또 요구불, 저축성으로 나눈 후, 다시 이것을 각 계정과목별로 나누고 나서야 비로서 그 계정에 대한 업무 특성을 설계하고 이율, 해지 방법, 적수 계산, 세금 계산 등을 설계 후 개발하기 시작했다. 따라서 업무와 개발 경험이 풍부한 개발자들은 새로운 계정과목이 하나씩 탄생할 때마다 유사 프로그램을 복사하여 또는 중복되는 부분은 라이브러리화 시키고, 다양한 형태로 프로그램을 수정하는 식의 나름대로는 합리적인 개발 방법을 사용하였다.

그러나 반대로, 기능을 기준으로 한 수평적인 방식으로 다시 접근해보면, Banking 업무는 좀더 간단 명료하게 만들 수 있다. 수신업무의 경우를 예로 들면, 업무의 기능 또는 절차를 기준으로 하면 신규와 입출금 그리고 해지 등 크게 네 가지로 구분할 수 있다. 그런데 지금 여기서는 수직적, 수평적 사고 방식의 차이를 언급하려 함이 아니라 행과 열이 확장될 때 덜 확장되는 편을 기준으로 삼아야 한다는 것을 말하려는 것이다. 다시 말해서 수신 계정과목 종류가 네 가지 이내라면 현재의 방식이 합리적일 것이겠지만, 불행히도 현실은 절대로 수신계정이 네 과목 이하가 아닐 뿐만 아니라 계속 증가할 것이 예상되므로 기준을 기능 위주로 해야 하는 것이다.

업무 기능 또는 절차를 기준으로 하였을 때는 실제적인 은행 업무의 흐름이 중요하다. 뿐만 아니라 고객이 은행에 들어와서 나갈 때까지 흐름을 전산화 하는 것이므로 지금까지의 계정 과목별로의 프로그램보다는 업무 위주로 되어 있어 합리적인 면도 갖추고 있다. 더욱이 중요한 것은 개발 방법상 수신업무는 네 개의 커다란 프로시저만 있으면 업무개발이 가능하게 된다는 점이다. 각각의 서브 기능은 자세히 들여다보면 모든 수신 업무의 공통 사항에 불과하다. 즉, 이율 및 적수 계산 등이 없는 예금은 없으며 그 계산 방법도 몇 가지에 한정된다. 따라서 약간의 차이에 대한 부분만을 옵션으로 선택하게 해 주면 복잡해 보이던 은행 수신업무 전체가 단번에 간단해지면서 마치 한 사람이 프로그램한 것처럼 Logic 설계가 가능해진다. 이와 같은 Logic 구사는 Parameter Driven 방식이 아니면 생각할 수 없다.

3. Parameter 개발 적용 사례

다음과 같이 Parameter 방법을 개발에 적용하여 완성된 금융시스템을 소개하고자 한다.

3.1 개념 정의

Parameter Driven 방식이란 모든 업무를 분석하여 정규화하고, 이 Parameter를 기능 및 과목별로 공통된 속성들을 추출하여 테이블로 관리한다. 새로운 상품 개발 및 상품속성 변경 시에도 응용프로그램을 수정하지 않고 Parameter 만을 변경하여 적용되도록 설계하여, 자신의 비즈니스 Logic을 환경변화에 신속, 유연하게 자동 대처하도록 한다.

3.2 상품 특성 정의

상품의 일반적 특성 정의는 상품 Parameter에서 정의하고 이자 및 수수료 등은 별도의 해당 Parameter에서 정의하는데 이의 입력/정정/삭제는 Parameter 파일 유지보수 시스템을 통해 등록한다.

3.3 상품 특성 적용

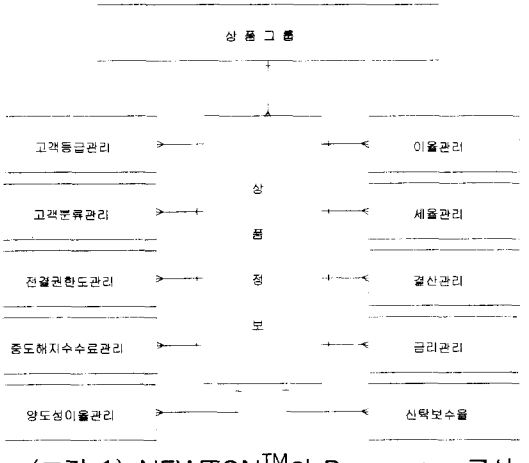
거래가 발생되면 각 응용프로그램은 처리를 하는 거래 요구계좌에 해당되는 상품 정보를 취하

여 신규/입금/지급/해지 등의 각 기능별로 어떤 유형을 적용하는 지의 여부를 해당 Parameter를 통해 알 수 있게 하며 이를 이용해 처리 유형을 결정할 수 있게 한다.

3.4 처리

각 거래의 처리는 주로 상품과 거래 Parameter에 의해 처리가 통제되는데 각 응용 시스템에서 필요한 각 기능별 모듈은 이미 존재하므로 새 상품의 적용으로 인해 처리 프로그램의 변경을 요구하지 않는다. 다만, 이 각각의 처리 모듈에서 상세한 처리 유형정보를 필요로 할 때 Parameter 파일에서 그 정보를 가져와 처리할 뿐이다. 이 때 아주 예외적으로 현존하는 상품 기능의 종류에 추가적 기능 유형이나 새로운 거래형태를 필요로 하는 상품의 경우에는 해당 기능 모듈에 약간의 코딩 추가가 요구될 수도 있다. 그러나 흔히 일어나는 새로운 이율 및 수수료 체계, 자격, 한도, 세율적용, G/L 집계 등이 기존의 상품과 다른 새로운 상품들의 적용은 몇 가지 Parameter 등록 만으로 가능하다

3.5 Parameter 구성도



(그림 1) NEWTON^{TM}의 Parameter 구성

3.6 수신 Parameter 테이블

1) 상품 그룹정보 관리

REFERENCE명	DESCRIPTION	TYPE	VALUE ITEMS
TGRP	상품그룹	CHAR	▶코드표 참조
STA	상태코드	CHAR	00:정상 09:삭제
DESCR	설명	CHAR	
TX_DATE	최초 등록일	CHAR	
TELR	최초 등록자 TELLER ID	CHAR	
CH_DATE	최종 변경일	CHAR	
CH_TELR	최종 변경자 TELLER ID	CHAR	

2) 상품정보관리

수신상품에 대한 과목별 기본특성, 속성 및 유형별로 코드화하고, 각종 제한사항을 정의하여 거래발생시 각각의 Application에서 상품 Parameter를 사용함으로써 업무처리에 효율성을 더해준다.

3) 이율정보관리

수신 상품에 대한 전반적인 이율을 관리하는 Parameter이며 은행별, 또는 지점별로도 이율 입력이 가능하며, 전지점 및 해당지점 적용여부 구분도 가능하다. 이율 입력시 대상과목은 대표과목이 아닌 소과목별로 입력하여 관리한다. 또한 금액별(MMDA) 이율관리도 할 수 있어 이자계산시 효과적인 업무처리를 할 수 있다.

4) 고시금리정보관리

5) 세율정보관리

이 Parameter는 국가에서 정한 각종세율을 관리하고, 과세코드별로 각각의 세율을 입력할 수 있도록 되어 있어 업무처리시 효과적으로 이용되고 있다. 또한 세율변동시 변동된 세율정보를 보관하므로 변경내용을 확인할 수 있다.

6) 수수료정보관리

이 Parameter는 수신업무에서 사용되는 수수료를 입력하여 관리하는 Parameter이다. 수신업무에서 발생하는 거래 및 타행환거래, 자기앞수표발행등에서 발생되는 수수료를 관리하므로 업무처리의 편리를 도모한다.

7) 타점권정보관리

이 Parameter는 각종 타점권을 입력하여 관리하는 Parameter이다. 타점권종류별로 코드에 대한 설명 및 부리기산일, 자금화일수, 미결제 타점권에 포함 여부 등이 입력되어 있어 타점권 업무처리시 사용된다.

8) 특별정보관리

이 Parameter는 주의 및 사고 신고등의 내용을 관리하는 Parameter이다. 수신업무 온라인 거래시 입력된 내용을 참조하여야 하는 경우가 발생하면 반드시 입력된 내용 여부를 확인하여 업무처리를 함으로서 사고 등의 문제 발생을 미연에 방지 할 수 있다.

9) 지로수납기관관리

10) 이 Parameter는 금융결제원에 등록되어 있는 지로수납기관번호를 받아 입력하고 변경사항이 있는 경우 변경내용을 정정하므로 지로업무 처리시 효율성을 극대화 하기 위해 사용된다.

11) 이체수납기관관리

12) 수신각종코드관리

13) 소득세추징세율관리

14) 결산일관리

이 Parameter는 상품별로 예금 및 대월결산일을 관리하는 Parameter 이다.수신에서 결산관련 프로그램 가동시 Parameter에 등록된 내용을 참조하여 결산 여부를 판단한다.

14) 결산진행절차관리

15) 결산용 그룹관리

4. 금융 개발 기술의 전망

Parameter Driven 방식이 주는 장점은 매우 크다. 앞에서의 은행 수신업무에서 이자 및 적수 계산 등의 부분은 계속 복잡하게 존재한다. 그런데 이런 부분을 기능화 시켜서 아이템 별로 집중적으로 모듈화 하는 것이 가능하다. 이러한 모듈화 기법은 최근의 Component화 기법과도 밀접한 연관을 갖게 되는데 이것은 마치 하나의 부품처럼 클래스화 함으로써 이를 불러 이용하는 주 프로그램에 질서정연하고 손쉬운 하나의 툴로서 제공하는 것이다.

따라서 앞으로 Banking 분야에서는 Component만을 골라서 개발하는 전문 개발자와 특정 업무 Logic 만을 이해하는 전문업무 컨설턴트의 등장을 예상할 수 있다. 현재의 은행 전산실에서 볼 수 있는 것처럼 업무와 전산 경험을 병행하는 형태는 금융업무 전용 패키지를 도입함으로써 점차 역할을 잃게 될 것이다. 즉, IT 분야의 발달로 현업의 요구 사항을 듣고 Parameter 등록, 또는 Component를 원하는 대로 재조립하면 새로운 업무 개발이 가능하기 때문이다.

그나마 여기서의 업무 개발의 의미는 어쩔 수 없이 프로그램 개발을 해야 하는 경우에 사용하는 Component 조립이라는 뜻이다. 실상은 프로그램 개발 이전에 금융 패키지가 제공하는 Parameter 테이블 내의 값을 변경하는 것만으로 끝나게 되어 있다.

그렇지만 금융기관에서의 주요 업무에 대하여 단순 모듈화 차원을 벗어나 사실상의 Component 개념을 도입한 개발은 당분간 생겨나기가 어려울 것이다. 그것은 우선 Component라는 개념은 JAVA와 같은 클래스 언어로 구현하게 되는데 현재로서는 JAVA가 코어 Banking 시스템을 감당할 수 있는지 입증되지도 않았기 때문이다. 따라서 합병이나 업무 추가로 인해 은행에서 보험이나 증권업무 취급하는 경우 등을 생각하면 몇 년 후를 내다보고 좀더 JAVA의 성능 향상을 기대해야 할 것이다.

5. 결 론

금융 전 업무의 획기적인 발전을 위해서는 향후의 Banking 시스템이 Parameter Driven 방식으로 다시 개발되어야 하는데 패키지 도입 방법을 제외하고는 직접 대대적인 개발을 하는 것은 쉬운 일이 아니다. 방대한 금융업무, 엄청난 소스라인 수 등에 의하여, 직접 개발한 모든 부분을 Document로 잘 남겨 놓았다 해도 그 많은 매뉴얼을 읽는 것마저도 낭비적인 요소마저 짙기 때문이다.

다행스럽게도 일부 Banking 패키지 회사들은 국내 은행에 대하여 Customizing을 끝낸 상태이다. Parameter Driven 방식이 갖고 있는 장점이기도 하지만, Banking 패키지들은 극소수의 코딩만 뒤따를 뿐, Parameter Driven사상을 변질시키지 않으면서, 일반적인 Banking 업무부분에 대해 Parameter 세트로 업무 구현이 가능하도록 한 것이다.

따라서 앞으로 금융기관의 IT 담당자들은 직접 프로그램을 개발하기 전에 항상 원하는 업무를 지원하는 Banking 패키지가 시장에 이미 존재하는지를 먼저 확인하여야 할 것이다. 또한, 동일한 말을 너무나 남발하여 좀 식상하겠지만, 이들 패키지가 말하는 Parameter Driven 방식이 자신들의 하고자 하는 업무에 어떤 효과를 주는지 분석하는데 주력하게 될 것이다.

박 기 영

1984년 세종대학교 경영학과
1984년-1988년 동아컴퓨터 금융 시스템 담당
1988년-1993년 한국 NCR UNIX 시스템 담당 과장
1993년-1998년 신기그룹 금융 시스템 담당 차장
1998년-현재 IMS 시스템 금융 시스템 담당 수석연구원