한국지능정보시스템학회논문지 제10권 제2호 2004년 11월(pp.39~51)

금융 프로덕트팩토리를 위한 복합상품 설계시스템의 개발

이성하
고려대학교 일반대학원 디지털경영학과
(lilyan@korea.ac.kr)

주정은
고려대학교 일반대학원 디지털경영학과
(kquilt@korea.ac.kr)

최성철
고려대학교 일반대학원 디지털경영학과
(laugh2r@korea.ac.kr)

구상회
고려대학교 일반대학원 디지털경영학과
(skoo@korea.ac.kr)

프로덕트팩토리는 온라인 고객을 위한 실시간 금융상품 설계시스템을 의미한다. 복합상품이란 두개의 서로 다른 상품의 속성을 적절히 결합하여 설계한 새로운 상품을 의미한다. 복합상품프로덕트팩토리는 위 두 개념을 결합한 것으로서 두개의 서로 다른 상품을 비즈니스 규칙과 사용자의 요구에 근거하여 적절히 결합하여 새로운 복합상품을 설계하는 프로덕트팩토리를 의미한다. 금융 및 보험 산업에서 자동화된 프로덕트팩토리의 필요성에 대하여 많은 논의는 있었지만 현재까지 이렇다 할 연구 성과는 보이지 못하고 있는 실정이다.

본 연구에서 우리는 복합상품을 설계할 수 있는 프로덕트팩토리를 개발하였다. 복합상품을 설계하기 위해서는 상품의 구조화된 표현 방법이 필요하며, 이렇게 표현된 서로 다른 상품의 속성을 결합하여 복합상품의 속성을 정의할 수 있는 속성결합방법이 필요하고, 또한 이들 속성의 결합방법을 제어하는 비즈니스 규칙을 표현하기 위한 방법이 필요하다. 본 연구에서 우리는 상품을 기술하는데 필요한 네 종류의 속성을 규명하였고, 이들 속성을 결합하기 위한 결합연산자를 개발하였으며, 또한 비즈니스 규칙을 표현하는 방법을 제시하였다. 또한, 본 연구의 효과를 검증하기 위하여 복합상품설계시스템을 구현하여 대출상품에 적용해 보았다.

논문접수일 : 2004년 3월 게재확정일 : 2004년 10월 교신저자 : 구상회

1. 서론

인터넷의 보급과 정보기술의 발달로 인하여 많은 업무가 온라인상에서 처리되면서, 고객의 온라인 업무에 대한 이해와 활용 그리고 필요성에 대한 인식도 점차 커지고 있다. 특히, 대부분의 업무가 물리적 물체의 이동이 아닌 순수한 정보의 흐름에 의해 수행되는 금융업무 처리에 대한 고객들의 불편함은 점차 커지고 있다. 현재 많은 은행 고객들은 단순한 현금 출금이나 계좌이체, 대금지불 뿐만 아니라 새로운 금융상품의 추천, 선택, 계약, 해지 등에 이르는 복잡다단한 금융업무를 온라인 실시간으로 처리하기를 원하고 있다(심철웅, 2001). 이러한 고객의 요구변화에 능동적으로 대처하기 위해서는 온라인 실시간으로 상품을 설계하고 제시해 주는 시스템이 필요하게 되었는데, 이를 '프로덕트팩토리(Product Factory)'라 한다. 프로덕트팩토리는 은행 직원의 개입없이 온라인

실시간으로 상품을 설계하고 생성하여 고객에게 바로 제공되는 자동화된 시스템을 의미한다(최성철, 2003).

최근에 발표된 프로덕트팩토리에 관한 연구로는 '맞춤형 온라인 금융상품 추천/설계시스템(최성철, 2003)'이 있다. 이 연구에서는 상품을 속성들의 집합(Set of Attributes)으로 정의된다고 보고, 이들 속성에 대해 값을 할당하는 것을 상품의 설계 과정으로 보고 있다. 즉, 이 연구에서 제시한 프로덕트팩토리는 상품을 정의하는 속성의 값을 결정하는 과정을 설계로 보는, "값 수준의 설계(Value Level Design)"라 볼 수 있다. 그러나 현재의 인터넷 고객은 특성과 취향면에서 서로 다르며, 이들의 니즈 또한 매우 다양하다. 이에 따라 속성값을 변화시키는 것만으로 복잡한 고객의 니즈를 만족시키는 상품을 설계하기란 쉽지 않다. 때로는 주어진 상품의 속성에 새로운 속성을 추가하거나 더 이상 필요치 않은 속성을 제거할 수도 있어야 하며, 아예 존재하지도 않았던 새로운 속성을 창조해서 추가할 수도 있다. 즉, 값을 결정하는 수준을 넘어서는 설계기능을 필요로 한다. 본 연구에서는 금융상품의 설계를 대상에 따라 세 종류로 나누어 본다.

첫째, "값 수준의 설계(Value Level Design)"는 상품의 정의에 필요한 속성집합에 변화를 주지 않고, 단지 이들 속성의 값만 설정함으로써 상품을 설계하는 것으로 가장 기본적인 프로덕트팩토리라 할 수 있다.

둘째, "속성 수준의 설계(Attribute Level Design)"는 속성의 값뿐 아니라 속성의 집합에 변화를 주면서 상품을 설계하는 과정으로, 상품의 정의에 새로운 속성을 추가하거나 이미 존재하는 속성을 제거하는 기능을 가진 설계과정을 의미한다. 예를 들면, 대출상품의 설계 시 설계기능으로 보증인이라는 속성 대신 담보설정 속성을 추가하거나, 신용도 등을 고려하여 보증이나 담보 속성을 삭제해 버릴 수가 있는데 이러한 경우, 이 프로덕트팩토리는 속성 수준의 설계기능을 갖는 시스템으로 볼 수 있다.

셋째, "구조 수준의 설계(Structure Level Design)"는 이미 존재하는 속성에 대한 추가와 제거뿐 아니라, 필요에 따라 지금까지 존재하지 않았던 전혀 새로운 속성을 추가하면서 상품을 설계하는 과정을 의미한다. 예를 들면, 인터넷 거래가 활성화됨에 따라 '인터넷 통장'이라는 것이 이미 사용되고 있는데 이 예금상품은 종이로 된 통장없이 전자화된 통장만 존재하고, 이에 따른 비용절감 부분을 고객에게 다양한 형태로 돌려줄 수 있는 상품이다. 이러한 상품의 설계를 구조 수준의 설계라고 볼 수 있다. 지금까지의 연구를 살펴보면, 현재의 프로덕트팩토리는 값 수준의 설계에 머무르고 있어, 현재 인터넷 고객의 다양한 니즈를 만족시키기 어렵다(최성철, 2003).

본 연구의 목적은 "복합상품을 설계할 수 있는 프로덕트팩토리 시스템의 개발"에 있다. 즉, 서로 다른 두 개의 상품을 고객의 니즈 측면에서 분석하여 두 상품이 가지고 있는 특성 중 고객의 니즈를 만족시킬 수 있는 특성들을 선택하거나 이들을 혼합한 속성을 가진 새로운 복합적 성격의 상품을 설계할 수 있는 시스템이다. 예를 들면, '학자금대출' 상품과 '생활자금대출' 상품을 결합한 '학생자금대출'이라는 복합상품을 새로이 설계하여 제공할 수 있는 시스템이다. 이렇게 설계된 상

품은 '학자금대출' 상품의 속성집합과도 다르고 '생활자금대출'과도 속성집합이 다른 새로운 상품으로 설계가 될 수 있다.

따라서 이는 값 수준의 설계에 머물러 있는 지금까지의 프로덕트팩토리보다 한 단계 앞선 기능을 가진 시스템으로 속성 수준의 설계시스템으로 볼 수 있다. 속성 수준의 프로덕트팩토리는 개별 고객의 복잡하고 다양한 니즈를 만족시킬 수 있는 상품을 설계하여 제공해 줄 수 있을 뿐 아니라, 은행 측면에서도 해당 상품의 부재로 인한 기회비용을 감소시켜줄 뿐만 아니라 고객의 만족도를 높여줌으로써 고객의 이탈방지 및 신규 고객의 유치에 효과를 기대할 수 있다.

2. 배경연구

본 연구는 '맞춤형 온라인 금융상품 추천/설계 시스템(최성철, 2003)'의 후속 연구로서, 값 수준의 설계를 속성 수준의 설계로 확장하여 고객의 니즈를 보다 효과적으로 만족시킬 수 있는 상품 설계방안을 개발한 연구이다. 본 연구와 선행연구는 성격상 지능형 설계(AID: AI in Design)의 범주에 속한다. 본 장에서는 본 연구의 배경이 되는 지능형 설계 관련 연구와 본 연구의 선행 연구인 '맞춤형 온라인 금융상품 추천/설계 시스템'에 대하여 설명한다.

설계(design)란 주어진 목적을 구현하기 위한 조형물(artifacts)의 구성 요소를 합리적으로 선택하고 선택된 요소를 올바르게 조합하는 방법을 계획하는 과정이다(Caplan, 1984, 노재호, 2000). 지능형 설계(AID: AI in Design)란 인공지능 기법을 적용하여 특정 영역에서 발생되는 설계 문제를 해결하기 위한 기술로서(Brown, 1998), 설계안을 찾기 위해 탐색이나 제약만족기법 등이 활용되고 있으며, 디자인 과정의 학습을 위해 기계학습 기술이 활용되며, 디자인 협업을 위해 다중에이전트 시스템이나 사례기반추론 등이 활용되고 있다(Chandrasekaran, 1990). 또한 지능형 설계에서는 설계문제를 설계과정의 복잡도에 따라 반복적 디자인(Routine Design), 혁신적 디자인(Innovative Design), 창조적 디자인(Creative Design) 등으로 나눈다(Brown, 1998). 반복적 디자인은 지식이라든가 문제 해결 전략을 미리 알고 있는 일상적이며 반복적인 설계를 말하고, 혁신적 디자인은 지식만 어느 정도 알고 있으며 디자인 과정에 대한 전략적 지식은 가지고 있지 않은 상태에서 수행하는 디자인을 말한다. 창조적 디자인은 지식과 전략에 대한 정보가 전혀 없는 상황에서 행해지는 디자인이다. 현재까지의 지능형 설계에 관한 연구를 살펴보면 주로 반복적 디자인에 집중 되어 있으며, 혁신적 디자인에 관한 연구도 부분적으로 수행되고 있다(Chandrasekaran, 1990).

본 연구의 목적은 개별 고객의 니즈를 만족시키는 맞춤형 금융상품 설계시스템의 개발에 있다. 금융상품이란 대출 종류, 대출금, 이자 등과 같은 기호형 또는 수치형 속성에 의하여 정의된다. 설계시스템이 갖추게 될 설계과정이란 이들 속성 사이에 존재하는 제약조건을 만족시키는 속성의 조합을 찾는 문제로 단순화된다. 따라서 본 연구의 설계 문제는 여러 가지 지능형 설계 기술 중에서 사례기반추론이나 제약만족기법이 적절히 사용될 수 있으며, 또한 설계 과정은 금융상품을 정의하는데 필요한 다양한 속성 값을 채우는 과정

으로, 고객의 요청이 있을 때마다 반복적으로 발생하므로, 혁신적 디자인이나 창조적 디자인보다는 반복적 디자인 문제에 가깝다.

맞춤형 금융상품 설계시스템이란 온라인으로 접근하는 고객의 요구사항을 고려하여 고객에게 가장 적합한 금융상품을 실시간으로 설계해 제공하는 시스템이다(최성철, 2003). 즉, 개별고객에게 맞는 금융상품을 온라인상에서 실시간으로 추천해 주고 적절한 상품이 없을 경우 바로 설계하여 제시해 주는 시스템이다.

이 연구에서 제안한 시스템은 고객의 요구사항에 적절한 상품유형을 결정하는 모듈, 결정된 유형의 사례를 모아놓은 사례베이스로부터 최적의 상품을 검색하여 추천하는 모듈, 만약 최적상품을 추천할 수 없을 경우 새로운 상품을 설계하여 제시하는 모듈(김영지, 2002), 사용자가 여러 가지 조건에 변화를 주면서 다양한 상품을 비교한 후 하나의 상품을 선택할 수 있도록 하는 의사결정지원 모듈 등의 4가지 모듈을 포함하고 있다. 이 연구에서는 이들 모듈의 구현을 위해 인공지능에서 개발된 여러 가지 기술을 정교하게 엮어 아키텍처를 구성하였다. 상품유형 결정 모듈은 RBS(Rule-Based System)로 구현했고, 구체적인 상품의 추천을 위해서 CBR(Case-Based Reasoning)을 활용하였으며(김승욱, 1997), 새로운 상품의 설계를 위해서 CS(Constraint Satisfaction) 기법을 적용하였고, 의사결정지원을 위해서 TMS(Truth Maintenance System)(Forbus and deKleer, 1993)에 기반한 What-if 분석 기법을 이용하였다.

3. 복합상품 설계시스템

본 연구의 목적은 온라인상에서 고객의 요구사항을 분석하여, 이를 만족시키는 복수개의 상품을 선택한 후, 이들 상품의 속성을 결합시켜 고객의 요구를 적절히 반영하는 하나의 복합상품을 만드는 것에 있다. 복수개의 상품을 결합하기 위해서는 상품의 기술에 필요한 속성과 속성 값들을 표현하는 방법(Attributes Representation)이 개발되어야 하며, 이들 속성값들을 고객의 니즈에 맞도록 효과적으로 결합하기 위한 연산자(Attribute Composition Operators)가 개발되어야 하고, 속성값의 결합을 제어해 주기 위한 비즈니스 규칙(Business Rules)을 표현할 수 있는 방법이 개발되어야 한다. 이를 요약하면 다음과 같다.

① 속성과 속성값들을 표현하는 방법을 정의한다. 상품을 속성들의 집합으로 볼 때, 각 속성이 가질 수 있는 값에는 수치값(Numeric Value), 구간값(Interval Value), 기호값(Symbolic Value), 나열값(List Value)이 있다. 본 연구에서는 속성값을 이와 같이 분류하고 각각의 표현방법을 제시한다.

② 위에서 정의된 각 속성값들을 결합하기 위한 연산자를 개발한다. 연산자란 동일 속성에 대한 서로 다른 두 개의 상품이 갖는 값들을 결합하여 설계할 상품의 속성값을 결정하는 방식을 정의한 것이다. 결합방식은 속성이 가지는 값의 종류에 따라 다양하게 개발한다.

③ 속성값의 결합을 제어하는 비즈니스 규칙을 만든다. 규칙이란 어떤 속성에 대하여 어떤 연산자를 적용할지를 정의하는 것이며, 이들 규

칙은 금융업무 영역에 대한 의존적 규칙 즉, 비즈니스 규칙(Business Rules)(Ross, 1998, Boston, 2002)으로서 표현방식은 일반적인 규칙기반시스템(Rule-Based System)의 표현방식에 따르며, 규칙의 생성은 금융업무 영역의 전문가의 자문을 구하여 만든다(Boston, 2002, Ross, 2003).

본 장에서는 본 연구에서 제시하는 상품 기술방법을 정의하고 각 상품의 속성값, 연산자(operators) 그리고 규칙(rules)의 기술방법에 대하여 설명한다.

3.1 상품과 속성의 표현

상품은 설계과정의 산출물로서 일련의 속성들을 조합하여 표현할 수 있으며, 다음과 같이 정의된다.

$$P = \{a_1=v_1,\ a_2=v_2,\ a_3=v_3,\ \cdots,\ a_i=v_j\}$$

위 정의에서 a는 속성(attributes)을 나타내며, v는 그 속성에 부여되는 값(values)을 나타낸다. 이들 속성과 값들의 집합이 하나의 상품을 정의한다. 본 연구에서는 이들 속성이 갖는 값을 표현하기 위한 몇 가지 방법을 개발하였다. 예를 들어, '학생직장인대출' 상품을 본 연구에서 개발한 방법에 의하여 표현하면 다음과 같다.

```
학생직장인대출 = {자금용도 = (학자금, 생활자금),
                  담보유형 = (주택),
                  직업 = 직장인,
                  직업 = 학생,
                  나이 = [20, 65],
                  소유금융상품 = (예적금, 보험, 신탁, 부금),
                  대출한도 = 35,000,000,
                  대출금리 = 5.97,
                  대출기간 = 3}
```

위의 예에서 '학생직장인대출'이라는 금융상품은 '자금용도', '담보유형', '직업', '나이', '소유금융상품', '대출한도', '대출금리', '대출기간'이라는 속성으로 구성되며, 이들 속성은 (학자금, 생활자금), (주택), '직장인', '학생', [20, 65], (예적금, 보험, 신탁, 부금), 35,000,000, 5.97, 3 등의 값을 가진다. 여기서 '직장인', '학생' 등의 속성값은 기호로 된 값이며(*기호형 속성값*), (학자금, 생활자금), (주택), (예적금, 보험, 신탁, 부금) 등은 기호의 리스트로 된 값으로 이들 기호값 중 하나를 취할 수 있다는 의미이다(*나열형 속성값*). 즉, 위 예제에서 '직업=직장인'과 '직업=학생'은 '학생직장인대출'이 되기 위해서 두 조건 다 만족시켜야 하지만, '자금용도=(학자금, 생활자금)'에서 '학자금'과 '생활자금'의 관계는 둘 중 하나만 만족해도 된다는 의미이다. 따라서 속성 간의 쉼표는 '그리고', 속성값 사이의 쉼표는 '또는'을 의미한다고 볼 수 있다. 그리고 금리 5.97%, 기간 3년 등은 속성이 수치값을 갖는 경우이며(*수치형 속성값*), 나이 [20, 65]는 값이 일정 구간의 값으로 갖는 경우이다(*구간형 속성값*). 이와 같은 네 가지 유형의 속성값에 대하여 요약하면 다음과 같다.

① 수치형 속성값(Numeric Value) : 상품의 속성은 수치형 값을 가질 수 있다. 위의 예제에서 '대출한도'의 35,000,000은 수치형 값이다.

② 구간형 속성값(Interval Value) : 구간형 값은 나이 속성인 [20, 65]과 같은 것으로서 경계치의 개폐 여부에 따라 아래와 같이 네 가지 형태로 나타낸다.

$[\![l, u]\!] \equiv l \le v \le u$　　$[\![l, u) \equiv l \le v < u$

$(l, u]\!] \equiv l < v \le u$　　$(l, u) \equiv l < v < u$

③ 기호형 속성값(Symbolic Value) : '학생'과 같은 값은 기호형 속성값이다.

④ 나열형 속성값(List Value) : 때로는 속성값으로 몇 개의 값 중에서 하나를 취할 수 있는데, 이때의 속성값을 나열형 속성값이라 한다. (예 적금, 보험, 신탁, 부금) 중 하나의 값을 가지면 된다는 의미이다. 나열값은 소괄호를 이용하여 표현한다.

3.2 속성결합 연산자

본 절에서는 앞서 살펴본 속성들의 결합에 사용하는 속성결합 연산자에 대하여 설명한다. 연산자는 속성값의 경우와 마찬가지로 수치 결합 연산자(Numeric Operators), 구간 결합 연산자(Interval Operators), 기호 결합 연산자(Symbolic Operators), 나열 결합 연산자(List Operators)의 4가지의 형태로 나타낸다.

<표 1>은 본 연구에서 개발한 연산자에 대한 이해를 돕기 위한 예로서, '학자금대출'과 '주택담보대출'을 복합한 '학생주택담보대출'이라는 가상의 상품을 나타낸 것이다. 기존 금융상품에 의하면, 직업이 학생인 사람은 등록금 마련을 위한 '학자금대출'을 선택할 수 있다. 하지만 이 학생이 학자금뿐 아니라 생활자금도 필요하며, 현재 주택을 소유하고 있다고 가정하자. 이러한 경우 '학자금대출'만으로는 금액이 많지 않아 생활자금으로 활용할 수 없다. 하지만 주택을 담보로 추가적인 대출을 받는다면, '학자금대출'과 '주택담보대출'을 결합한 복합상품인 '학생주택담보대출'을 설계하여 제공할 수 있다.

이러한 복합상품의 장점은 '학자금대출'의 한도보다 훨씬 높은 금액을 대출받을 수 있으며, 때로는 학생신분을 이용하여 일반 담보대출보다 낮은 우대금리로 대출을 받을 수 있을 것이다. 또한, 은행의 입장에서도 여러 구좌를 하나로 통합함으로써 운영비용을 절감하며 동시에 고객만족도를 향

<표 1> 복합상품 예제

학자금대출	주택담보대출	학생주택담보대출
{ 자금용도={학자금}, 담보유형={연대보증인}, 직업=학생, 나이=[20,], 소유금융상품={예적금}, 대출한도=3,000,000, 대출금리=5.25, 대출기간=4 }	{ 자금용도={주택구입, 일반자금}, 담보유형={아파트, 주택, 기타부동산}, 나이=[20, 60], 소유금융상품={예적금, 신탁, 보험, 부금}, 대출한도=60,000,000, 대출금리=6.7, 대출기간=2 }	{ 자금용도={학자금}, 자금용도={주택구입, 일반자금}, 담보유형={아파트,주택, 기타부동산}, 직업=학생, 나이=[20, 60], 소유금융상품={예적금, 보험, 신탁, 부금}, 대출한도=60,000,000, 대출금리=5.97, 대출기간=2 }

상시킬 수도 있다.

위의 <표 1>에서 '자금용도', '담보유형', '소유금융상품' 등의 속성은 나열형 값을 갖고, '직업'은 기호형 값을 가지며, '나이'는 구간형 값을 그리고 '대출한도', '대출금리', '대출기간' 등은 수치형 값을 갖는다. 속성에 대한 결합연산자는 개별속성의 형태에 따라 개발되었다. <표 2>는 본 연구에서 개발한 모든 연산자를 요약한 것이며, 이들에 대하여 설명하면 아래와 같다.

① 수치 결합 연산자(Numeric Composition Operators) : 수치 연산자는 수치형태로 된 두 개의 속성값을 결합하는 연산자로서, 큰 값을 택하는 O_{max}, 작은 값을 택하는 O_{min}, 적절한 값을 선택하는 O_{sel}, 그리고 가중평균값(interpolated value)을 택하는 O_{int} 등이 있다. 이들 연산자 중 어떤 것을 선택할지의 여부와 가중평균 방식 등은 모두 비즈니스 규칙(Business Rules)으로서 영역전문가에 의하여 결정된다. 앞 예제에서 '대출한도' 값의 경우 결합 연산자로 O_{max}를 사용한 경우이다. 즉, '학자금대출'의 한도 3,000,000과 '주택담보대출'의 한도 60,000,000 중 큰 값인 60,000,000을 선택한 것이다.

② 구간 결합 연산자(Interval Composition Operators) : 구간 연산자는 구간값을 갖는 두 개의 속성을 결합하는 연산자로서, 두 구간의 합집합 값을 취하는 O_{cup}, 두 구간의 교집합 값을 취하는 O_{cap}, 두 가지 구간 중 하나를 취하는 O_{sel}이 있다. 이들 연산자 간의 선택규칙은 비즈니스 규칙으로 영역전문가에 의하여 결정된다. 앞의 예에서 '나이' 속성의 경우 결합할 두 개의 값이 각각 [20,](20세 이상)과 [20, 60](20세 이상 60세 이하)이다. 이때 O_{cap}을 적용하면 결과는 [20, 60]이 된다.

③ 기호 결합 연산자(Symbolic Composition

<표 2> 속성결합을 위한 연산자

속성	연산자		설명
수치형	수치	O_{max}	두개의 값 중 큰 값을 선택
		O_{min}	두개의 값 중 작은 값을 선택
		O_{sel}	두개의 값 중 한 값을 선택
		O_{int}	두개의 값의 보간값
구간형	구간	O_{cup}	두개의 구간의 합집합
		O_{cap}	두개의 구간의 교집합
		O_{sel}	두개의 구간 중 한 구간을 선택
기호형	기호	O_{and}	두개의 값을 모두 포함한 값 선택
		O_{sel}	두개의 값 중 한 값을 선택
나열형	나열	O_{and}	두개의 값을 모두 포함한 값 선택
		O_{or}	두개의 값 중 임의의 한 값만 선택
		O_{sel}	두개의 값 중 한 값을 선택

Operators) : 기호 연산자는 기호 형태를 취하는 두 개의 속성값을 결합하는 연산자로서, 두 값을 모두 만족해야 하는 O_{and}와 두 값 중 하나만 선택하도록 하는 O_{sel}이 있다. 예를 들어, '학자금대출'과 '직장인대출'을 결합하여 '학생직장인대출'을 만들 경우, '직업=학생'과 '직업=직장인'에 O_{and}를 적용하면, '직업=학생, 직업=직장인'이 되어 직업이 학생이며 동시에 직장인인 고객에게 대출되는 상품을 기술할 수 있게 된다. O_{and}와 O_{sel}사이의 선택과 O_{sel}에서의 선택방식은 모두 비즈니스 규칙으로 결정된다[1].

④ 나열 결합 연산자(List Composition Operators) : 나열 연산자는 나열형의 속성을 결합할 때 사용하는 연산자로서, 두 나열형 값을 모두 만족시켜야 하는 O_{and}과 두 나열형 값 중 하나만 만족해도 되는 O_{or}, 그리고 두 나열형 값 중 하나를 선택하는 O_{sel}가 있다. <표 1>의 예제에서 '자금용도={학자금}'과 '자금용도={주택구입, 일반자금}'을 결합하여 '자금용도={학자금}, 자금용도={주택구입, 일반자금}'이 나왔는데 이는 O_{and}를 적용한 경우이다. 또 같은 표에서 '소유금융상품={예적금}'과 '소유금융상품= {예적금, 신탁, 보험, 부금}'을 결합하여 '소유금융상품={예적금, 신탁, 보험, 부금}'이 되었는데, 이는 O_{or}를 적용한 경우임을 알 수 있다.

1) 기호 연산자의 경우, O_{or}와 같은 연산자는 정의되지 않았는데, 그 이유는 만약 이를 허용할 경우 '직업=학생'과 '직업=직장인'을 결합했을 때, '직업=(학생, 직장인)'이 되어 속성의 형태가 기호형에서 나열형으로 바뀌기 때문이다. 본 연구에서는 속성형의 변화를 허용하지 않기로 한다.

3.3 비즈니스 규칙

비즈니스 규칙(Business Rules)이란 앞서 설명한 다양한 연산자들 중 어떤 것을 선택할지 그리고 선택된 연산자를 구체적으로 어떤 방식으로 적용할지를 규칙의 형태로 표현한 것이다. 예를 들면, 수치형 속성값을 결합하는 연산자는 O_{max}, O_{min}, O_{sel}, O_{int} 로 구성되어 있는데, 이들 중 특정 상품의 특정 속성에 대하여 어떤 연산자를 적용할지에 대한 지식이 비즈니스 규칙이다. 그리고 O_{sel}의 경우 어떤 속성을 선택할지, 그리고 O_{int}의 경우에 구체적으로 가중치는 어떻게 결정할지 등에 관한 지식도 비즈니스 규칙에 해당된다. 이들 규칙은 영역전문가와 상담을 통하여 결정된다. 본 시스템은 전체적으로 규칙기반시스템(Rule-Based Systems)로 구축된다. 따라서 이들 비즈니스 규칙은 아래와 같이 'IF-THEN' 형식으로 표현하였다.

<표 3>은 복합상품 설계를 위한 비즈니스 규칙의 예이다. <규칙 1>은 '자금용도' 속성을 위한 비즈니스 규칙으로서, 나열 연산자인 O_{and}가 쓰인 예이다. 즉, 상품 1의 자금용도 값과 상품 2의 자금용도 값이 주어졌을 때, 두 값을 모두 만족시키는 값을 얻는 규칙이다. <규칙 2>는 '담보유형' 속성을 위한 비즈니스 규칙으로서, 기호연산자인 O_{sel}이 쓰인 예이다. 즉, 상품 1의 담보유형이 '아파트'이고 상품 2의 자금용도가 '무보증'일 경우, 담보유형에 대한 우선순위 규칙에 따라 '아파트'라는 값이 선택되는 것이다. <규칙 3>은 '나이' 속성을 위한 비즈니스 규칙으로서, 구간연산자인 O_{cap}이 쓰이는 예이다. 두 개의 구간값에 대해 겹치는 부분을 복합상품의 속성값으로 정하는 것으로, '20세 이상 65세 이하'인 값과 '20세 이상 70세

이하'인 값에서 겹쳐지는 교집합 값인 '20세 이상 65세 이하'를 새로운 값으로 산출해 내는 것이다. <규칙 4>는 '대출금리' 속성을 위한 비즈니스 규칙으로서, 수치 연산자인 (O_{int})가 쓰이는 예이다. 즉, 6.0라는 금리와 7.0이라는 금리가 주어졌을 때 두 값의 평균값인 6.5가 산출되는 것이다.

또한, 각각의 연산자는 별도의 비즈니스 규칙에 따라 다르게 적용 될 수 있다. '대출기간'의 경우, '주택'을 담보로 대출을 신청한 사람이라면 '주택'의 전세기간에 의거하여 대출기간이 바뀔 수 있다. 즉, 대출기간이 3년인 상품과, 주택을 담보로 하면서 주택의 전세기간인 1년이 대출기간이 되는 상품이 있다고 할 때, 별도의 규칙에 따라 O_{min}이 적용될 수 있다. 반면, '대출금리'의 경우, 직업이 학생이고 금융거래를 하고 있는 고객이면 O_{int}대신 O_{max}를 적용하여 더 많은 금액을 대출해 줄 수 있게 한다.

4. 복합상품 설계시스템의 구현과 사례

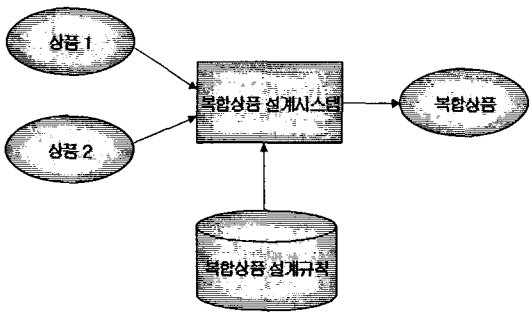
본 장에서는 본 연구에서 개발한 복합상품 설계시스템의 아키텍처와 구현에 대하여 설명한다.

4.1 시스템 아키텍처

본 연구에서 제안한 복합상품 설계시스템은 [그림 1]과 같은 구조를 갖는다. 즉, 복합 대상이 되는 상품에 대한 기술을 입력받아([그림 1]의 상품1과 상품2), 복합상품 설계에 사용되는 비즈니스 규칙을 근거로([그림 1] 중 복합상품 설계규칙), 이들 상품의 기술을 결합한 새로운 상품을 설계하여 제시한다([그림 1] 중 복합상품).

<표 3> 각 속성별 비즈니스 규칙 예제

규칙	연산자	적용사례
자금 용도	O_{and}	IF 자금용도$_1$ = x AND 자금용도$_2$ = y THEN 자금용도$_3$ = x, 자금용도$_3$ = y
담보 유형	O_{set}	IF 담보유형$_1$ = "예적금" AND 담보유형$_2$ = "부동산" THEN 담보유형$_3$ = "부동산"
나이	O_{cap}	IF 나이$_1$ = [l_1, u_1] AND 나이$_2$ = [l_2, u_2] THEN 나이$_3$ = [max(l_1, l_2), min(u_1, u_2)]
대출 금리	O_{int}	IF 대출금리$_1$ = x AND 대출금리$_2$ = y THEN 대출금리$_3$ = (x + y) / 2



[그림 1] 복합상품 설계시스템 아키텍처

4.2 시스템 구현

본 연구에서 개발한 복합상품 설계시스템의 프로토타입은 3장에서 설계한 내용을 바탕으로 구현하였다. 운영체제는 MS(Microsoft)사의 Windows XP Professional을 이용하였고, 시스템은 CA(Computer Associates)사의 CleverPath Aion Business Rules Expert 9.5를 이용하였다 (http://www.ca.com).

[그림 2]는 Aion으로 구현한 비즈니스 규칙의 예이다. 이 예들은 수치형 속성값을 결정하는 규칙이다. 예를 들어, LoanAmount의 경우 Product1과 Product2의 값 중에서 큰 값을 복합상품인 Product3의 대출가능한 금액으로 결정하는 결합 규칙을 나타낸다.

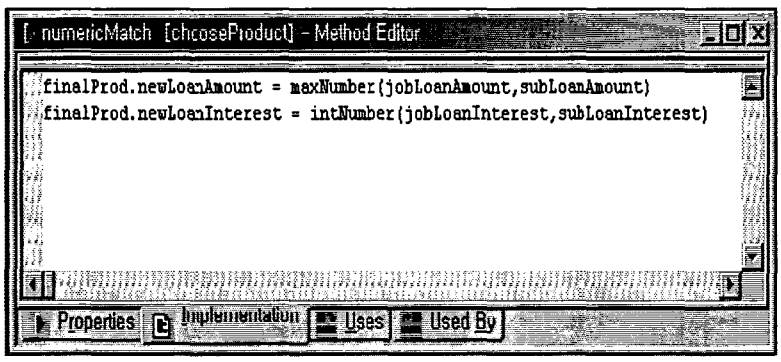
[그림 3]과 [그림 4]는 복합상품 설계의 예이다. 결합대상이 되는 두 개의 입력상품(ProductA, ProductB)을 나타낸 것으로 본 연구의 배경연구가 된 프로덕트팩토리에서 개인 정보를 입력하여 설계된 두개의 상품이다.

[그림 5]는 본 시스템의 비즈니스 규칙을 활용하여 설계한 복합상품을 나타낸 것이다.

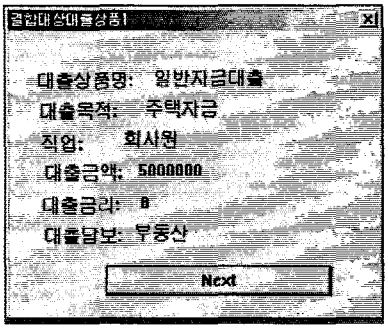
[그림 5]에서 대출목적은 '주택자금'과 '생활자금' 값에서 두 개값을 모두 취하는 O_{and} 연산자를 사용하여 '주택자금이면서 생활자금'인 값이 산출된 것을 보여주고 있다. 또한, 담보의 경우 '부동

<표 4> 시스템 구현 환경

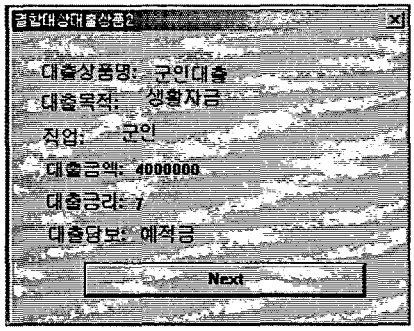
구 분	내 용
운영체제	MS Windows XP Professional
언 어	CA CleverPath Aion Business Rules Expert 9.5



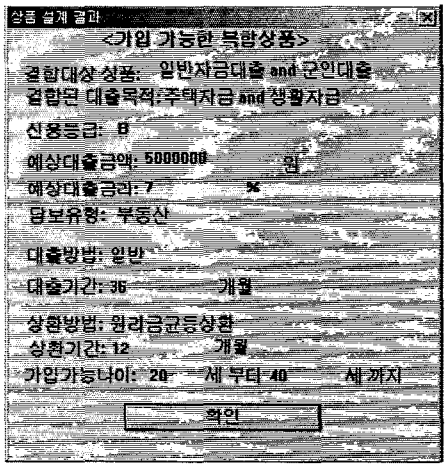
[그림 2] 비즈니스 규칙의 예



[그림 3] ProductA



[그림 4] ProductB



[그림 5] 설계된 복합상품

산'과 '예적금' 값 중 우선순위 규칙에 의거하여 '부동산' 값이 선택되었으며, 예상대출금액의 경우 '5000000'과 '4000000' 값 중에서 Omax 연산자를 사용하여 큰 값인 '5000000'이 산출되었다.

5. 결론

본 연구는 두 개의 상품을 비교 분석함으로써, 고객의 요구사항을 적절히 반영하는 복합상품을 설계하는 구체적인 방법을 제안하였다. 즉, 상품과 속성의 표현방법을 수치, 구간, 기호, 나열형으로 나누어 개발하였으며, 속성값들을 결합하기 위한 다양한 연산자를 수치형 속성, 구간형 속성, 기호형 속성, 나열형 속성에 대하여 개발하였다. 또한 결합을 제어해 주기 위한 비즈니스 규칙을 표현할 수 있는 방법을 개발하였다. 아울러 본 연구의 효과성을 입증하기 위하여 복합상품 설계시스템의 프로토타입을 구현하여 이를 대출상품에 적용해 보았다.

프로덕트팩토리에 관한 기존 연구가 속성이 갖는 값을 결정하는 수준의 설계에 한정하고 있는 반면, 본 연구에서는 설계의 대상을 새로운 속성의 추가 및 기존 속성의 제거까지 가능한 속성 수준의 설계 방법론으로 구현한 것이다. 본 연구를 활용하면, 기능 측면뿐 아니라, 고객만족 측면이나, 금융기관의 서비스 제공 측면에서 보다 향상된 프로덕트팩토리를 구현할 수 있을 것이다.

참고문헌

[1] 김승욱, "제약조건과 사례 기반 추론을 이용한 설계 지원 시스템 개발에 관한 연구", *서울대학교 박사학위논문*, 1997.

[2] 김영지, 문현정, 옥수호, 우용태, "사례기반추론 기법을 이용한 개인화된 추천시스템 설계 및 구현", *정보처리학회논문지*, Vol. 9-D, No. 6, 2002.

[3] 노재호, "디자인의 개념 및 역사", 2000.

[4] 심철웅, "국내외 은행들의 온라인 금융서비스 개발흐름과 시사점", *대은경제리뷰*, 2001.

[5] 최성철, "맞춤형 온라인 금융상품 추천/설계 시스템의 개발에 관한 연구", *고려대학교 석사학위논문*, 2003.

[6] Brown D., *Intelligent Computer-Aided Design*, WPI, 1998.

[7] Caplan R, *By Design*, McGraw-Hill Paperbacks, 1984.

[8] Chandrasekaran B, "Design Problem Solving: A Task Analysis", *AI Magazine*, Winter, 1990 pp. 59-71.

[9] Forbus, K. and deKleer,J, *Building Problem Solvers*, MIT Press, 1993.

[10] Morgan T. Boston, *Business Rules and Information Systems: Aligning IT with Business Goals*, Addison-Wesley Publishing, 2002.

[11] Ronald G. Ross, *Business Rule Concepts: the New Mechanics of Business Information Systems*, Business Rule Solutions, Inc., 1998.

[12] Ronald G. Ross, *Principles of the Business Rule Approach*, Addison-Wesley Publishing, 2003.

[13] http://www.ca.com, Computer Associates Home Page.

Abstract

A Hybrid Product Design System for Financial Product Factory

Seong-ha Lee · Jung-eun Ju · Seong-cheol Choi · Sang-hoe Koo*

Product factory is a real-time financial product design system for the Internet customers. The hybrid product is a product taking combined characteristics of two different products. Hybrid product factory is a product factory that designs hybrid products from two different products based on both business rules and customer requirements. Though the importance of product factory is emphasized in the industry, there has not been much research peformed regarding product factory.

In this research, we developed a product factory system that designs hybrid products. To design a hybrid product, it is necessary to have a method to combine attributes and values of two different products, and a method to control the combining operations to properly reflect business requirements. In this research, we developed four different combining operators and business rule representations. In addition, to prove the effectiveness of this methods, we implemented a prototypical system and demonstrated on cases regarding financial loan products.

Key words : 복합상품, 금융상품설계, 속성수준의 설계

* Department of Digital Management, Korea University Graduate School