

## 전자상거래에서의 벡터 공간 모델링을 통한 Configuration 시스템

김세형

인하대학교 전자계산공학과  
(shkarn@eslab.inha.ac.kr)

조근식

인하대학교 전자계산공학과 교수  
(gsjo@inha.ac.kr)

최근 전자상거래에는 일대일 마케팅이나 협력적 정보여과기법등을 이용한 다양한 추천서비스가 도입되고 있다. 이러한 추천 서비스의 형태는 다양한 제약 조건을 갖고 계산 복잡도가 높은 제품의 경우에는 고객을 만족시킬 만큼 적절한 추천서비스가 이루어지기 어려울 것으로 본다. 본 논문에서는 Clancey의 Classification Problem Solving 방법과 제약조건 기반 Configuration기술을 통합하여, 이러한 문제를 해결할 수 있는 방법을 제시하였다. 이 방법은 Clancey의 이론에 따라 구성 복잡도가 높은 제품의 해집합 도메인을 분할하여 문제의 복잡도를 줄일 수 있도록 하였으며, 여기에서 선택된 도메인을 제약조건 기반 Configuration기술에 적용시킴으로써, 구매자와 제품 컴포넌트 사이에 존재하는 제약조건을 처리할 수 있도록 하였다. 제약조건기반 Configuration기술은 구매자에게 적합한 제품을 구성하기 위해서 제약 조건 만족 문제(Constraint Satisfaction Problem; CSP)해결 기법을 이용한다. 또한 Clancey이론은 구매자의 만족도를 고려하기 위해서 정보검색 분야의 벡터공간 모델링 방법을 변형하여 적용하였다. 마지막으로 본 모델의 평가를 위해 전체 시스템의 수행시간 및 구매자 만족도를 비교 분석하였다

### 1. 서 론

전자 상거래 시장이 점점 활성화됨에 따라 고객에 대한 서비스의 형태는 빠르게 성장하고 있다. 구매자 개개인의 특성을 반영하지 못했던 초기의 불특정 다수에 대한 서비스는 일대일 마케팅이나 정보여과기법을 이용하여 구매자 개개인의 구미에 맞는 제품을 제공하는 서비스 형태로 발전해나가고 있다. 하지만 이러한 서비스 역시 전문적 지식을 필요로 하거나 구성 복잡도가 큰 제품의 구입에 있어 충분한 맞춤서비스를 제공하는 데에는 한계점을 갖게 되었다. 인공지능분야에

서 연구되어오던 Configuration기법을 변형하여 웹에 적용한다면 이러한 문제점을 어느 정도 해결할 수 있을 것으로 본다(12, 13, 14). Configuration기법은 요구 명세서에 따라 생성된 제약 조건을 이용해서, 미리 정의된 제품 컴포넌트에 대해 선택 및 재배치의 기능을 수행하여 구매자의 요구에 부합되는 제품을 만드는 기술이다 (Bernd, 1988). 이 기술은 구매자의 요구사항과 전문가의 지식으로부터 추출된 제약 조건과 도메인 제품의 컴포넌트에 대해서 인공지능의 제약만족 문제 해결기법을 적용시켜 최종적인 제품을 구성한다. 그러나, Configuration문제는 NP complete

한 문제이기 때문에 많은 수행 시간상의 비용을 요구한다. 이러한 점은 Configuration기술을 전자상거래에 적용하는데 갖는 어려움중의 하나이다.

본 논문에서는 Clancey가 제안한 Classification Problem Solving방법(Clancey, 1985)과 제약조건기반 Configuration기술(Fleischanderl et al., 1998)을 통합한 시스템을 구성하여 도메인 제품 구성의 복잡도를 최소화하고, 문제의 제약조건을 처리하여 구매자의 만족도 및 시스템의 수행속도를 향상시키는데 초점을 두었다. Clancey가 제시한 이론은 문제의 해집합 도메인의 추상화 작업을 통하여 선택 가능한 해집합 카테고리들 생성한 후에, 전문가의 경험적 지식을 이용하여 요구 데이터와 일치하는 카테고리를 선택하는 것이다. 여기에서 생성된 카테고리는 완전 상호 배타적이지 못하다는 특성이 있고, 카테고리의 선택을 위해 고객의 선호도 정보가 반영되어야 한다는 측면에서 본 논문에서는 정보 검색 분야의 벡터 공간 모델링을 이용하였다. 이 모델에서는 고객의 요구 데이터 및 제품 도메인에 대한 가중치를 벡터 공간상에 표현함으로써, 두 가지 문제점을 보완할 수 있었다. 한편 본 논문에서는 벡터 공간상에 표현된 해집합 제품 도메인을 제품 프로토타입 모델로써 정의하였다. 이러한 제품 프로토타입 모델은 각 모델에 적합한 컴포넌트들을 클러스터링 하여 갖고 있는데, 이때 클러스터링 되는 컴포넌트의 수를 적정 수준으로 조정하는 것은 제약조건 처리 모듈에서의 성능을 결정짓는 중요한 요소가 될 수 있다. 따라서 이 부분을 위해서 실험을 통하여 적정 수의 클러스터링 수를 조절해야 함을 보였다.

본 논문의 전개를 위해서 우선 Configuration이 전자상거래에 적용되기 위한 멀티 에이전트 구조를 정의해 보고, 전체적 수행과정을 살펴 본

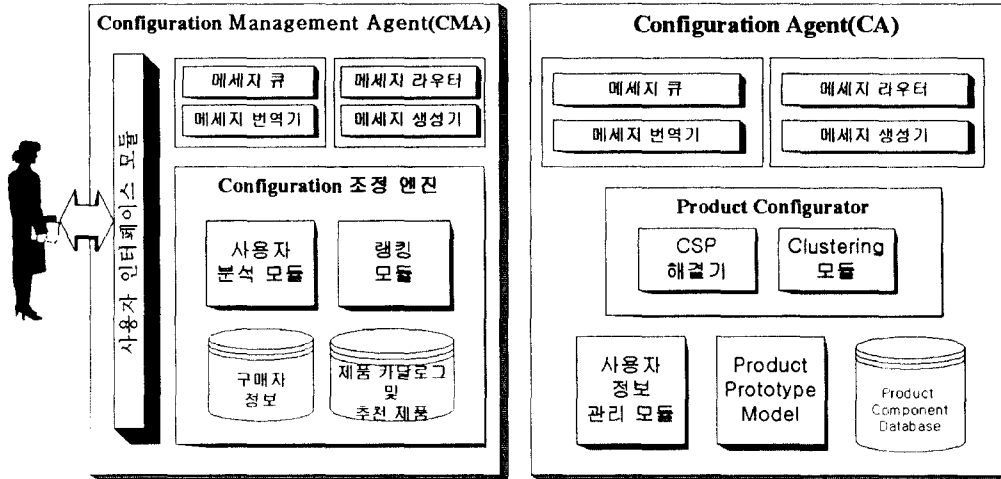
후에 벡터 모델링 단계와 제약조건 처리 모듈의 알고리즘을 살펴보겠다.

## 2. 전자상거래에서의 Product Configuration System

Product Configuration 시스템은 전자상거래 상에서 이뤄지는 구매행위와 판매 행위의 보조를 위해 제품을 자동 구성하기 위한 시스템이다. 이 시스템은 크게 구매자의 요구분석 모듈, 제품 프로토타입 모델 정의 및 선택 모듈, 선택된 모델을 기반으로 제약조건을 생성하여 해를 구하는 모듈로써 구성되어 있다. 이 장에서는 제약조건 처리모듈에서의 작업을 위한 전처리 단계로써 벡터 모델링을 위한 전체적인 시스템의 모습과 수행과정을 먼저 소개하였다.

### 2.1 전체 Configuration 시스템의 구조

Configuration 에이전트(CA)는 가상 쇼핑물을 대상으로 제품을 관리하고 있으며, 웹에서 제품을 구입하려는 고객은 Configuration 조정 에이전트를 통해서 가상 쇼핑물에서 원하는 제품을 구입할 수 있다. 구매자는 자신이 원하는 제품을 구입하기 위해서 여러 쇼핑물을 방문하지 않고, 각 쇼핑물의 서비스를 해주고 있는 Configuration 에이전트를 통해서 자신에게 제일 적합한 물품을 선택할 수 있다. 쇼핑물과 고객과의 사이에는 Configuration 조정 에이전트가 있어서 구매자의 제품에 대한 요구 명세서를 얻기 위한 분석과정을 수행한다. <그림 1>에서는 Configuration 조정 에이전트와 Configuration 에이전트 각각의 내부구조를 보여주고 있다.



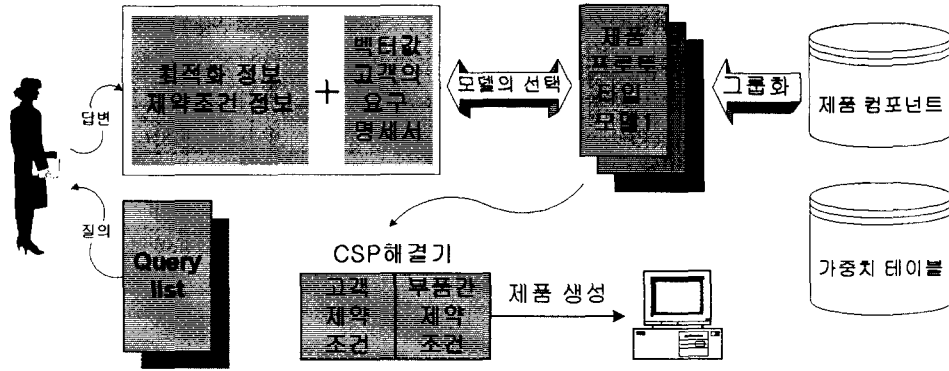
<그림 1> CMA와 CA의 구조

Configuration 조정 에이전트는 사용자 분석 모듈, 랭킹 모듈, 구매자 정보를 저장하기 위한 데이터 베이스를 가지고 있다. Configuration 조정 엔진은 사용자 분석 모듈을 통해 구매자의 정보를 수집하여 구매자 정보 데이터베이스에 저장하고, 벡터화 된 요구 명세서와 세부 요구 정보로 나누어서 메시지 큐에 저장을 한 후에, 각 Configuration 에이전트에게 이 정보를 보내주는 역할을 한다. 또한 각 Configuration 에이전트들이 구성을 해서 보내준 제품들의 목록들은 데이터 베이스에 저장한 후에 랭킹모듈에 의해서 랭킹을 해서 구매자에게 보여준다. Configuration 에이전트는 Configuration 조정 에이전트에서 받은 구매자의 벡터화 된 요구 명세서와 세부 정보를 받아 사용자 정보 관리 모듈에 저장을 한다. Configuration 에이전트는 주 엔진인 Configurator을 가지고 있다. Configurator은 제품의 프로토타입 모델을 쇼핑 물의 제품 데이터 베이스를 클러스터링 하기 위한 모듈을 가지고 있고, 요구명세서와 마찬가지로 벡터 공간상에서

표현되어 나타난다. Product Configurator은 구매자의 요구명세서 벡터 값을 이용하여 제품 프로토타입 모델 벡터 공간에서 제일 적절한 모델을 선택을 하게 된다. 선택된 제품 프로토타입 모델의 컴포넌트들을 제약조건을 이용하여 제품을 재구성하기 위한 제약조건 처리모듈을 통해서 최종 제품을 구성하는 작업을 하게 된다.

## 2.2 Configuration 처리과정

본 논문에서는 전자 상거래에서 멀티 에이전트 환경을 이용하여 Configuration 구성하는 시스템의 처리과정은 <그림 2>와 같이 요약될 수 있다. 단계4는 실제 제품을 구성하는 주 엔진인 제약조건 처리모듈을 위한 단계이고, 단계2, 3은 본 논문에서 추가적으로 정의한 도메인 컴포넌트의 탐색공간을 최소화하기 위한 전처리 단계로써 제품요구명세서와 프로토타입 모델에 대한 벡터 모델링 단계 및 이들간의 매칭 단계로써 정의될 수 있다.



<그림 2> 시스템의 전체 수행 흐름

① 구매자 요구정보 수집 단계

Configuration 조정 에이전트는 구매자의 요구가 발생하면 미리 준비된 질의를 통해 구매자의 정보 수집을 한다.

② 구매자 요구 명세서의 작성 단계

전 단계에서 수집된 정보는 각 컴포넌트의 요구수준별로 가중치가 부여되어 벡터화 된 요구명세서로 작성이 된다. 세부 정보와 함께 각 Configuration 에이전트에게 보내진다.

③ 제품 프로토타입 모델의 선정 단계

각 Configuration 에이전트에 존재하는 제품 프로토타입 모델 중에서 고객의 요구명세서와 제일 유사성이 강한 제품 프로토타입 모델을 선정한다.

④ 요구 제품의 Configuration 단계

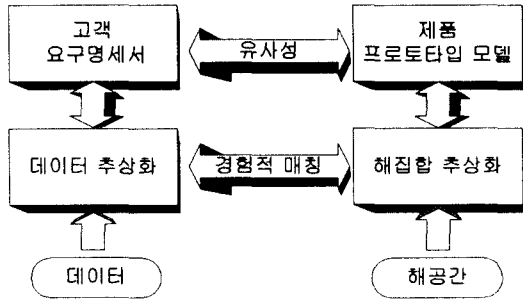
선택된 프로토타입 모델에 대한 컴포넌트들에 대해서 제약조건 처리모듈을 이용해서 최종 제품을 구성한다.

⑤ 완성 제품의 추천 단계

각 Configuration 에이전트에서 구성된 제품은 Configuration 조정 에이전트에 모아진 후 구매자에게 추천한다.

### 3. Configuration을 위한 Classification Problem Solving

본 연구에서 Configuration 문제에 대한 초점은 문제의 특성상 NP complete문제인 제품 구성 방법을 위한 탐색공간의 최소화를 유도함과 동시에 구매자가 원하는 제품을 구성할 수 있도록 하는 것이다. 이를 위한 이론적인 기반으로써 Clancey가 명시한 Classification Problem Solving 방법(Clancey, 1985)에 기반을 한 전처리 단계를 정의하여 제약조건 처리모듈에서의 수행 과정을 지원하였다. <그림 3>은 Classification 방법을 이용한 본 문제의 해결을 위한 추론 구조를 보여주고 있다. 구매자에 대한 요구 데이터는 추상화 과정 후에 요구명세서로 변환이 되고, 제품들은 추상화에 의한 제품 프로토타입 모델로 정의가 된다. 추상화 된 데이터와 해집합 간의 경험적 매칭은 구매자의 요구명세서와 제품 프로토타입 모델사이의 유사성을 이용하여 변환이 될 수 있다. 경험적 매칭은 전문가의 지식을 이용하여 지식베이스를 구축함으로써 가능하지만, 유사성은 양쪽의 값을 벡터화 함으로써 가능하고 지



<그림 3> 각 구매자에 대한 제품의 추론구조

식베이스를 이용하는 것보다 유지와 관리에 유리하다고 할 수 있다. 유사성의 측정을 위해서 벡터 값을 이용하는 것은 정보 검색(Information Retrieval)에서 사용되어지는 다루어지는 벡터 유사성을 이용하였다. 정보 검색의 분야에서 질의와 검색된 문서 사이의 유사성은 이들을 대표하는 키워드 빈번도(word frequencies)의 벡터로써 표현하고 두 빈번도 벡터에 의해 형성되는 코사인 값의 크기를 계산함으로써 특정되어진다 (Salton, 1988). 질의와 검색된 문서는 구매자의 요구 명세서와 제품의 프로토타입 모델로 대치할 수 있고, 각 키워드는 제품을 구성하는 컴포넌트으로써 구성할 수 있다. 각 요구 명세서와 프로토타입 모델의 벡터화 작업을 하기 위해서 키워드의 빈번도에 대응할 수 있는 각 컴포넌트에 대한 가중치 테이블을 만들어야 한다. 요구 명세서에 있어서의 각 컴포넌트에 대한 가중치 값은 제품을 구성하는 컴포넌트 중에서 고객이 선호하는 컴포넌트에 대한 표시이고, 제품 프로토타입 모델에 있어서 가중치 값은 해당 모델을 특성 짓게 하는 역할을 한다. <그림 3>의 추론 구조에 의해서 최종적으로 선택된 제품 프로토타입 모델들은 제약조건 처리모듈에 의해서 구매자에게 전달해 줄 제품을 자동 구성하게 된다.

## 4. 단계별 Configuration 알고리즘

### 4.1 요구명세서 벡터의 생성

- 고객 요구정보 획득

구매자 요구 정보는 실제 제품을 구성할 때 요구되는 제약을 생성시킨다. 사용자 인터페이스를 통해 구매자에게 전달되는 질의 페이지로부터 메인 제품의 각 컴포넌트들에 대한 구매자의 선호도를 획득할 수 있다. 선호도로써는 각 컴포넌트들에 대한 가격이나 성능, 모델명 등으로 분류가 되고, 구매자는 Configuration 조정 에이전트가 제공하는 카탈로그에 의존해서 요구 정보를 입력하게 된다.

- 요구명세서 벡터의 생성

모든 컴포넌트들은 서로 직교하는 단위벡터를 두고, 각 벡터의 크기를 각 컴포넌트에 대한 가중치의 값으로 나타내면 다음과 같은 벡터 스페이스로 표현될 수 있다. 컴포넌트 가중치  $w$ 는 고객의 답변을 통해서 얻어진다. 고려하는 선호도  $p$ 에 따라서 하나 이상의 다차원 벡터로써 나타내어질 수 있다. 다음은 하나의 선호도  $p$ 에 대한 요구명세서 벡터의 경우이다.

$$\vec{C}_{us} = w_{cm1,spec}^p \vec{i}_1 + w_{cm2,spec}^p \vec{i}_2 + \dots + w_{cm(n-1),spec}^p \vec{i}_{n-1} + w_{cm(n),spec}^p \vec{i}_n$$

( $w$ : 각 컴포넌트 가중치  $p$ : 선호도  $n$ : 컴포넌트의 개수)

### 4.2 제품 프로토타입 모델

다양한 구매자를 상대로 수많은 제품의 구성을 하다 보면, 제품들은 특정 형태의 구매자에 따라 각 컴포넌트별 선호도가 특정 프로퍼티로 수렴을 하게 된다. 이러한 프로토타입 모델의 특

성은 전문가로부터 획득되어질 수 있다. 분산 환경에 있는 각 Configuration 에이전트는 자신들의 제품에 대한 프로토타입 모델을 갖게 되며 구매자의 요구가 발생되면 요구명세서와 부합되는 모델만을 선택하게 된다.

• 제품 프로토타입 모델 그룹핑

제품 프로퍼티 모델은 각각 고유한 특성의 프로퍼티를 갖는 도메인 컴포넌트들로 이루어진다. 각 컴포넌트들은 가격이나 성능등과 같은 선호도와 관련한 자기 다른 가중치의 값을 가지며, 이 값은 전문가의 지식을 통해 획득된다. 다음과 같은 벡터 공간으로 나타낼 수 있다.

$$\vec{cmp} = w_{cmp1, pct}^p \vec{i}_1 + w_{cmp2, pct}^p \vec{i}_2 + \dots + w_{cmpn, pct}^p \vec{i}_n$$

(w: 각 컴포넌트 가중치 p: 선호도 n: 컴포넌트의 개수)

```

K_Component_Grouping(k, Prototype_Object, SpatialIndex)
  NearestList ← NewList(k)
  Transform_Coordinate(SpatialIndex)
  K_Nearest_Grouping(NearestList, k, Prototype_Object, SpatialIndex)
  Return NearestList

K_Nearest_Grouping(NearestList, k, Prototype_Object, Node)
  If Node is a leaf node then
    For each Object in Node do
      If Dist(Prototype_Object, Object) < NearestList.MaxDist then
        Insert(NearestList, Dist(Prototype_Object, Object), Object)
      Endif
    Enddo
  Else
    ActiveBranchList ← entries in Node
    SortBranchList(Prototype_Object, ActiveBranchList)
    For each Child node in ActiveBranchList do
      If Dist(Prototype_Object, Child) < NearestList.MaxDist then
        K_Nearest_Grouping(NearestList, k, Prototype_Object, Child)
      Else
        Exit loop
      Endif
    Enddo
  Endif
  
```

<그림 4> 제품 컴포넌트 그룹핑 알고리즘

이러한 프로토타입 모델은 각 configuration 에이전트에 의해 관련 상품과의 그룹핑이 되어진다. 알고리즘은 각각의 프로토타입 모델의 컴포넌트 좌표에 가장 근접한 k개의 컴포넌트를 그룹핑 한다. k에 대한 크기의 결정은 적용되는 도메인의 크기에 따라서 변화될 수 있다. 많은 실험을 통해서 구매자의 만족도와 시스템의 전체 수행시간을 고려해서 k값에 대한 적절한 조정이 요구된다<그림 4>.

4.3 프로토타입 모델의 결정

프로토타입 모델의 결정을 위해서 고객의 요구명세서 벡터와 프로토타입 모델의 벡터에서 각 선호도에 따른 유사도를 결정을 한 후에, 이들 유사도 값들은 선형합성법에 의해서 결합되어져서 최종적 유사도 값을 결정하게 된다. 각 configuration 에이전트에 있는 다양한 프로토타입 모델들로부터 이러한 유사도 값을 얻은 후에 제일 알맞은 프로토타입 모델을 결정한다. 다음은 프로토타입 모델을 결정하기 위한 목적함수와 관련된 식이다.

$$\text{목적함수: } Sim_{spec, pct} = \sum_{i=1}^n w^{p_i} \cdot \alpha^i$$

(p = 각 선호도 i=고려되는 선호도수)

$$\alpha = \frac{R_{ct}(\vec{p}_{ct}, \vec{c}_{cus})}{|\vec{p}_{ct}| \times |\vec{c}_{cus}|} = \frac{\sum_{i=1}^i w_{cmp1, spec} \times w_{cmp1, pct}}{\sqrt{\sum_{i=1}^i w_{cmp1, spec}^2} \sqrt{\sum_{j=1}^i w_{cmpj, pct}^2}}$$

i, j = 컴포넌트의 수

$$\vec{Cus} = w_{cmp1, spec}^p + w_{cmp2, spec}^p + \dots + w_{cmp(n-1), spec}^p + w_{cmp(n), spec}^p$$

$$\overline{Pct} = w_{cmp1, pct}^p + w_{cmp2, pct}^p + \dots + w_{cmp(n-1), pct}^p + w_{cmp(n), pct}^p$$

제품 프로토타입의 모델 중에서  $Sim_{spec, pct}$ 의 크기가 가장 큰 값을 가지는 프로토타입이 구매자로부터 분석되어진 요구명세서 정보와 가장 유사한 모델로써 선택되어진다.

#### 4.4 제약조건 처리 모듈을 이용한 제품의 생성

각 Configuration에이전트는 선택된 프로토타입 모델을 제약조건 처리모듈에 보내어 최종 제품을 생성시킨다. 제약 조건 문제에서는 문제의 표현이 변수와 도메인과 이들 사이의 제약으로 표현이 된 집합  $(Z, D, C)$ 이다. 변수들은 각 컴포넌트들에 대한 리스트로써 생성이 되고, 도메인은 각 변수들에 대한 하위 도메인들로써 구성이 된다. 제품 Configuration에서는 반드시 모든 변수들이 제품 구성을 완료하기 위해서 고려되어야 할 필요는 없다. 한 컴포넌트의 변수가 다른 컴포넌트 변수에 선택 품목 이라면 전자의 컴포넌트는 고려하지 않아도 된다. 또한 제약조건은 약 제약(Soft constraints)인 구매자의 제품 요구에 대한 제약 조건과 강 제약(Hard constraints)인 제품간의 제약 조건으로 분류가 된다. 또한 제약 만족 문제에서 고려해야 할 점은 제약조건 처리모듈에 의해서 만족되는 제품이 발견되지 않았을 경우에 대한 방법이다. 이런 경우에는 약 제약인 구매자의 요구 제약을 완화시켜서 다시 수행하는 방법을 택한다. 해결기의 알고리즘 수행 시 요구되는 중요한 것은 탐색 순서(Search Ordering)이다. 본 시스템에 적용한 탐색 순서 알고리즘은 최소폭 순서(Edward, 1993)를 이용한다. 이 방법은 더 적은 제약 조건을 갖는 변수들을 더 늦게 탐색하는 방법이다. 제약조건이 적은 변수들을 더

늦게 탐색함으로써 백트래킹의 확률적 가능성을 떨어뜨림으로써 수행속도를 증가 시킬 수 있다. 본 논문에서 제안하는 제약조건 처리모듈의 수행 알고리즘은 <그림 5>와 같다.

```

PROCEDURE Configuring Component Algorithm
BEGIN
evaluation: // evaluation is current evaluation value
feasible list: // feasible list is current solution list
start in state 0: // termination state
DO
select a component which has the minimal width of a component in domain:
DO // assign a component which has next minimal width
select a component which has the next minimal width in domain:
assign a component to the feasible list:
remove the component in domain:
propagate constraints by AC-4:
IF the other domain is empty by AC-4 THEN
undo the constraint propagation:
ELSE
update feasible list:
update evaluation:
IF the assigned component is last THEN
IF feasible list is empty THEN
loose the soft constraints:
backtrack to first selection state:
END IF
END IF
END IF
WHILE (assigning is not fail OR domain of the component not empty)
WHILE (do not backtrack to state 0)
END
END-Procedure
    
```

<그림 5> 제약조건 처리모듈 문제 해결 알고리즘

#### 5. 실험 및 평가

본 논문의 시스템의 수행능력을 평가하기 위해서 제품의 도메인은 전자상거래에서 흔히 이뤄질 수 있는 개인 컴퓨터의 조립을 대상으로 하였다. 제약조건 처리모듈을 위한 전처리 단계로써 제시하고 있는 벡터 모델링 중에서 각 프로토타입 모델에 클러스터링 되는 컴포넌트의 수의 적정 수준을 유지시키기 위해서 다음과 같은 두 가지 부분에 대한 평가를 수행하였다. 첫째로 본 시스템은 제품의 추천 시스템이기 때문에 최종적으로 제품이 추천되었을 때, 각 클러스터링의 크기의 변화에 대한 구매자의 만족도를 구해 보았다. 둘째로는 시스템이 각 클러스터링 크기의 변

화에 대한 수행시간의 측정을 해 보았다. 본 실험은 Windows98의 환경 하에서 수행되었다. 언어는 Visual C++을 사용하였으며, ILOG사의 ILOG Solver 4.31을 제약조건 만족해결 엔진으로 사용하였다. 시스템의 추천제품에 대한 구매자의 만족도를 구하기 위해서 F-Measure 측정 방법을 이용하였다(Ricardo, 1999). F-Measure는 정확도(precision)와 재현율(recall)의 조합으로써 나타내어진다. 본 논문에서의 정확도는 추천된 컴포넌트들 중에서 올바르게 추천된 컴포넌트의 수를 나타내고, 재현율은 전체 추천대상 컴포넌트들 중에서 올바르게 추천된 컴포넌트의 수를 나타낸다. F-measure의 범위는 0과 1 사이의 값을 가진다.

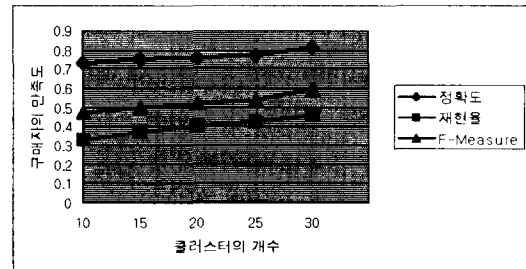
$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

실험을 위한 데이터로 사용하기 위한 테스트 환경은 다음과 같다. 우선 20개의 제품 프로토타입 모델을 만들었고, 240개의 제품 컴포넌트를 만들었다. 제품 프로토타입 모델의 선호도 벡터 값은 가격과 성능만을 고려하였다. 테스트 환경을 통해 <표 1>과 같은 실험 결과값을 얻을 수 있었다. <표 1>에서 클러스터의 수는 각 제품 프로토타입의 클러스터링 된 컴포넌트의 크기를 의미한다. 제품의 만족도에 있어서는 클러스터링의 크기가 증가하면서 고객의 요구에 좀 더 부합된 결과를 가져왔지만, 시스템의 수행속도는 이에 반비례하는 모습을 보임을 알 수 있다. 이 클러스터링의 크기가 실제 제품 컴포넌트의 도메인 크기와 같아진다면 전처리 단계가 없는 제약조건 처리모듈만의 성능과 같아질 것이다. 그러한 경우에 있어서 만족도는 더 높아지겠지만, 수행 시간에 있어서는 급격한 증가를 보일 것이다. 때문

에 이 두 가지 부분을 고려한다면 클러스터의 크기는 대략적으로 15~20정도의 크기가 적당할 것으로 보인다. <그림 6>은 이 모습을 그래프의 형태로 표현해 주고 있다.

<표 1> 시스템의 구매자 만족도와 수행속도 측정 결과

클러스터수	정확도	재현율	F-measure	수행속도
10	0.73	0.49	0.59	7 sec
15	0.75	0.56	0.64	8 sec
20	0.76	0.59	0.66	11sec
25	0.77	0.62	0.69	12sec
30	0.81	0.68	0.74	15sec



<그림 6> 고객 만족도 결과 그래프

## 6. 결 론

본 논문은 Configuration 시스템에 있어서 제품을 구성하는 일이 문제의 복잡도가 높고, 구매자의 특성을 반영하기가 힘들기 때문에 구매자의 만족도를 적정 수준으로 유지하는 가운데 시스템의 수행 속도 향상에 초점을 두었다. 이러한 문제 해결을 위해 제품 프로토타입 모델을 정의하고, 구매자의 요구와 프로토타입 모델에 대한



벡터 모델링 작업을 통하여 탐색 공간의 감소를 유도하였다. 또한 제품 컴포넌트들을 프로토타입에 클러스터링 할 때 이것의 크기를 조정하는 것이 제약조건 처리모듈과 전체 시스템의 성능을 위해서 중요함을 보였다. 대부분의 제품을 자동 구성하는 시스템들에서는 구매자의 요구 정보를 수집하기 위한 부분과 실제 제품을 구성해주는 부분의 성능 개선에 많은 관심을 갖고 있으며, 본 시스템 역시 이러한 부분에 대한 연구가 더 필요하다. 구매자의 요구 정보에 대한 효율적인 획득을 위해서는 질의의 정적인 제공보다는 전문가의 지식을 이용하여 동적으로 다양한 질의를 생성시키는 방법이 좋은 결과를 가져올 것으로 보인다. 또한 제약만을 이용한 방법보다는 ELIZA와 같은 채터봇 에이전트를 이용하여 전문가의 지식을 구매자 정보 획득하는데 적용한다면 더 나은 성과를 가져올 것으로 본다. 또한 협동적 여과 시스템을 적용시킬 수 있다면 다른 구매자들의 구매 정보를 이용하는 것도 좋은 추천 서비스를 위한 방법이 될 것으로 보인다.

## 참고문헌

- [1] Bernd Neumann, Configuration Expert Systems: A Case Study and Tutorial, *SGAICO Conference on Artificial Intelligence in Manufacturing, Assembly, and Robotics*, (1988).
- [2] Clancey, W.J., Heuristic Classification", *Artificial Intelligence*, 27, (1985), 289-350.
- [3] Edward Tsang, "Foundations of Constraint Satisfaction", Academic Press, (1993), 157-188.
- [4] Fleischanderl, G., G. E. Friedrich and A. Haselböck and H. Schreiner and M. Stumptner, "Configuring large systems using generative constraint satisfaction", *IEEE Intelligent Systems & their applications* 13(4), (1998), 59-68.
- [5] Helen Lowe, "Proof Planning: A Methodology for Developing AI Systems Incorporating Design Issues", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, (1994).
- [6] Jörg Rahmer, Voß Angi, "Case-Based Reasoning in the Configuration of Telecooperation Systems", *AAAI 1996 Fall Symposium Workshop*, (1996).
- [7] McDermott, J., "R1: A Rule-Based Configurer of Computer Systems", *Artificial Intelligence* 19, North-Holland, (1982), 39-88.
- [8] Michal Pechoucek, "Proof Planning and Industrial Configuration", *M.Sc Thesis, Dept. of Artificial Intelligence*, University of Edinburgh, (1996).
- [9] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, "Modern Information Retrieval", Addison-Wesley Publish, (1999), 27-30.
- [10] Salton, G., C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval", *Journal of the American Society for Information Science*, (1988), 513-523.
- [11] Strecker A., R. Cunis and A. Günter, "PLAKON-Übersicht über das System", *Informatik-Fachberichte, Springer*, (1991).
- [12] "Build Your Own System", URL: <http://www.dell.com/store>
- [13] "PersonaLogic", URL: <http://www.personalogic.com>
- [14] "Trilogy's Selling Chain", URL: <http://www.trilogy.com/prodserv/>

Abstract

## Configuration System through Vector Space Modeling In E-Commerce

Se-Hyoung Kim\*  
Geun-Sik Jo\*

There have been lots of researches for providing a personalized service to a customer using one-to-one marketing and collaborative filtering techniques in E-Commerce. However, there are technical difficulties for providing the recommendation of products for users, which often involve high complexity of computation. In this paper, we have presented an integrated method of classification problem solving method and constraint based configuration techniques. This method can reduce a complexity of computation by classifying a solution domain space that has a higher complexity of composition. Thereafter, we have modeled customers constraints and the components of products to configure a complete system by passing it to constraint processing module in Constraint Satisfaction Problems. Constraint-based configuration uses the constraint propagation using the constraints of buyers and the constraints among PC components to configure a proper product for a customer. We have transformed and applied vector space modeling method in the field of information retrieval to consider a customer satisfaction in addition to the CSP. Finally, we have applied our system to test data for evaluating a customers satisfaction and performance of the proposed system.

**Key words:** Configuration, Classification, Prototype Model, CSP, Electronic Commerce

---

\* College of Electronic Engineering, Inha University