

Rule-based 모델링 방법을 이용한 Product Configuration 에서 CSP 해결 프로세스

임정혁*, 정현준**, 백두권**[†]

*고려대학교 컴퓨터정보통신공학과

**고려대학교 컴퓨터전파통신공학과

e-mail : solme7000@empal.com, junghj85@gmail.com, baikdk@korea.ac.kr

Rule-based modeling method in Product Configuration for CSP solving process

ChoungHyuck Lim*, HyunJun Jung**, Doo-Kwon Baik**[†]

* Dept. of Computer and Information Communication Engineering, Korea University

** Dept. of Computer and Radio Communications Engineering, Korea University

요 약

현대 산업 발전에 따라 대량 생산 체제에서 소비자 개별 주문 요건을 반영하여 생산하는 대량맞춤 생산(Mass Customization) 체제로 변화해가고 있다. 이런 대량맞춤 생산 체제의 제품구성(Product Configuration) 시에 나타나는 CSP(Constraint Satisfaction Problem) 해결 방법론을 제시하고 있다. 기존 방법에서는 복잡 다양한 제약조건을 수학적으로 표현하여 제약 조건 검사에서 소요시간과 검색 공간을 최소화 할 수 있는 알고리즘이 제시했다. 그런데 소비자의 주문 요건들이 빠르게 변화되고 복잡 다양 해지고 있다. 그래서 대상 모델 및 제약조건이 변경이 되었을 경우 기존 방법에서 제시하고 있는 CSP 해결에서는 수정 관리 하기가 쉽지 않고 확장성이 낮다. 본 연구에서는 룰 기반 모델링으로 CSP 문제 해결 프로세스 정립을 하여 관리의 편리성을 제공하고 확장성을 향상시킨다.

1. 서론

현대 기업에서는 규격화된 제품을 대량생산(Mass Production) 기반으로 제품 생산 및 관리를 하고 있다. 대량생산 체제에서 생산된 제품에 또 다른 경쟁 요소인 고객화(Customization)가 더해져 대량맞춤 생산(Mass Customization)이 만들어 졌다. 대량맞춤 생산 체제에서 B2B 또는 B2C 형태가 이루어지고 있으며 고객의 다양한 요구 만족 조건을 반영하는 제품 구성(Production Configuration)을 하고 운영할 때 제품 관리 및 고객 주문 형태가 발생한다[1,2]. 주어진 제약조건을 검사하여 결과의 해를 구할 때, 제약만족문제(Constraint satisfaction problem)가 발생한다[3]. 제품 구성 및 주문 시에 발생하는 제약만족문제 해결 기법을 이용하여 보다 효율적인 문제 해결 하는 접근 방법이 있다.

기존 연구에는 제약조건을 대수(Algebraic) 연산자로 연결된 부울(Boolean) 변수를 사용하여 제약만족문제를 해결하였다[5]. 그리고 주어진 문제를 논리적 기호로 표현을 하여 처리하는 지식 기반으로 제약만족문

제를 해결하였고[6], 또는 문제 해결을 도메인 집합 개념을 도입하여 검색을 통한 합당한 해를 구하는 해결기법들이 이용되고 있다[7]. 기존 연구 기법들에서 공통적으로 나타나는 복잡한 제약조건을 단순화 및 수학적 수식 표현화하여 보다 발전된 알고리즘 기법과 제약 조건을 검사하여 해답을 구할 때 실행 시간 문제를 해결하기 위한 연구가 진행 되고 있다. 그런데 요구사항 변화에 따른 제약조건 수정이 어려우며 이로 인한 확장성이 낮다는 한계를 내포하고 있다.

본 논문의 제약만족 문제를 해결하기 위한 룰 기반 모델링 접근 기법을 제안하여 제약만족 문제 해결 모델 관리의 편리성, 확장성을 향상시키고자 한다. 적용 사례 연구로는 금융 상품을 대상으로 제시한 프로세스를 적용한다.

본 논문의 구성은 2 장에서 Integer Programming, Knowledge-based System, CSP 해결 Approach 에 대한 설명 및 문제점을 분석하고, 3 장에서 룰 기반 모델링 접근 기법을 제안하고, 4 장에서는 금융상품 실제 적용한 사례를 보이고, 5 장에서는 룰 기반 모델링 측면에서 기존 접근 방법과 비교 평가를 기술한다.

2. 관련 연구

2.1 Integer Programming Approach

제약 조건을 기호적 제약 조건과 수치적 제약 조건

* 이 연구에 참여한 연구자는 '2 단계 BK21 사업'의 지원을 받았으며 이 논문은 정보통신산업진흥원의 SW 공학 요소기술 연구개발사업의 결과물임을 밝힙니다.

† 교신저자 : 백두권

으로 모델링으로 표현하기 위해서 대수(Algebraic) 연산자로 나타내며 변수의 값은 정수가 사용되어 최적의 해를 구하는 접근 방식이다[4,5]. 제약조건과 주어진 문제의 대상이 커질수록 방대한 양의 변수들을 필요로 하며, 이들 사이의 복잡한 상관 관계가 존재하여 사용자의 요구 변화에 따른 시스템에 대한 수정을 어렵게 한다는 한계가 존재한다.

2.2 Knowledge-based Approach

주어진 문제의 대상을 논리적으로 자연어와 같이 자연스럽게 표현할 수 있으며 기호적인 제약 조건의 처리가 편하다. 따라서 지식 기반으로 추론 메커니즘을 이용하여 문제를 해결한다[6]. 특히 사례 기반 추론(Case-based Reasoning)과 같은 문제해결 분야에서 널리 사용되어 왔다. 그러나 지식 습득 과정이 시스템의 개발에 커다란 병목 현상이 되고, 시스템 성능을 평가하기 어렵다. 즉 논리 언어는 수치 제약 조건의 처리에 대한 어려움 때문에 기호적인 제약 조건의 처리와 함께 수치 제약 조건의 처리도 동시에 요구되는 문제의 적용에 많은 한계를 나타내고 있다.

2.3 CSP(Constraint Satisfaction Problems)-based Approach

CSP 는 여러 가지 제약 조건을 만족하는 해를 구하는 것으로 CSP 의 구성요소는 변수, 도메인, 그리고 변수들간의 제약 조건으로 이루어진다.

- 1) 변수들의 집합: $V = \{ v_1, v_2, v_3, \dots, v_n \}$
- 2) 각 변수 V_i 가 가질 수 있는 도메인:
 $D = \{ d_1, d_2, d_3, \dots, d_n \}$
- 3) 변수들 또는 변수간의 제약조건:

$$C = \{ c_1, c_2, c_3, \dots, c_n \} \text{ (단, } n \text{은 변수의 개수)}$$

제약만족은 변수의 개수와 각 변수에 들어갈 도메인의 범위에 의해 지수적으로 증가하는 복잡도가 증가하며, 변수나 도메인이 많은 경우에는 그 해를 구하는데 많은 시간이 걸린다.

이러한 문제를 해결하기 위한 세가지 방법이 있는데

1) Classical CSP 방법은 변수들을 전향검사(Forward Checking)를 이용하여 제약 조건의 일치성 검사(Consistency Checking)을 하여 도메인 여과(Domain Filtering)을 적용하여 전체 탐색 공간의 크기 줄여 검색 시간을 단축하는 방법이다[7].

2) Dynamic CSP 방법은 도메인 내에서 탐색 범위를 줄여 실행 속도를 향상 시키기 위해 초기에 할당된

변수와 값 순서의 제약을 추가 또는 제거 변경을 통해 좀더 빠르게 해를 구하도록 방법인데, 절대적으로 조건에 해가 위배되지 않는다[8].

3) Flexible CSP 방법은 부분적으로 제약 조건 완화를 통해 모든 조건을 따르지 않아도 해를 찾는 것인데, 경우에 따라 최선의 선택을 할 수가 있다[9].

위의 해결 방법들 역시 제한 조건이 복잡하거나 많을 경우나 도메인을 좁히지 못하였을 경우에 backtracking[10]이 자주 발생하는 문제를 안고 있으며 조건이 자주 변경되는 경우 제약조건 위반(충돌)에 대한 정보를 사용자에게 제시하여 변경관리가 쉽지 않고, 이로 인한 확장성이 낮아지는 문제점이 발생한다.

2.4 Rule-Engine Framework

2.4.1 DB(Data Base)기반의 룰 엔진

DB 기반의 룰 엔진은 비즈니스 규칙을 정의하는 매개변수를 데이터베이스에 저장하고, 매개변수를 이용한 실제 규칙 처리, 즉 비즈니스 로직 처리는 프로그램이 처리하는 방식과 독립적인 룰 엔진을 사용하는 방식이 있다. 독립적인 룰 엔진을 사용하는 방식은 룰 엔진이 비즈니스 로직을 처리하도록 설계되어 룰 데이터베이스에 매개변수 값은 물론, 이들을 이용한 상세 로직을 저장하고 처리한다[11,12].

2.4.2 Inference 기반의 룰 엔진

Inference 기반의 룰 엔진은 스스로 판단하여 실행할 비즈니스 룰을 선택하여 실행시킨다. 비즈니스 룰을 표현하는 스크립트 언어를 지원하고 있으며 입력된 비즈니스 룰의 로직이 부정확, 불완전하다면 전혀 예상치 못한 결과를 초래한다 [12].

본 논문에서 적용한 룰 엔진은 제품구성 모델 과 CSP 모델이 구조적 객체기반으로 설계하여 데이터베이스 모델링 되기 때문에 상호 호환성 및 성능을 고려할 때 DB 기반의 룰 엔진으로 적용을 하였다

3. 룰 기반 모델링 Approach 제안 모델

본 연구에서는 다양하고 조건변화가 많은 대량맞춤 생산 기업 환경에서 제약조건과 제품구성 도메인의 구조적 재고를 통하여 CSP 해결 프로세스를 정립하여 관리의 편리성을 제공하여 확장성을 높이는 방향으로 제안 모델을 제시하기로 한다. 그러기 위해서는 룰 기반 제약조건 모델링을 통한 CSP 해결을 데이터베이스 스키마로 정의하고 데이터로 관리하여 표준화된

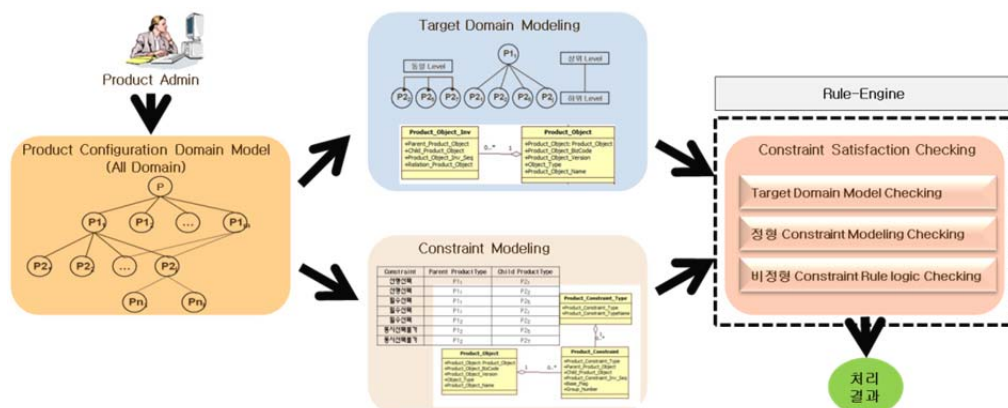


그림 1. Rule-based 모델링 프로세스

SQL 로 접근성을 높인다. 또한 룰 기반에서는 제약조건 검사 로직이 변경 되었을 경우 로직 변경관리가 용이하다[11]. 그림 1 에서는 룰 기반 모델링으로 CSP 해결 프로세스를 도식화 하였다.

3.1 제약조건 모델

제품 구성에서 제약조건은 상위 객체와 하위 객체 간 제약조건과 동일 객체간의 제약조건으로 관계 규칙을 정의한다. 이 제약조건 모델에서는 조건 규칙 유형을 정의 한다.

- 선행선택 1: 첫 번째 선택될 객체를 기준으로 하위 객체가 선행선택이 되는 조건으로 순서에 영향을 받는다.
- 선행선택 2, 선행선택 3: 위 유형과 비슷하게 두 번째, 세 번째 선택된 객체를 기준으로 하위 객체가 선행 선택이 되는 조건으로 순서에 영향을 받는다.
- 필수선택: 선택된 객체 상호간에서 필수적으로 선택되어야 하는 한다.
- 동시선택불가: 선택된 객체 상호간에 선택을 할 수 없다.
- 동시선택: 선택된 객체는 반드시 동시에 선택되어야 하는 객체가 있어야 한다.
- 객체종속관계: 선택된 객체에 종속관계가 필요한 경우의 관계 정의한다.
- 단독선택 : 단독 선택 객체이어야 한다.

그림 2 에서 보면 제약조건 유형 정의를 *Product_Constraint_Type* 에 필수선택인지, 순서를 정의 해야 하는지, 동시적으로 검사하는 유형인지를 관리자가 정의해서 관리 등록한다.

3.2 대상 Product Configuration 모델

제품구성(Product Configuration) 정의된 구조의 도메인에서 실제 검사를 해야 하는 대상 도메인을 정형화된 객체 데이터 모델로 정의 한다.

그림 2 에서 *Product_Object_Inv* 에 정의된 전체 도메인정의를 *Product_Constraint* 으로 대상 도메인을 축소하여 관리를 하게 한다. 이 모델의 구조는 상위객체와 하위객체로 구성을 하되 동일 레벨 구성에서는 Dummy 객체를 두어 구성 할 수 있다.

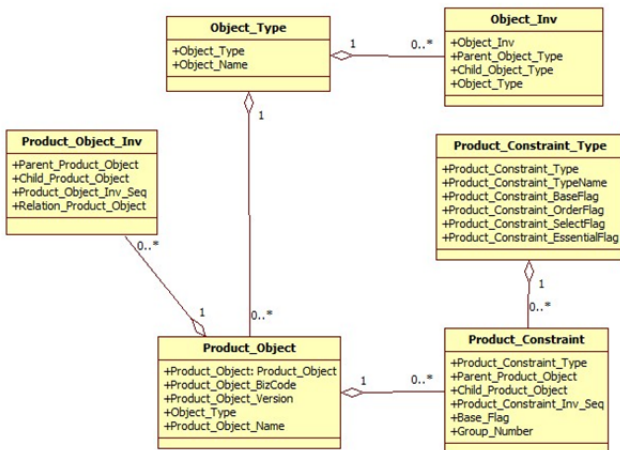


그림 2. Product Configuration & Constraint model Class diagram

3.3 CSP 해결 Rule-engine process

룰 엔진에서는 실제 CSP 를 검사하는 로직 부분을 포함한다. 그래서 첫 번째로 대상 도메인 모델을 하고, 두 번째로 정형 제약조건 모델링 검사, 세 번째로 비정형 제약조건 규칙 로직 검사하는 순서를 가진다. 제약조건의 정형정보는 *Product_Constraint* 에서 관리가 되며, 제약 조건이 비정형적 정보로 자주 변경이 이루어지는 조건들은 비즈니스 규칙으로 로직 처리 해야 한다.

$$C_i = \begin{cases} 1, & \text{if } c_i \text{ is true} \\ 0, & \text{otherwise} \end{cases}, \text{ 단 } i=1,2,3,\dots,n$$

그래서 제약조건의 checking 값은 1 또는 0 가 되어 로직 구현을 1 일 때 pass 가 되어 다음 단계 제약조건을 검사하게 되고, 0 일 때 fail 이 되어 메시지를 띄워 사용자에게 알린다.

결국, 제약조건들은 정의하여 분석 모델링 하여 데이터로 관리를 하게 되므로 인해 정형적 형태가 되므로 추후 변경 관리적 측면에서 접근성, 확장성이 증가하게 되며, 비정형적 조건들은 룰 엔진에서 변화의 대상으로 간주하여 비즈니스 로직 형식으로 처리가 되므로 제품 관리자는 변경이 자주 발생하는 제품 환경에서 제약조건 관리가 용이해져 확장성이 높아진다.

4. 적용 사례

제품(Product) 은 유형적 제품과 무형적 제품으로 나누어진다[13]. 본 연구에서 제시한 룰 기반 모델링 프로세스를 검증하기 위하여 요건 변화가 많은 무형적 제품인 금융상품을 대상으로 프로세스를 적용한다. 적용 방법은 3 장에서 제시한 프로세스를 따라 자동차 보험 상품에 대해 모델링을 수행한다.

4.1 제약조건 모델 적용

3.1 에서 언급한 제약조건 유형을 금융상품에 맞게 수정 정의하여 적용한 조건 유형이 그림 3 이다.

코드	이름
SNG_ENV	단독가입
NOSYNC_BOTH_NOSTD	동시가입불가
SYNC_BOTH_NOSTD	동시가입
SLT_PRE_ONE_STD	선행선행가입
PRE_ONE_STD	선행가입

그림 3. 제약조건 유형정의

4.2 대상 Product Configuration 모델 적용

4.1 에서 정의된 제약조건 유형에 해당되는 대상 도메인을 정의 적용한 것으로 3.2 에선 언급한 모델을 적용한다.

관계유형	SET_ID	관계구분	객체
선행가입(+)	802	기준객체	[판매담보][200004]대물배상책임
		관계객체	[판매담보][200001]대인배상책임
		관계객체	[판매담보][200003]대물배상책임(의무)
선행가입(+)	803	기준객체	[판매담보][200002]대인배상책임
		관계객체	[판매담보][200001]대인배상책임
		관계객체	[판매담보][200003]대물배상책임(의무)
선행가입(+)	804	기준객체	[판매담보][200005]자기신체사고
		관계객체	[판매담보][200001]대인배상책임
		관계객체	[판매담보][200003]대물배상책임(의무)
		관계객체	[판매담보][200004]대물배상책임

그림 4. 대상도메인 적용 정보

4.3 CSP 해결 Rule-engine 적용

실제 해당 제품에서 제시된 제약조건을 바탕으로 검사를 진행하는 로직을 구현한 비즈니스 룰 부분이

다. 여기서는 이노콜스의 Rule Builder Client 으로 구현한 화면이 그림 5이다.

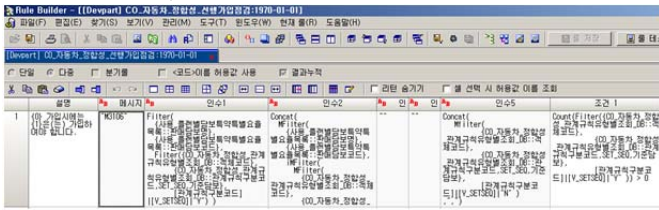


그림 5. 제약조건 검사 비즈니스 룰

위 적용 사례를 통하여 본 논문에서 제시한 프로세스를 통하여 제약조건 변경이 잦은 기업환경에서 변경 처리와 확장성을 검증하였다.

5. 비교 평가

본 연구에서 제안한 룰 기반 모델링을 이용한 CSP 해결 프로세스 접근 방식을 통하여 기존 CSP 해결 접근 방식과 정성적 비교 결과 표 1 과 같다. 정형적 제약조건을 모델링 하여 데이터베이스로 관리하여 확장성 및 변경처리를 우수하게 만드는 효과가 있었다. 특히 DB 기반의 룰 엔진과 높은 호환성으로 인해 부수적인 처리 효과를 보았다. 그러나 알고리즘 복잡도에서는 기존 방식과 별다른 차이가 없었고, 처리 속도는 기존 방식의 속도 측정이 불가하므로 적용사례를 기준으로 룰 기반 모델링 접근만 평가를 하였다.

이로써 제약조건을의 모델 정립을 통하여 변화가 많은 기업 환경에서 관리의 변경처리 편리성을 제공하고 확장성을 향상시킬 수 있는 장점이 있었다.

표 1. Rule-based modeling Approach 의 정성적 품질 비교

구분	Integer Programming Approach	Knowledge-based Approach	CSP-based Approach	Rule-based modeling Approach
처리속도	3	3	4	4
알고리즘 복잡도	3	4	5	4
변경처리력	1	4	2	5
확장성	1	3	3	5
재사용성	1	2	2	2

[범례] 5: 아주 높음, 4: 높음, 3: 보통, 2: 낮음, 1: 아주 낮음.

6. 결론 및 향후 연구

고객의 요구 조건의 변화가 많은 대량맞춤 생산 (Mass Customization) 환경에서 기존 CSP 해결 방법론에서 복잡하고 다양한 제약조건을 시스템적으로 표현하여 적용 후에 대상 모델 및 문제조건 등이 변화가 되었을 때, 확장성이 낮고 복잡한 수식으로 되어 있어 수정 관리가 쉽지 않다. 본 논문에서 룰 기반 모델링을 통한 프로세스 정립하여 변화가 많은 금융상품에 적용하여 이 접근법의 실효성을 정성적으로 비교 평가를 하였다. 결과로 변경처리가 용이하며 확장성에서 높게 나타났다. 특히, 소비자의 변경 요건이 많고 제약조건이 많은 대량맞춤 생산 환경에서 더욱 유용하다.

향후 연구로서, 비정형적인 로직을 처리하는 비즈니스 룰에서 복잡한 제약조건 처리 알고리즘을 좀더

단순화하여 수학적 수식화를 하는데 많은 연구를 할 필요가 있으며, 이를 통해 재사용성을 높이는데 좀더 많은 노력을 해야 할 것이다.

참 고 문 헌

[1] D. Sabin, R. Weigel, "Product Configuration Frameworks -A Survey," IEEE Intelligent Systems, Vol.13, No.4, pp.42-49, 1998

[2] M. Aldanondo, K. Hadj-Hamou, G. Moynard, J. Lamothe, "Mass customization and configuration: requirement analysis and constraint based modeling propositions," Integrated Computer. Aided Eng., Vol.10, No.2, pp.177-189, 2003

[3] V. Kumar, "Algorithms for constraint satisfaction problems: a survey," AI Magazine, Vol.13, No.1, 1992

[4] P.M. van den Broek, "Optimization of Product Instantiation using Integer Programming," Proceedings of the 14th International Software Product Line Conference, Vol.2, pp.107-111, 2010

[5] I. Rivin, R. Zabih, "An Algebraic Approach to Constraint Satisfaction Problems," Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pp.284-289, 1989

[6] M.H. Sqalli, L. Purvis, E.C. Freuder, "Survey of Applications Integrating Constraint Satisfaction and Case-Based Reasoning," Procs. of the 1st International Conference and Exhibition on PACLP, 1999

[7] Linda L. Zhang, Qianli Xu, Yugang Yu, Roger J. Jiao, "Domain-based production configuration with constraint satisfaction," International Journal of Production Research, 2012

[8] J. Amilhastre, H. Fargier, P. Marquis, "Consistency restoration and explanations in dynamic CSP: Application to configuration," Artificial Intelligence, Vol.135, No.1, pp.199-234, 2002

[9] F. H'Mida, F. Vernadat, "A constraint approach (flexible CSP) for alternative cost estimation of a mechanical product," International Journal of Production Research, Vol.47, No.2, pp.305-320, 2009

[10] H. Xie, P. Henderson, M. Kernahan, "Modelling and solving engineering product configuration problems by constraint satisfaction," International Journal of Production Research, Vol.43, No.20, pp.4455-4469, 2005

[11] 박준호, "증권 상품 업무 개발을 위한 BRMS기반의 상품시스템 설계 및 적용에 관한 연구," 국민대학교 대학원, 2011

[12] 김성민, "BRMS(Business Rule Management System) 적용에 의한 업무프로세스 개선효과 검증," 국민대학교 비즈니스IT전문대학원, 2009

[13] P. Kotler, S.J. Levy. "Broadening the concept of marketing," Journal of Marketing, Vol.33, No.1, pp.10-15, 1969