The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 14, No. 2, pp.115-120, Apr. 30, 2014. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2014.14.2.115

JIIBC 2014-2-16

상호작용을 고려한 최적의 제품휘처형상 도출 방법

A Method for Deriving an Optimal Product Feature Configuration Considering Feature Interaction

이관우*

Kwanwoo Lee*

요 약 많은 소프트웨어 프로덕트 라인 공학 방법들은 휘처모델을 사용하여 제품들 간의 공통성과 가변성을 휘처 단위로 구조화시키고, 특정 제품 개발을 위해 필요한 휘처 집합인 제품휘처형상을 도출한다. 제품 생산 시에 선택될 휘처는 주로 제품의 요구되는 품질 속성에 의해서 결정된다. 지금까지 발표된 대부분의 방법들은 휘처와 품질속성 간의 선형적 상관관계를 통해 최적의 품질 속성을 만족시킬 수 있는 제품휘처형상을 도출하였다. 하지만, 휘처 간의 상호작용을 고려한다면 휘처와 품질 속성 간의 관계는 비선형식으로 정의될 수 있다. 본 논문에서는 휘처 간의 상호작용을 고려하여 요구되는 품질 속성을 최적으로 만족시킬 수 있는 제품휘처형상 도출 방법을 제안한다. 제안된 방법을 평가하기 위해 네 가지 프로덕트 라인 사례에 대해 실험한다.

Abstract Many product line engineering methods use the feature model to structure commonality and variability among products in terms of features and to derive a product feature configuration, which is the set of features required for the development of a product. Features to be selected during product derivation are mainly determined based on the quality attributes required for a product. Most methods published so far derived an optimal product feature configuration through linear co-relationship between features and quality attributes. However, the co-relationship between features and quality attributes can be formulated as a non-linear function because of feature interactions. This paper proposes a method that derives an optimal product feature configuration considering feature interactions. Four product line cases are used to validate the proposed methods.

Key Words : software product line, optimization, product derivation, feature interaction, feature configuration

Ⅰ. 서 론

소프트웨어 프로덕트 라인 공학은 유사한 휘처(Feature)를 지닌 소프트웨어 제품들을 개별적으로 개발하지 않고, 제품 개발에 공통으로 재사용가능한 핵심자산을 개발 한 후에 이를 이용하여 개별 제품을 생산해 내는 소프트웨어 재사용 패러다임이다[1]. 지금까지 많은 소프트웨어 프로덕트 라인 공학 방법들[2,3,4]은 휘처모델(Feature Model)[5]을 기반으로 제품들 간의 공통적 혹은 가변적 특징을 휘처(Feature) 관점에서 구조화시키고, 이를 바탕으로 특정 제품 생산을 위한 제품휘처형상(Product Feature Configuration)을 도출하는 접근방법을

*정회원, 한성대학교 정보시스템공학과
접수일: 2014년 3월 11일, 수정완료일: 2014년 4월 2일
게재확정일: 2014년 4월 11일

Received: 11 March, 2014/ Revised: 2 April, 2014
Accepted: 11 April, 2014
*Corresponding Author: kwlee@hansung.ac.kr
Dept. of Information Systems Engineering, Hansung University, Korea

택하였다. 여기서 제품휘처형상은 휘처모델 내의 휘처 중에서 특정 제품을 위해 선택된 휘처 집합으로서 핵심 자산 내의 아키텍처 및 컴포넌트를 선택하거나 적응시키는 파라미터 역할을 한다.

제품 생산 시에 선택될 휘처는 주로 제품의 품질 속성에 의해서 결정된다. 지금까지 발표된 휘처 기반의 제품 형상 도출 방법들[2,3,4]은 휘처와 품질 속성 간의 선형적 상관관계를 통해 최적의 제품휘처형상을 도출하였다. 하지만, 이러한 방법들은 휘처 간의 상호작용을 고려하지 않으므로, 도출된 제품휘처형상이 최적의 해가 아닐 수 있다.

본 논문에서는 품질 요구사항이 제품 특성의 선택에 영향을 미치는 대응 관계를 휘처 상호작용을 고려하여 정의하고 이를 바탕으로 요구되는 품질 요구사항을 최적으로 만족시킬 수 있는 제품 특성을 선택 하는 방법을 제안한다. 제안된 모델과 방법을 평가하기 위해 네 개의 프로덕트 라인 사례를 바탕으로 최적화된 제품 구성을 실험하고 평가한다.

Ⅱ. 기본 개념

본 논문의 핵심은 휘처모델로부터 휘처 상호작용을 고려하여 품질속성을 최적화시키는 제품휘처형상을 도출하는 기법을 제안하는 것이므로, 제안된 기법의 기반 개념인 휘처모델, 휘처와 품질속성간의 관계, 휘처 상호작용에 대해서 간략히 설명한다.

1. 휘처모델

휘처모델은 소프트웨어 프로덕트 라인 내의 제품들 간의 공통성과 가변성을 휘처 단위로 표현하고, 유효한 제품휘처형상을 도출해 낼 수 있는 휘처 간의 의존관계를 정의한다.

정의 1. 휘처모델은 $FM=(F,\ \ _q, C_{EX}, C_M, C_{Op},\ C_{Alt}, C_{OR})$으로 정의되며, F는 소프트웨어 프로덕트 라인 내의 휘처 집합을 의미하고, $C_{Req}, C_{EX}, C_M, C_{Op},\ C_{Alt}, C_{OR}$은 휘처 간의 의존관계를 의미한다.

- 요구 의존관계 C_{Req}는 두 휘처 $f1$과 $f2$사이의 이진관계로서 $f1$이 제품휘처형상에 포함되면 $f2$도 반드시 제품휘처형상에 포함되어야 함을 나타낸다.
- 배타 의존관계 C_{EX}는 두 휘처 $f1$과 $f2$사이의 이진관계로서 $f1$과 $f2$는 동일한 제품휘처형상에 포함될 수 없음을 나타낸다.
- 필수 의존관계 C_M은 두 휘처 $f1$과 $f2$사이의 이진관계로서 $f1$이 제품휘처형상에 포함되면 $f2$는 반드시 제품휘처형상에 포함되어야 하고, $f1$이 포함되지 않으면 $f2$는 포함될 수 없음을 나타낸다.
- 선택 의존관계 C_{Op}는 두 휘처 $f1$과 $f2$사이의 이진관계로서 $f1$이 제품휘처형상에 포함되면, $f2$는 선택적으로 제품휘처형상에 포함됨을 나타낸다.
- 택일 의존관계 C_{Alt}는 휘처 $f1$과 휘처 집합 $FG\ \ F$사이의 이진관계로서 $f1$이 제품휘처형상에 포함되면 FG내의 휘처 중에 오직 하나의 휘처만 동일한 제품휘처형상에 포함되어야 함을 나타낸다.
- 다중선택 의존관계 C_{OR}은 휘처 $f1$과 휘처 집합 $FG \subseteq F$사이의 이진관계로서 $f1$이 제품휘처형상에 포함되면 FG내의 휘처 중에 하나 이상의 휘처가 동일한 제품휘처형상에 포함되어야 함을 나타낸다.

2. 휘처와 품질속성간의 관계

휘처모델 내의 휘처 선택은 임의로 이루어지기 보다는 제품이 요구하는 품질속성 값에 의해 중요한 영향을 받는다. 가령 자동차 프로덕트 라인에서 전기 엔진과 가솔린 엔진은 택일 의존관계를 가지는 휘처이다. 이들 휘처 각각이 연료효율성에 미치는 품질속성 값이 알려져 있다면, 품질속성 값이 높은 값을 가지는 휘처가 선택될 것이다. 따라서 휘처모델로부터 제품휘처형상을 도출하기 위해서는 휘처모델의 각 휘처에 대한 품질속성의 영향도를 측정할 필요가 있다.

1. 휘처의 품질속성 값 계산

Table 1. Calculating quality attribute values of features

요구 의존관계 $(f1,f2)\in C_{Req}$	$QF_k(f1)=Q_k(P\{f1,f2\})-Q_k(P\{parent(f1)\})-Q_k(P\{f2\})$ $QF_k(f2)=Q_k(P\{f1,f2\})-Q_k(P\{parent(f2)\})-Q_k(P\{f1\})$
필수 의존관계 $(f1,f2)\in C_M$	$QF_k(f1)=Q_k(P\{f1,f2\})-Q_k(P\{parent(f1)\})$ $QF_k(f2)=0$
선택 의존관계 $(f1,f2)\in C_{Op}$	$QF_k(f1)=Q_k(P\{f1,f2\})-Q_k(P\{parent(f1)\})$ $QF_k(f2)=Q_k(P\{f1,f2\})-Q_k(P\{f1\})$

택일 의존관계 $(f1,\{f2,f3\})\in C_{Alt}$	$Q_k(P\{f1\})=\mathrm{Min}(Q_k(P\{f1,f2\}),\ Q_k(P\{f1,f3\}))$ $QF_k(f1)=Q_k(P\{f1\})-Q_k(P\{parent(f1)\})$ $QF_k(f2)=Q_k(P\{f1,f2\})-Q_k(P\{f1\})$ $QF_k(f3)=Q_k(P\{f1,f3\})-Q_k(P\{f1\})$
다중선택 의존관계 $(f1,\{f2,f3\})\in C_{OR}$	$QF_k(f2)=Q_k(P\{f1,f2,f3\})-Q_k(P\{f1,f3\})$ $QF_k(f3)=Q_k(P\{f1,f2,f3\})-Q_k(P\{f1,f2\})$ $QF_k(f1)=Q_k(P\{f1,f2\})-Q_{f2}$ $-Q(P\{parent(f1)\})$

본 논문에서는 휘처별 품질속성의 영향도 측정을 위해 Siegmund et. al[6]의 방법을 사용한다. 이 방법의 기본적인 아이디어는 제품휘처형상 별로 특정 품질속성 값을 측정하고 두 제품휘처형상 간의 차이가 나는 휘처의 품질속성의 영향도는 두 제품휘처형상 간의 품질속성 값의 차로 측정한다.

휘처모델의 휘처 별로 품질속성 값을 계산하는 식은 표1과 같고, 휘처모델의 최상위 휘처 부터 시작하여 의존관계를 가지는 서브휘처로 확장하면서 계산한다. 이때, 휘처 $f1$과 $f2$를 포함한 제품휘처형상 집합을 $1,f2\}$으로, $Q(P\{f1,f2\})$는 제품휘처형상 $P\{f1,f2\}$에 대한 품질속성 Q_k의 값을 나타낸다고 하고, 특정한 휘처 $f1$이 품질속성 Q_k에 미치는 영향도는 $QF_k(f1)$으로 나타낸다.

3. 휘처 상호작용

앞서 기술한 휘처 별 품질속성 값의 계산은 휘처 들이 상호 독립적인 것을 가정한다. 만약 하나의 휘처에 대한 품질속성 값이 다른 휘처의 존재 유무에 따라 다른 값을 가지게 된다면, 이들 휘처는 특정 품질속성 관점에서 상호작용 함을 나타낸다. 따라서 품질속성 관점에서 휘처 상호작용은 다음과 같이 정의된다.

정의 1. 품질속성관점의 휘처 상호작용

$\sum_{i=1}^{n} Q_k(P\ f_i\}) \neq Q_k(P\{f_1,\cdots,f_n\})$ 이면, 휘처 $f_1,\cdots,f_n$들은 품질속성 Q_k에 대해서 상호작용을 한다고 정의하고, 이때, 휘처 상호작용이 품질속성이 미치는 영향은 다음과 같이 계산된다.

$$I\ (\{f_1,\cdots,f_n\}) = \sum_{i=1}^{n} Q_k(Pf_i) - Q_k(P\{f_1,\cdots,f_n\})$$

휘처 간의 상호작용은 휘처 사이의 공유되는 부분이 존재하기 때문에 발생된다. 예를 들면, 두 휘처의 실행시간을 측정할 때, 두 휘처가 공유하는 부분이 존재하고, 공유되는 부분의 실행결과가 다른 휘처의 실행 시에 이용된다면, 각 휘처의 개별 실행 시간의 합이 두 휘처를 조합하여 실행한 전체 실행 시간보다 크게 될 수 있다.

휘처 간에 존재하는 상호작용의 검출은 크게 두 가지 방법으로 가능하다. 첫 번째 방법은 휘처모델로부터 도출 가능한 모든 제품휘처형상에 대해서 제품휘처형상의 품질속성 값이 해당 제품휘처형상을 구성하는 각 휘처의 품질속성 영향도의 합과 같은 지를 판별하여 얻는 방법이 있고, 두 번째 방법은 도메인 전문가의 지식을 이용하여 상호작용하는 휘처 조합을 알아내는 것이다. 첫 번째 방법은 모든 가능한 제품휘처형상에 대해서 품질속성 값을 계산하여야 하므로, 휘처의 수가 많은 경우에는 매우 많은 시간이 걸리는 단점이 있지만, 모든 가능한 휘처 상호작용을 검출할 수 있는 장점이 있다. 반면에 두 번째 방법은 도메인 전문가에 따라서 정확도가 다소 떨어질 수는 있지만 중요한 휘처들 간의 상호작용을 비교적 정확하게 도출해 낼 수 있고, 상호작용이 예상되는 휘처 조합에 대해서만 품질속성 값을 측정함으로써 실질적인 상호작용이 일어나는 지를 확인할 수 있다.

Ⅲ. 휘처 상호작용을 고려한 제품휘처형상 도출

본 논문에서는 측정된 휘처 별 품질속성 값과 휘처 상호작용을 바탕으로 요구되는 품질속성을 최적화 시키는 제품휘처형상을 도출하는 방법을 제안한다. 이를 위해서 최적의 제품휘처형상 도출 문제를 기존에 잘 알려진 최적화 문제를 통해 풀고자 한다. 기존의 많은 방법들은 최적의 제품휘처형상을 도출하기 위해서 휘처상호작용을 고려하지 않았다. 따라서 이를 위한 최적화 문제 또한 휘처 상호작용을 고려한 문제가 아니다. 하지만 본 논문에서 휘처 상호작용을 고려한 최적해 도출이 목적이므로, 비선형 프로그래밍의 최적화 문제에 해당 문제를 대응시켜서 휘처 상호작용을 고려한 최적의 제품휘처형상을 도출하는 기법을 제안한다.

정의 2. 이진 정수 비선형 계획법은 다음과 같이 목적함수와 제약사항으로 문제로 정의한다.

$$\min \quad (x)$$

subject to:

$$x \le b$$
$$A_{eq}x = b_{eq}$$
$$x_j \quad 0,1$$

이때, $f(x)$는 비선형식으로 정의된 목적 함수이고, x는 $\times 1$ 결정변수들의 벡터로서 정수 또는 0 이나 1의 값을 가진다. 또한 A, A_{eq}는 계수 행렬을, b, b_{eq}는 계수 벡터를 나타낸다.

가. 목적함수 정의

목적함수는 크게 두 부분으로 구분된다. 첫 번째 부분은 휘처 간의 상호작용을 고려하지 않고 정의된 식이고, 두 번째 부분은 휘처 간의 상호작용을 고려한 식이다.

먼저 특정한 품질속성 에 대해서 각 휘처 별 품질속성 값을 바탕으로 상호작용을 고려하지 않고 정의된 목적함수 $f \ (x)$는 다음과 같다.

$$f \ (x) = \sum_{j}^{n} QF_{kj}x_j \tag{1}$$

이때, x_j는 휘처를 나타내는 의사결정 변수로 해당 휘처가 특정 제품휘처형상을 위해 선택되면 1, 그렇지 않으면 0인 값을 가지고, QF_{kj}는 휘처 x_j가 품질속성 Q_k에 미치는 영향도를 나타낸다.

식 (1)을 여러 품질속성을 고려하여 정의한다면, 다음과 같이 정의된다.

$$f(x) = \sum_{k}^{m} f_{Q_k}(x) = \sum_{k}^{m}\sum_{j}^{n} QF_{kj}x_j \tag{2}$$

식 (2)는 품질속성 간의 중요도를 고려하지 않은 식이므로, 각 품질속성 간의 중요도를 나타내는 가중치(0에서 1사이의 값)를 각 품질속성 값에 곱하여 가중치가 고려된 목적함수를 얻게 된다. 이 때, 휘처 별로 측정된 품질속성 값이 정규화되지 않은 값이므로, 각 휘처 별 품질속성 값에 대해서 정규화 과정을 거치고, 이 정규화된 휘처 별 품질속성 값에 가중치를 곱하여 전체 목적함수를 얻게 된다.

정규화 된 휘처별 품질속성 값은 각 품질속성 별로 휘처의 품질속성 값이 최대인 것을 1로 최소인 것을 0으로 하여 정규화 시킨다. 따라서 정규화 된 휘처 별 품질속성 값은 다음과 같이 정의된다.

$$NQF_{kj} = \frac{QF_{kj} - \min(QF_{kj})}{\max(QF_{kj}) - \min(QF_{kj})} \tag{3}$$

따라서 가중치를 고려한 목적함수는 다음과 같이 정의된다.

$$f(x) = \sum_{k}^{m} w_k \sum_{j}^{n} NQF_{kj}x_j \tag{4}$$

하지만 식 (4)는 휘처 간의 상호작용을 고려하지 않은 식이다. 휘처 간의 상호작용을 고려하기 위해서 다음과 같이 위 목적함수를 확장해야 한다.

$$f(x) = \sum_{k}^{m} w_k \left(\sum_{j}^{n} NQF_{kj}x_j + \sum_{I \in FI} I_{Q_k}(I) \prod_{l \in I} x_l \right) \tag{5}$$

이때, 집합 FI는 상호작용을 하는 휘처들의 집합을 나타낸다.

나. 제약사항 정의

휘처 모델은 유효한 제품휘처형상을 도출하기 위한 제약사항을 제품휘처 간의 의존관계로 표현한다. 따라서, 휘처 간의 의존관계를 정의 2의 제약사항으로 변환하는 규칙은 다음과 같다.

- 요구 의존관계에 있는 두 휘처 x_1과 x_2는 x_1이 선택되면 x_2는 반드시 선택되어야 함을 나타내므로, 제약사항은 다음과 같이 정의된다.
  $$x_1 \le x_2 \tag{6}$$
- 필수 의존관계에 있는 두 휘처 x_1과 x_2는 반드시 같이 선택되던지 같이 선택되지 않아야 하므로, 제약사항은 다음과 같이 정의된다.
  $$x_1 = x_2 \tag{7}$$
- 배타 의존관계에 있는 휘처 x_1과 x_2는 동시에 선택될 수 없으므로, 제약사항은 다음과 같이 변환된다: $x_1 + x_2 \le 1$ (8)
- 선택 의존관계에 있는 휘처 x_2는 x_1이 선택된 경우에 선택될 수도 있고 선택되지 않을 수도 있으므로, 제약사항은 다음과 같이 정의된다.
  $$x_2 \le x_1 \tag{9}$$
- 택일 의존관계에 있는 휘처 x_0와 휘처그룹 $x_i \, (i = 1, \cdots, k)$는 휘처 x_0가 선택되었을 때, 휘처그룹의 단 한 휘처만 선택되어야함을 나타내므로, 제약사항은 다음과 같이 정의된다.
  $$x_0 = \sum_{i=1}^{k} x_i \tag{10}$$
- 다중선택 의존관계에 있는 휘처 x_0와 휘처그룹 $x_i \, (i = 1, \cdots, k)$는 휘처그룹의 휘처 중에서 하나 이상이 선택되어야 함을 나타내므로, 제약사항은 다

음과 같이 정의된다.

$$_{1,\cdots,k\}}x_i \le x_0, x_0 \le \sum_{i=1}^{k} x_i \qquad (11)$$

이와 같이, 주어진 휘처 상호작용을 고려하여 품질속성을 최적화 시키는 제품휘처형상 도출 문제를 식 (5)의 목적함수와 식 (6)-(11)의 제약사항으로 구성된 비선형 계획법 문제로 대응시키고, 비선형 계획법의 해를 구해주는 해결자 (Solver)를 이용하여 최적해를 구할 수 있다. 이에 대한 구체적인 사례연구는 다음 장에서 기술한다.

Ⅳ. 사례 연구 및 평가

본 장에서는 앞에서 설명한 휘처 상호작용을 고려한 제품휘처형상 도출 기법을 평가하기 위해서 네 개의 실제 프로덕트 라인의 사례를 사용하였다. 실험으로 사용된 사례는 Java 언어로 작성된 공개 소프트웨어로서 코드가 사용가능하고, 여러 프로덕트 라인 공학 방법에서 적용한 사례이므로 선택되었다.

2. 평가를 위해 사용된 소프트웨어 프로덕트 라인
Table 2. Software product lines used for evaluation

소프트웨어 프로덕트 라인	도메인	언어	라인수	휘처수	제품형상수
Prevayler	데이터베이스	Java	4.0K	6	24
Scientific Calculator	계산기	Java	6.5K	8	192
LinkedList	자료구조	Java	2.5K	16	492
Berkeley DB JE	데이터베이스	Java	42.6K	32	400

본 실험에서 사용된 프로덕트 라인에서 측정된 품질속성은 공통으로 CPU 사용량, 메모리 사용량, 바이너리 크기로 정하였다. 프로덕트 라인 별로 요구되는 품질 속성이 다른 것이 일반적이나, 동일한 품질 속성 관점에서 네 개의 프로덕트 라인을 비교 평가하기 위해서 세 개의 품질 속성을 측정하였다.

각 휘처 별 품질 속성 값을 측정하기 위해서 표1의 식을 이용하였다. CPU 사용량 측정은 JMX (Java Management eXtensions)을 이용하였고, 메모리 사용량은 Java의 Runtime 클래스를 사용하였다. 마지막으로 바이너리 크기는 클래스 파일의 크기를 측정하여 계산하였다.

측정된 휘처 별 품질 속성 값을 식 (3)을 이용하여 정규화한 후에 품질 속성 별 가중치와 휘처 상호작용을 고려하여 식 (5)의 형식으로 목적함수를 정의하였다. 마지막으로 각 프로덕트 라인의 제약사항을 식 (6)-(11) 형태로 정의함으로써, 제품휘처형상 문제를 이진 정수 비선형 계획법 문제로 대응시켰다.

이진 정수 비선형 계획법의 최적해를 구하기 위해서, OPTI Toolbox에서 제공하는 혼합 정수 비선형 계획법 해결자를 이용하였다. 표 2의 네 가지 사례에 대해 최적의 제품휘처형상을 도출하기 위해 걸린 시간을 측정한 결과는 그림 1과 같다.

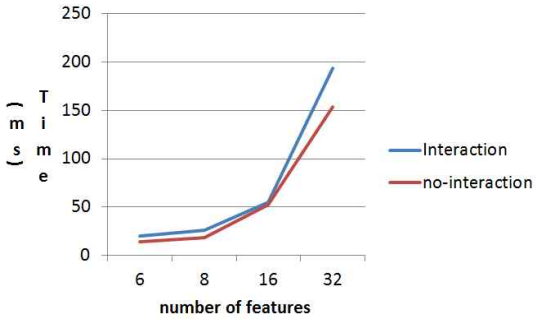

그림 1. 최적 제품휘처형상 도출을 위한 성능 결과
Fig. 1. Performance results for deriving an optimal product feature configuration

그림 1은 휘처 상호작용을 고려하여 모델링된 문제에 대한 최적해와 휘처 상호작용을 고려하지 않고 모델링된 경우의 성능을 비교 평가한 결과이다. 각각의 경우 신뢰도 있는 값을 얻기 위해서 10번의 수행결과의 평균값을 채택하였다.

결론적으로 휘처의 수가 증가할 수도록 최적해를 구하는 시간은 증가하지만, 휘처 상호작용을 고려한 경우와 고려하지 않는 경우의 시간 차이는 그리 크지 않은 것을 확인할 수 있다. 이러한 결과가 나오는 이유는 각 품질속성에 대해서 휘처 간의 상호작용을 가지는 휘처의 조합이 많지 않기 때문에 휘처 상호작용을 고려하여 정의된 목적함수와 그렇지 않은 경우의 목적함수의 복잡도의 차이가 그리 크지 않기 때문이다.

하지만, 휘처 상호작용을 고려하여 모델링 된 문제의 경우, 휘처 상호작용을 고려하지 않고 모델링 문제에 비

해 최적해를 도출하는 정확도가 향상되는 장점이 있다.

Ⅴ. 결 론

품질속성을 최적화시킬 수 있는 제품휘처형상 도출은 프로덕트 라인 공학에서 매우 중요한 문제이다. 본 논문에서는 휘처 간의 상호작용을 고려하여 최적의 제품휘처형상을 도출하는 방법을 제안하였다.

본 논문의 주된 공헌은 다음 두 가지로 요약된다. 첫째, 휘처 간의 상호작용을 고려하여 최적의 제품휘처형상을 도출하는 문제를 최적화 기법인 이진 정수 비선형 계획법으로 대응시킴으로써, 많은 최적화 도구를 이용하여 효과적으로 제품휘처형상을 도출하는 방법을 개발한 것이다. 둘째, 네 가지 실제 프로덕트 라인 사례에 제안된 기법을 적용함으로써, 제안된 기법의 효율성을 입증하였다.

References

[1] P. Clements, L. Northrop, “Software Product Lines: Practices and Patterns”, Addison-Wesley, 2002

[2] P. van den Broek, “Optimization of Product Instantiation using Integer Programming”, Proceedings of the 14th International Software Product Line Conference, Vol. 2, 14 Sep 2010, Jeju Island, South Korea. pp. 107-111.

[3] K. Lee, “Deriving an Optimal Product Feature Configuration Using Binary Integer Programming”, Journal of KIISE: Computing Practices and Letters, Vol. 17, No. 4, April 2011, pp. 254-258.

[4] G. Zhang, H. Ye,, Y. Lin, “Quality Attributes Assessment for Feature-Based Product Configuration in Software Product Line”, Proceedings of the 2010 Asia Pacific Software Engineering Conference, 2010, pp. 137-146.

[5] K. C. Kang, et al. “Feature-oriented domain analysis(FODA) feasibility study”, No. CMU/SEI-90-TR-21. CARNEGIE-MELLON UNIV. PITTSBURGH PA SEI, 1990.

[6] N. Seigmund, et al., “Scalable Prediction of Non-functional Properties in Software Product Lines”,. Proceedings of the 2011 15th International Software Product Line Conference, 2011, pp.160-169.

소개

관 우(정회원)

- 2003년 : POSTECH, 박사
- 2008년 : University of Limerick, 방문교수
- 2003년 ~ 현재 : 한성대학교 정보시스템공학과, 부교수
- 주관심분야: Software Product Line Engineering, M2M, Aspect-Oriented Programming, Software Architecture

※ 본 연구는 한성대학교 교내연구비 지원 과제임.