

# Bayesian Belief Network를 이용한 아키텍처 전술 품질

## 평가 방법

이정빈\*, 이동현\*, 김능희\*, 인호\*

\*고려대학교 정보통신대학 컴퓨터학과

e-mail:{jungbini, tellmehenry, nunghoi, hoh\_in}@korea.ac.kr

## Quality Evaluation of Architecture Tactics using Bayesian Belief Network

Jung-Been Lee\*, Dong-Hyun Lee\*, Neung-Hoe Kim\*, Hoh Peter In\*

\*Dept of Computer Science & Engineering, Korea University

### 요 약

소프트웨어 아키텍처는 소프트웨어의 품질에 지대한 영향을 미치는 요소 중 하나이다. 소프트웨어 개발 생명주기 초기에 아키텍처를 분석하고 평가하지 않으면, 점점 품질결함을 발견하고 수정하는 비용이 증가한다. 기존 소프트웨어 아키텍처 분석 및 평가 방법은 아키텍처라는 상당히 추상화된 수준에서 분석 및 평가가 이루어지기 때문에 평가기준이 주관적이며, 선택된 아키텍처 후보들만으로 서로에게 미치는 품질속성의 영향을 파악하기 힘들다. 따라서 품질 속성 시나리오나 아키텍처 전략을 구현하기 위한 세부적인 아키텍처 전술들의 품질평가가 필요하다. 본 연구는 이러한 아키텍처 전술의 품질 평가를 위해, Q-SIG(Quantified Softgoal Interdependency Graph)을 이용한 품질속성과 이를 달성하기 위한 아키텍처 전술의 관계를 정성적, 정량적으로 표현한다. 또한 Bayesian Belief Network(BBN) 모델 구축을 통해 Q-SIG에서 표현할 수 없는 다수의 품질속성을 만족하는 아키텍처 전술들 간의 조합에 대해 분석하고, 평가하여 아키텍처가 소프트웨어 디자인 단계에서 높은 품질속성을 달성할 수 있는 아키텍처 전술들의 조합을 선택할 수 있는 방법을 제시한다.

### 1. 서론

소프트웨어의 품질과 직결되는 비기능적 요구사항은 개발 초기에 정의 및 측정이 어렵기 때문에 기능적 요구사항에 비해 소프트웨어 개발 생명주기의 후반에 고려되는 경향이 있다. 특히 소프트웨어 아키텍처는 소프트웨어의 품질에 영향을 미치는 중요한 요소로써 소프트웨어 개발 생명주기 초기에 분석되고, 설계되어야 한다.

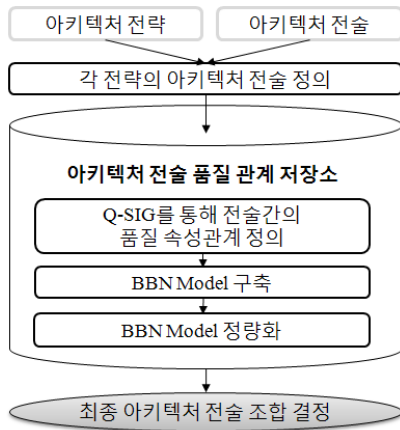
하지만 소프트웨어 개발 생명주기 초기에 수행되는 아키텍처 평가방법인 ATAM (Architecture Tradeoff Analysis Method)와 CBAM (Cost-Benefit Analysis Method)[1]은 품질속성을 달성을 위한 시나리오 기반의 아키텍처 평가는 이해관계자들의 경험적 직관에 의존하는 경향이 짙고, 아키텍처의 평가기준이 합리적인지 검증하기 어렵다. 또한 평가에 의해 선별된 아키텍처 전략의 후보들이 각각의 품질속성들에 미치는 영향에 대해서 판단이 힘들기 때문에, 여전히 시스템 전반적인 품질에 대한 위험요소를 안고 있다.

본 논문에서 이러한 아키텍처 전략을 구현하기 위한 결정 사항으로써 아키텍처 전술[2]의 품질 평가 방법을 통해 아키텍처 전략의 품질을 비교·평가 할 수 있는 방법을 제시한다. 기존의 방법론을 통해 아키텍처 전략에 대한 평

가를 수행한 후, 구체적인 아키텍처 전술의 품질 평가를 위한 Q-SIG (Quantified Softgoal Interdependency Graph) [3]과 아키텍처 전술 조합에 따른 품질의 불확실성을 예측하는데 도움을 주기 위해 이미 많은 도메인 영역에서 성공적으로 적용된 BBN (Bayesian Belief Network) 모델[4]을 통해 아키텍처 전술 후보들을 평가한다.

### 2. BBN을 이용한 아키텍처 전술 품질 평가 방법

하나의 아키텍처 전략을 달성하기 위해서는 여러 가지 아키텍처 전술의 조합이 필요하다. 따라서 기존의 방법론 [1]에 따라 선별된 아키텍처 전략을 달성하기 위해 구체적인 아키텍처 전술의 품질 평가를 수행한다. <그림 1>은 이러한 아키텍처 전술 품질을 평가할 수 있는 아키텍처 품질 평가인 BBN 기반 아키텍처 전술 품질 평가 절차를 나타내고 있다. 구체적인 평가 절차에 앞서 선별된 아키텍처 전략들과 품질속성 달성에 필요한 아키텍처 전술들을 수집한다. 그 후, 해당 전략의 품질속성 목표를 달성하기 위한 아키텍처 전술을 정의한다. 그 후, 정의된 아키텍처 전술을 다음과 같은 과정을 통해 최종적으로 그 조합들을 결정한다.



(그림 1) BBN 기반 아키텍처  
전술 품질 평가 절차

① Q-SIG를 통해 전술간의 품질속성 관계 정의: 해당 품질속성을 달성하기 위해 구현할 수 있는 전술들의 관계를 정성적으로 표현(AND, OR)한다. 하지만 Q-SIG은 다른 품질속성의 전술이 동시에 선택되었을 때의 서로간의 조합에 따른 품질속성의 영향력은 알 수 없기 때문에 이를 위해 BBN 모델을 구축한다.

② BBN 모델 구축: 이렇게 모델링 된 BBN의 각 노드는 부모 노드에 있는 변수에 대한 조건부 확률을 가지고 있다. 예를 들어, 보안(AT<sub>S</sub>)과 정확성(AT<sub>A</sub>)에 관련된 아키텍처 전술 간의 조건부 확률을 고려하면, 이 두 전술이 어떻게 조합되느냐에 따라 성능이 좌우되므로 다음과 같은 수식을 통해 조건부 확률을 구할 수 있다.

$$P(\text{성능} | AT_S, AT_A)$$

③ BBN Model 정량화: 모델링된 BBN을 통해 각 품질속성을 만족하는 아키텍처 전술들 간의 조건부 확률을 구한다.

### 3.4. 최종 아키텍처 전술 조합 결정

아키텍처 전술들의 조합을 바탕으로 각 품질속성을 평가하고, 타겟 시스템에서 중요시 여기는 품질속성을 우선 순위로 고려하여 최종적인 아키텍처 전술 조합을 결정한다. <표 1>은 특정 아키텍처 전략을 달성하기 위한 아키텍처 전술의 조합을 품질속성 별로 판단할 수 있도록 도와주는 아키텍처 전술 조합 비교표의 예이다.

아키텍처 전략 1(AS<sub>1</sub>)를 구현할 수 있는 아키텍처 전술의 조합은 두 조합 모두 시스템의 품질을 높이는 전술의 조합이라고 판단할 수 있었다. 이는 타겟 시스템이 중요하게 여기는 성능의 가중치(0.24)와 가용성의 가중치(0.23)가 비슷하기 때문에, 어떤 조합을 선택하더라도 시스템의 성능과 가용성에 큰 영향을 주지 않는다.

(표 1) 아키텍처 전술 조합 비교표의 예

| AS              | 전술 조합  | 성능<br>(0.24) | 가용성<br>(0.23) |
|-----------------|--|--------------|---------------|
| AS <sub>1</sub> | (AT <sub>P1</sub> +AT <sub>P2</sub> ) + AT <sub>A1</sub> | (+)59%       | 41%           |
|                 | (AT <sub>P1</sub> +AT <sub>P2</sub> ) + AT <sub>A2</sub> | 41%          | (+) 59%       |
| AS <sub>2</sub> | AT <sub>P3</sub> + AT <sub>A3</sub>                      | (+) 53%      | (+) 61%       |
|                 | AT <sub>P4</sub> + AT <sub>A4</sub>                      | 47%          | 39%           |
|                 | AT <sub>P3</sub> + AT <sub>A3</sub>                      | 47%          | (+) 61%       |

반면 아키텍처 전략 2(AS<sub>2</sub>)를 구현할 수 있는 아키텍처 전술의 조합은 가용성을 달성할 수 있는 전술 3(AT<sub>A3</sub>)과 품질을 달성할 수 있는 전술 3(AT<sub>P3</sub>)의 조합이 3가지 전술의 조합 중에서 가장 높은 성능과 가용성의 품질을 보장해 준다.

## 4. 결론

본 논문에서는 Q-SIG와 BBN 모델을 이용하여 아키텍처 전략을 실현할 수 있는 아키텍처 전술의 품질을 평가하는 방법을 제시하였다. Q-SIG를 통해 아키텍처 전술을 정성적·정량적으로 표현하고, BBN 구축을 통해 Q-SIG에서는 판단할 수 없었던 다수의 품질속성을 달성하기 위한 아키텍처 전술들의 조합이 각 품질속성에 미치는 영향을 확률적으로 계산하고 평가한다. 최종적으로 아키텍처들은 아키텍처 전술 조합의 비교를 통해 소프트웨어 품질을 높일 수 있는 최적의 아키텍처 전략들의 조합을 선택할 수 있다.

## 사사(謝辭)

본 연구는 정보통신산업진흥원의 SW 공학 요소기술 개발과 전문인력 양성사업의 결과물임을 밝힙니다.

## 참고문헌

[1] L.Bass, P.Clements, and R. Kazman. "Software Architecture In Praticce 2nd Edition", Addison Wesley, 2003.  
 [2] S. Kim, D. Kim, Lunjin Lu, S. Park, "A Tactic-Based Approach to Embodying Non-functional Requirements into Software Architectures", 12th International IEEE Enterprise Distributed Object Computing Conference, p.139-148, 2008.  
 [3] Tegegne Marew, J.S., Lee, D.H., Bae, "Tactics based approach for integrating non-functional requirements in object-oriented analysis and design", Journal of Systems and Software , Vol.82(10), 2009  
 [4] Hongyu Z., Stan J. and Bo Yang, "Quality Prediction and Assessment for Product Lines", 15th International Conference, p.681-695, 2003.