

개발자 중심의 평가 방법을 활용한 아키텍처 설계 방법론

윤성춘*, 고현희**, 궁상환***, 박재년**
*안양대학교 컴퓨터학과
**숙명여자대학교 컴퓨터과학과
***천안대학교 정보통신학과
e-mail : hkhoh@sookmyung.ac.kr

Architecture Design Method Using Developer Centric Evaluation Method

*Yoon Sung Chun, **Koh Hyon Hee, ***Kung Sang Hwan, **Park Jae Nyun
*Dept. of Computer Science, AnYang University
**Dept. of Computer Science, Sookmyung University
***Dept. of Information and Communication, CheonAn University

요 약

복잡하고 방대한 시스템 개발에서 아키텍처의 설계는 성공적인 프로젝트를 위한 중요한 이슈가 되었다. 본 논문에서는 아키텍처 설계를 위해 가장 적합한 아키텍처 접근법을 선정하기 위한 평가 방법을 제안하고, 설계 단계에서 본 평가 방법을 설계 방법론에 통합하여 활용함으로써 아키텍처 설계의 신뢰성과 완전성을 높일 수 있는 아키텍처 설계 방법론을 제안한다

1. 서론

복잡하고 방대한 시스템 개발에서 다양한 이해관계자들의 요구사항을 시스템에 정확히 반영해야 하고, 이를 위해 시스템의 품질 속성과 이해관계자들의 이해관계를 반영한 소프트웨어 아키텍처의 설계가 성공적인 프로젝트를 위한 중요한 이슈가 되었다

이에 따라 성공적인 아키텍처 설계를 위한 다양한 방법과, 설계된 아키텍처의 적합성을 판단하기 위한 여러 평가방법이 연구되어지고 있다. 본 연구에서는 아키텍처 설계자 또는 개발자가 설계과정 중에 자신이 담당할 부분의 아키텍처 설계를 위한 가장 적합한 아키텍처 접근법을 찾기 위한 평가모형을 제안하고 또한 설계 단계에서 아키텍처 접근법 선정을 위한 평가 방법을 활용하는 아키텍처 설계 방법론을 제안하고자 한다.

2. 관련연구

2.1 아키텍처 설계방법

아키텍처 설계 방법은 미국 Carnegie Mellon 대학의 SEI(Software Engineering Institute)에서 연구 개발한 아키텍처 기반 설계 방법(ABD)과 품질 속성에 기반 한 아키텍처 설계 방법(ADD)을 대표적으로 볼 수 있다. ABD(Architecture Based Design)와 ADD(Attribute Driven Design)는 기능요구사항과 품질 요구사항을 기반으로, 사용자의 요구사항을 최상으로 실현시키는 아키텍처를 설계하고자 하는 방법이다.

ABD 는 상위 수준의 소프트웨어 아키텍처를 설계하기 위한 방법으로 특정한 제품을 만들 때 다양성에 대한 필요를 허용하는 추상적인 수준에서, 기능적 요구사항, 품질 요구사항, 그리고 비즈니스 요구사항을 만족하도록 한다[2].

ABD 방법론의 수행단계는 다음과 같다.

1. 기능을 분할 한다.
2. 아키텍처 스타일을 선택한다.
3. 기능을 스타일에 할당한다.
4. 템플릿을 정제한다.
5. 기능을 검증한다.
6. 동시성(Concurrency) 뷰를 생성한다.
7. 배치(Deployment) 뷰를 생성한다.
8. 품질 시나리오를 검증한다.
9. 계약사항을 거즈한다.

ADD(Attribute Driven Design) 방법론은 스템을 위한 아키텍처를 설계하기 위한 접근법으로 기능요구사항과 품질 요구사항 둘 다에 그리고 다른 시스템에서 이러한 품질을 달성하는데 성공적이었고 증명된 아키텍처 접근법을 아는 것에 기반을 두고있다[8]. 즉, ADD 방법론은 아키텍처 패턴을 통해 어떻게 품질 목표를 달성할 것인가를 이해하는데 기반을 두고 있다.

ADD 방법론의 수행 단계는 다음과 같다 ...

- 단계 1: 분해할 모듈을 선택한다.
- 단계 2: 선택한 모듈을 분해하고 정제한다.
 - 단계 2a. 아키텍처 동인을 결정한다.
 - 단계 2b. 아키텍처 스타일을 선택한다.
 - 단계 2c. 모듈을 실체화하고 기능을 할당한다.
 - 단계 2d. 하위 모듈의 인터페이스를 정의한다.
 - 단계 2e. 유스케이스와 품질 요소를 정제하고 검증해서 하위 모듈의 제약사항으로 바꾼다.
- 단계 3: 모듈을 분해할 필요가 없을 때까지 반복한다.

2.2 아키텍처 평가 방법

아키텍처는 초기 설계 단계의 산출물이고, 어떻게 아키텍처를 설계하느냐에 따라 시스템과 프로젝트에 미치는 영향이 크다. 따라서 아키텍처 평가는 아키텍처 설계가 가져올 수 있는 프로젝트의 재앙을 피할 수 있는 가장 경제적인 방법이다.

아키텍처 평가의 결과는 아키텍처가 시스템에 적절하게 설계되었는지, 시스템을 위해서 가장 적절한 아키텍처가 무엇인지, 즉 아키텍처의 적합성을 판단하는 것이다[7]. 아키텍처 평가방법에는 평가 방법이나, 평가 대상, 평가의 구체성에 따라 여러 방법들이 나와 있다. 그 중 아키텍처 접근법 선정을 위한 평가 방법으로 CBAM 이 있다.

비용과 이득을 고려한 아키텍처 평가 방법인 CBAM 은 아키텍처 접근법을 실현하는 데 필요한 비용과 아키텍처 접근법을 적용했을 때 달성할 수 있는 품질 속성이 가져다 주는 이익을 측정하여 '투자대비효과(ROI)'를 계산한다.

CBAM 은 주로 아키텍처 트레이드오프 분석 방법인 ATAM 등에 의해 아키텍처 설계에 대한 평가가 끝난 후 새롭게 제시된 대안 아키텍처 접근법들 중 ROI 가 높은 접근법 대안을 선택 할 수 있도록 하고 있다.

CBAM 의 실행 절차는 <표 1>과 같다.

<표 1> CBAM 의 실행 절차[5]

시나리오 결정	1. 시나리오를 수집한다.
	2. 시나리오를 정제한다.
	3. 시나리오의 우선순위를 결정한다.
효용-반응값 곡선 작성	4. 선별한 시나리오의 효용-반응값 곡선을 작성한다.
아키텍처 접근법 전체 이익 계산	5. 시나리오를 담당하는 아키텍처 접근법을 찾아서 예상 반응값을 결정한다.
	6. 아키텍처 접근법의 예상효용을 계산한다.
	7. 아키텍처 접근법의 전체 이익을 계산한다.
아키텍처 접근법 선정과 검증	8. 아키텍처 접근법의 ROI 를 계산하여 순위를 결정한다.
	9. 비용과 일정을 고려해서 아키텍처 접근법을 선정하고 결과를 검증한다.

2.3 아키텍처 설계 및 평가 방법의 문제점

기존의 아키텍처 설계 방법과 평가 방법은 설계가 끝난 후 아키텍처 평가 과정을 거치게 된다. 예를들어 아키텍처 트레이드오프 분석 방법인 ATAM 은 아키텍처 설계 시 사용된 아키텍처 접근법을 찾아서 그 접근법의 위험요소나 상충점들을 분석하고 새로운 아키텍처 접근법 대안을 제시한다. CBAM 은 ATAM 에 평가 이후 제시된 아키텍처 접근법 대안들을 가지고 품질 요구 사항에 미치는 효용과 비용을 고려하여 가장 적합한 아키텍처 접근법을 찾게 된다[9].

이때 선정된 아키텍처 접근법이 설계 시 사용한 아키텍처 접근법과 다를 경우 다시 설계 과정을 다시 시작하여야 하기도 하고, 품질 요구사항을 만족시키는 접근법이 없을 경우 다시 아키텍처 동인을 재조정하여야 하는 과정으로 되돌아와야 한다.

또한 아키텍처 평가팀을 구성하여 개발조직과의 합의를 통해 아키텍처를 평가하는 방법으로 아키텍처 설계자나 개발자가 자신이 맡고 있는 서브시스템이나 모듈 단위로 아키텍처를 검증해 볼 수 있는 소규모 단위의 아키텍처 평가 방법론으로 활용하기 힘들다,

따라서 설계자 또는 개발자가 설계 단계에서 품질 요구사항을 고려한 아키텍처 접근법 대안에 대한 고려를 하고 아키텍처 대안을 선정할 때 충분한 평가 과정을 거쳐 가장 적합한 아키텍처 접근법을 찾는다면 설계된 아키텍처의 완전성과 신뢰성이 높아질 뿐만 아니라 설계 이후의 평가과정에서 새로운 아키텍처가 선정되는 사례도 감소할 것이다.

3. 평가를 통합한 아키텍처 설계 방법

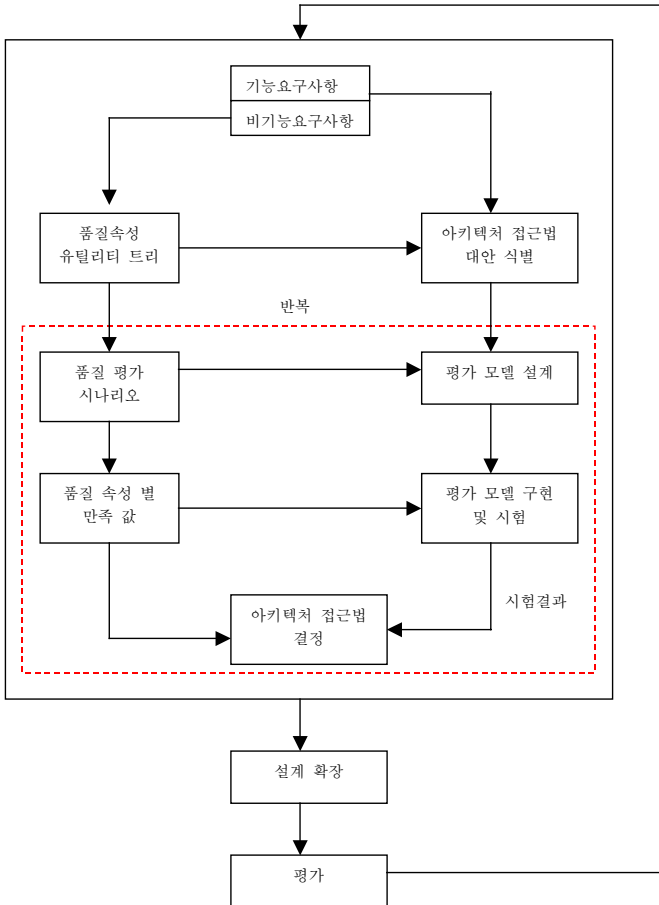
3.1 아키텍처 접근법 선정을 위한 평가모델

아키텍처 설계는 시스템이 만족해야 하는 기능요구사항과 품질 속성요구사항을 만족시키도록 설계되어야 한다. 아키텍처 설계를 위한 아키텍처

접근법은 품질 속성 우선 순위에 따라 우선순위가 높은 속성을 만족시키는 아키텍처 접근법을 먼저 식별해 나가게 된다. 즉 하나의 품질 속성 요구사항을 만족시키기 위한 아키텍처 접근법을 식별하고 다음으로 다른 품질 속성 요구사항을 만족시키는 아키텍처 접근법을 찾는 과정을 반복함으로써 아키텍처 설계를 확장해 나가게 된다.

설계 단계에서의 아키텍처 접근법 선정을 위한 평가 모델은 다음의 (그림 1)과 같다.

시스템의 기능 요구사항과 비 기능 요구사항으로부터 아키텍처 동인을 얻어 아키텍처 접근법을 식별하게 되고, 아키텍처 접근법이 여러 개의 대안이 존재할 경우 각 대안에 대한 평가 과정을 거쳐 해당 품질 속성에 가장 적합한 하나의 아키텍처 접근법을 선정하게 된다. 이 과정을 만족해야 하는 품질 속성들에 대해 반복하여 아키텍처 접근법들을 선정해 나가고, 이렇게 선정된 아키텍처 접근법에 기능을 할당하여 아키텍처를 확장 설계 하고, 설계된 아키텍처에 대해서는 다시 한번 평가 과정을 거쳐 최종적으로 설계를 완성하게 된다.



(그림 1) 아키텍처 접근법 선정을 위한 평가 모델

3.1 아키텍처 설계 절차

본 논문에서 제안하는 아키텍처 접근법 선정을 통한 아키텍처 설계 절차를 세부 단계로 정리하면 다음과 같다.

- 단계 1: 요구사항 정의
- 단계 2: 아키텍처 동인 결정
- 단계 3: 설계 뷰 선택
- 단계 4: 시스템 및 모듈 분해
- 단계 5: 아키텍처 접근법 선정
- 단계 6: 아키텍처 설계 확장
- 단계 7: 아키텍처 설계 평가

각 단계별 세부 실행 내용은 다음과 같다.

(1) 단계 1: 요구사항정의

요구사항은 시스템의 기능 요구사항과 비기능 요구사항으로 나누어 볼 수 있다. 기능요구사항은 시스템이 제공해야 하는 기능을 나타내고, 비 기능 요구사항은 시스템이 만족 시켜야 하는 품질 속성을 말한다. 이 단계에서는 요구사항을 시나리오 형태로 표현하고 정제하는 과정을 거친다.

(2) 단계 2: 아키텍처 동인 결정

아키텍처는 기능요구사항과 비기능 요구사항이 구체화된 것이다. 이렇게 아키텍처로 구체화 되는 요구사항을 아키텍처 동인이라고 한다[8]. 아키텍처 동인을 선정할 때는 가장 높은 우선순위를 가진 비즈니스 목표를 찾고, 찾은 비즈니스 목표를 품질 시나리오나 유스케이스로 바꾼다. 그리고 아키텍처에 가장 많은 영향을 미치는 시나리오나 유스케이스를 찾는다. 이 과정은 품질 속성 유틸리티 트리를 작성하는 과정을 통해 실행한다. 유틸리티란 시스템이 제공해야 하는 모든 품질을 의미하며, 품질 속성 유틸리티 트리는 시스템이 만족해야 하는 모든 품질 속성에 대한 시나리오를 순서대로 트리 형태로 나타낸 것이다.

(3) 단계 3: 시스템 및 모듈 분해

아키텍처 설계를 하기 위해 우선 시스템을 적절한 서브시스템으로 나눈다. 또한 각 서브시스템은 개발 단위의 모듈로 분해되고 이 과정은 적절한 크기의 모듈로 분해될 때까지 반복하게 된다[8]. 이렇게 분해된 모듈은 아키텍처 설계의 단위가 되며, 각 팀 또는 조직으로 할당되는 단위가 된다.

(4) 단계 4: 설계 뷰 선택

설계 뷰는 논리(Logical) 뷰(기능관점, 기능요구사항), 병행성(Concurrency) 뷰(동시성/병행처리 관점, 성능요구사항), 배치(Deployment) 뷰(하드웨어상의 소프트웨어 설치 관점)로 구분된다. 모듈별로 설계할 뷰 타입을 선택한다.

(5) 단계 5: 아키텍처 접근법 선정

아키텍처 동인에 따른 아키텍처 접근법은 하나 또는 여러 개의 대안이 나올 수 있다. 아키텍처 접근법 대안이 2 개 이상인 경우, 각 대안을 비교 평가 할 수 있는 평가 모델을 만들고 구현하여 각 품질 속성에 대해 접근법 대안의 효용을 비교 평가하여 아키텍처 접근법을 선정하여야 한다. 이 때 거치는 평가 과정은 3.1 절에서 제안한 평가 모델을 기반으로 하여 기존의 평가 방법에서 사용하는 검토(Review)방법 이외에

참고문헌

평가 모델에 의한 시험을 통해 보다 객관적인 평가를 하게 된다.

(6) 단계 6 : 뷰 타입별 아키텍처 설계
각 모듈별 선택된 아키텍처 스타일에 기능을 할당하여 실체화 함으로써 아키텍처를 설계를 확장하여 목표 시스템의 아키텍처를 설계한다[8].

(7) 단계 7 : 아키텍처 설계 평가
선택된 아키텍처 접근법에 따라 설계된 아키텍처를 평가한다. 즉 모듈이 충분히 분해 되었는지는, 뷰간 동기화가 되고 있는지, 설계산출물이 품질 요구사항을 만족시키고 있는지를 검증한다. 평가가 만족하게 이루어지는 경우 설계가 끝나고 설계 산출물이 나오게 되고, 평가 결과가 적합하지 않은 경우 설계 과정을 반복한다.

4. 결론 및 향후 연구 과제

본 논문에서는 아키텍처 설계자 또는 개발자들이 설계 과정 중에 활용할 수 있는 아키텍처 접근법 선정을 위한 평가 모델과 본 평가 방법을 활용한 설계 방법론을 제안하였다.

본 연구에서 제안하고 있는 아키텍처 설계 방법의 특징 및 장점은 다음과 같다.

첫째 기존의 아키텍처 설계 방법과 평가 방법을 수정, 보완하여 활용하고 있다. 즉, 아키텍처 동인을 결정에는 ATAM 에서 사용하고 있는 품질 속성 유틸리티 트리를 활용하여 요구사항의 분석을 체계적으로 할 수 있도록 하여 명확하게 아키텍처 동인을 찾을 수 있도록 하고 있다. 아키텍처 설계 방법은 ABD 나 ADD 에서처럼 시스템을 분할하고, 아키텍처 동인에 따라 뷰 타입별 아키텍처 설계를 해 나가는 방법을 활용하고 있다.

둘째 보다 객관적인 평가 방법을 활용하고 있다. 아키텍처 접근법 결정시에 평가 시나리오를 만들고 평가 모델의 구현 및 시험을 통해 결과를 비교 평가 함으로써 기존의 검토로 이루어 지던 평가 과정보다 객관적인 평가가 이루어 지도록 하고 있고, 이로 인해 선택된 아키텍처에 대한 신뢰성을 높여주고 있다. 또한 아키텍처 접근법 평가 뿐만 아니라 설계된 아키텍처의 완전성에 대해서도 다시 한번 평가 과정을 거침으로써 아키텍처 설계의 완전성을 높이고 있다.

셋째 설계 이후 이루어 지던 아키텍처 평가를 설계 단계로 통합하고 있다. 즉 설계 후 모든 이해 관계자들이 모여 이루어지던 기존의 아키텍처 평가방법을 아키텍처 설계 단계로 통합함으로써 아키텍처 설계 단계의 산출물에 대한 품질을 높일 수 있고, 평가 이후 설계의 문제가 발생한 경우 다시 재설계 해야 하는 비용도 줄일 수 있는 효과가 있다.

향후 연구과제로는 본 방법론을 실 사례에 적용하여 타당성을 검증하여야 하고, 적용 시 나타나는 문제점을 보완하는 연구가 이루어져야 한다.

- [1] Jayatirtha Asundi, Rick Kazman, Mark Klein, "Using Economic Considerations to Choose Among Architecture Design Alternatives" Technical Report CMU/SEI, 2001
- [2] Felix Bachmann, Len Bass, Gary Chastek, Patrick Donohoe, Fabio Peruzzi, "The Architecture Based Design Method" Technical Report CMU/SEI, 2000
- [3] Mario R. Barbacci, Mark H. Klein, Charles B. Weinstock "Principles for Evaluating the Quality Attributes of a Software Architecture" Technical Report CMU/SEI, 1997
- [4] Mario R. Barbacci, S. Jeromy Carriere, Peter H. Feiler, Rick Kazman, Mark H. Klein, Howard F. Lipson, Thomas A. Longstaff, Charles B. Weinstock, "Steps in an Architecture Tradeoff Analysis Method : Quality Attribute Models and Analysis", CMU/SEI, 1998
- [5] Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice Second Edition", Addison Wesley, 2003
- [6] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture : A System of Patterns volume1", Wiley, 1996
- [7] Paul Clements, Rick Kazman, Klein, "Evaluating Software Architectures : Methods and Case Studies", Addison Wesley, 2002
- [8] SEI " Attribute Design Method(ADD) Tutorial" SEI 2004
- [9] Robert L. Nord, Mario R. Barbacci, Paul Clements, Rick Kazman, Mark Klein, Liam O'Brien, James E. Tomayko "Intergrating the Architecture Tradeoff Analysis Method(ATAM) with the Cost Benefit Analysis Method(CBAM)" Technical Report CMU/SEI, 2003