

평가를 통한 아키텍처 접근법 선정 방법

고현희⁰ 공상환 윤성춘 박재년
숙명여자대학교, 천안대학교, 안양대학교
hhkoh@sookmyung.ac.kr

The Method to choose Architectural Approaches by Evaluation

Hyonhee Koh⁰ Sanghwan Kung, Sungchun Yoon, JaeNyon Park
Sookmyung Womens' University, Cheonan University, Anyang University

요 약

복잡하고 방대해진 시스템 개발에서 다양한 이해 관계자들의 요구사항을 시스템에 정확히 반영해야 하고, 이를 위해 시스템의 품질 속성과 이해관계자들의 이해관계를 반영한 소프트웨어 아키텍처의 설계가 성공적인 프로젝트를 위한 중요한 이슈가 되었다 본 연구에서는 아키텍처 설계 과정에서 식별된 아키텍처 접근법 대안들에 대해 시스템의 요구사항에 대한 만족도를 평가하여 설계단계에서 설계자들이 가장 적합한 아키텍처 접근법을 선정 하도록 하고 있다. 즉 평가를 통해 선정된 아키텍처 접근법을 기반으로 아키텍처 설계를 확장해 나감으로써 설계 산출물의 신뢰성과 완성도를 높이는 효과를 기대 할 수 있다.

1. 서 론

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

본 연구에서는 아키텍처 설계 과정에서 식별된 아키텍처 접근법 대안들에 대해 시스템의 요구사항에 대한 만족도를 평가하여 설계단계에서 설계자들이 가장 적합한 아키텍처 접근법을 선정 하도록 하고 있다. 즉 평가를 통해 선정된 아키텍처 접근법을 기반으로 아키텍처 설계를 확장해 나감으로써 설계 산출물의 신뢰성과 완성도를 높이는 효과를 기대 할 수 있다.

2. 아키텍처 접근법 선정의 중요성

아키텍처 설계의 기반이 되는 아키텍처 접근법은 품질 속성을 달성하기 위한 아키텍처 모델 또는 아키텍처 스타일의 선택을 통해 식별 될 수 있다.

아키텍처 스타일은 SEI의 Shaw와 Garlan에 의해 처음으로 아키텍처 스타일의 모음이 발표되었고 컴포넌트의 타입, 컴포넌트들 간의 상호작용 방식, 컴포넌트의 배치, 그리고 배치와 행위의 제약사항 등에 의해 표현이 된다. [그림 1]은 Shaw와 Garlan에 의해 발표된 아키텍처 스타일의 모음이다.

그러나 아키텍처 스타일은 아직까지 산업계로는 널리 알려지지 않았고, 사용이 되고 있다 하더라도 일반적이지

않거나 잘못 이해되는 사례도 많다[1]

아키텍처 접근법은 이러한 아키텍처 스타일이라는 용어보다 좀더 중립적인(neutral) 위치에서 아키텍처 설계의 기반을 위한 용어로 사용이 되고 있다[1]

이러한 아키텍처 접근법의 중요성은 다음과 같다.

첫째, 아키텍처 접근법 선정은 아키텍처 설계와 후속 설계작업의 성패를 좌우하게 된다.

소프트웨어 엔지니어는 아키텍처 모델에 근거하여 시스템 설계의 모든 측면을 토의한다. 아키텍처 모델을 개발할 때 내린 결정들은 나머지 설계 작업에 대하여 지대한 영향을 미친다. 아키텍처 모델은 설계의 핵심이며 참여한 모든 엔지니어가 이해하여야 한다[2]

아키텍처 모델은 시스템의 전체적인 효율성, 재사용성, 유지보수성에 지대한 영향을 미치는 구속력을 가지고 있다. 따라서 아키텍처 모델의 잘못된 결정은 후속 설계 작업의 성패를 크게 좌우한다[2]

아키텍처 설계자는 아키텍처 설계를 위해 먼저 시스템의 요구사항을 해결하기 위한 아키텍처 접근법을 식별하고, 식별된 아키텍처 접근법에 기능을 할당하여 아키텍처 설계를 확장해 나가게 된다. 따라서 모든 아키텍처 설계는 선정된 아키텍처 접근법에 따라 달라지게 된다.

이처럼 아키텍처 접근법은 아키텍처 설계의 기반이 되어 아키텍처 설계의 방향을 좌우하게 되고, 아키텍처 설계는 후속 설계 작업의 성패를 크게 좌우하게 되므로, 설계 초기에 가장 적합한 아키텍처 접근법을 선정하는 것이 설계의 성공여부를 좌우하는 중요한 일이라고 할 수

자료 흐름 (Dataflow system) 배치 순차 처리 (Batch sequential) 파이프와 필터 (Pipes and filters) 콜과 리턴 (Call-and-return systems) 메인프로그램과 서브루틴 (Main program and subroutine) 객체지향 시스템 (OO systems) 계층 시스템 (Hierarchical layers) 독립 컴포넌트 (Independent components) 프로세스간 통신 (Communicating processes) 이벤트 시스템 (Event systems) 가상 머신 (Virtual Machines) 인터프리터 (Interpreters) 규칙 기반 시스템 (Rule-based systems) 데이터 중심 (Data-centered systems) (Repository) 블랙보드 (Blackboards)	저장소
--	-----

[그림 1] 아키텍처 스타일의 분류

3. 아키텍처 접근법 선정을 위한 평가 절차

본 논문에서 제안하는 설계단계에서의 아키텍처 접근법 선정을 위한 평가 과정은 다음과 같다.

1. 관련된 품질 속성들을 그룹화 한다.
2. 품질 속성 우선순위를 결정한다.
3. 품질 속성 또는 품질 속성 그룹별 평가 시나리오를 작성한다.
4. 품질 속성별 최대/최소 만족값을 정의한다.
5. 품질 속성 또는 품질 속성 그룹별 만족도 수준을 정의한다.
6. 식별된 아키텍처 대안들에 대해 평가 시나리오 별 만족도 수준을 결정한다.
7. 아키텍처 접근법 선정 원칙에 따라 아키텍처 접근법을 선정한다.

각 단계별 세부 실행 내용은 다음과 같다

3.1 관련 품질 속성 그룹화

품질 속성들 간에는 상호 종속적이 될 수 있다. 예를 들어 확장성과 성능의 경우, 확장성을 높이기 위한 설계가 성능을 저하 시킬 수 있고, 성능을 높이기 위한 설계는 확장성을 저하 시킬 수 있다. 이 경우 하나의 품질 속성은 결국 다른 품질 속성 값의 결정에 영향을 줄 수 있다. 따라서 각각의 품질 속성에 대해 평가되기 보다는

두개의 품질 속성을 하나로 그룹화하여 두 품질 속성간의 적정 값으로 선정되도록 평가 되어야 한다.

3.2 품질 속성 우선 순위 결정

품질 속성 우선순위는 요구사항 분석단계에서 산출된 품질 속성 유틸리티 트리에서 결정된다고 볼 수 있다. 만약 결정이 되지 않았다면 이해관계자들에 의해 시스템이 만족시켜야 할 품질 속성의 우선순위를 결정하도록 한다.

이렇게 결정된 우선순위에 따라 아키텍처 설계를 위한 아키텍처 접근법 식별이 이루어지게 된다. 즉 가장 중요한 품질 속성을 만족시키기 위한 아키텍처 접근법이 먼저 선정이 되고, 다음으로 다른 품질 속성들을 위한 아키텍처 접근법들을 선정해 나가면서 아키텍처 설계를 발전시켜 나가게 된다.

3.3 평가 시나리오 작성

시스템이 만족해야 할 품질 속성과 품질 속성 그룹별로 평가 시나리오를 작성한다. 이 때 작성된 평가 시나리오는 평가 모델을 만들기 위한 기반이 된다. 품질 속성 시나리오의 비기능 요구사항 시나리오를 참조할 수 있다.

3.4 품질 속성별 최대/최소값 결정

먼저 품질 속성 별로 만족해야 하는 최대값과 최소값을 결정한다. 이 값은 시스템이 해당 시스템이 만족해야 하는 품질 속성의 최대/최소 한계 값으로 아키텍처 접근법의 시험 결과값이 이 한계값을 넘는 경우 해당 아키텍처 접근법은 선정될 수 없다.

3.5 만족도 수준 정의

품질 속성 또는 품질 속성 그룹에 대해 평가 시나리오의 결과값에 대한 만족도 수준을 정의한다. 만족도 수준 결정에는 설계자와 시스템과 관련된 이해관계자들의 의견을 반영하여 결정한다. 이해관계자들의 의견이 완전히 같을 수는 없지만, 시스템의 목적에 대한 이해도가 유사하다고 볼 수 있으며, 조정과정을 통해 만족도 수준을 결정한다.

3.6 아키텍처 접근법의 예상 만족도 수준 결정

이 단계에서는 품질 속성들에 대한 아키텍처 접근법의 예상 품질 속성값을 결정한다. 아키텍처 접근법의 예상 품질 속성값은 해당 품질 평가 시나리오에 대한 시험

모델의 구현과 시험을 통해 시험 결과를 가지고 시스템의 결과값을 예측해 볼 수 있다. 이 때 아키텍처 설계자와 개발자가 아키텍처 접근법의 시험 모델을 만들고 구현하여 그 결과를 가지고 가장 객관적인 값을 예측하도록 한다.

이렇게 예측된 값을 가지고 각 아키텍처 접근법 대안들의 품질 속성에 대한 만족도 수준을 결정한다.

3.7 아키텍처 접근법 선정 원칙

위의 모든 과정을 거쳐 아키텍처 접근법 대안들의 만족도를 구했으면 그 결과를 가지고 아키텍처 접근법을 선정하여야 한다. 기본 원칙은 품질 속성별 결과값이 최대/최소값 한계를 넘는 경우 선택에서 제외하며, 만족도가 가장 높은 아키텍처 접근법을 선택하도록 한다. 그러나 아키텍처 접근법의 만족도 결과에 관계없이 반드시 있어야 하는 아키텍처 접근법이 있을 수 있다. 예를 들어 개발 환경의 특성이나, 개발 언어의 특징에 의해 만족도가 낮더라도 개발의 편의성이나 시스템의 안전성을 위해 선택하여야 하는 아키텍처 접근법들이다. 이런 아키텍처 접근법들을 식별하여 우선 선정한다.

다음으로는 먼저 선정된 아키텍처 접근법과의 의존성을 파악하여 이미 선정된 아키텍처 접근법과의 의존성이 높은 경우 그 아키텍처 접근법 또한 우선적으로 선정하여야 한다.

위의 두 가지 제약에 의한 선정을 마친 후 나머지 아키텍처 접근법 대안들 중 높은 만족도를 나타내는 접근법을 선정한다.

4. 결론 및 향후 연구과제

본 논문에서는 소프트웨어 아키텍처 접근법 대안들을 식별하고 품질 속성 우선 순위에 따라 평가 과정을 반복함으로써 가장 적합한 아키텍처 접근법을 선정하기 위한 방법을 정의하고, 본 평가방법을 통해 설계단계에서 가장 적합한 아키텍처 접근법을 선정함으로써 설계 산출물의 신뢰도를 높일 수 있도록 하고 있다.

아키텍처 설계의 기반이 되는 아키텍처 접근법 선정을 위해 객관적인 평가방법을 활용하도록 하였다. 즉 평가 모델에 의한 시험과 그 결과를 가지고 아키텍처 접근법을 선정함으로써 선정된 아키텍처 접근법에 대한 신뢰도를 높이고 있다.

향후 연구과제로는 본 연구에서 제안한 평가방법을 보다 많은 실 사례에 적용하여 보고, 나타나는 문제점들에 대한 보완작업이 이루어져야 한다.

5. 참고 문헌

- [1] Paul Clements, Rick Kazman, Mark Klein, Evaluating Software Architectures : Methods and Case Studies, Addison Wesley, 2002.
- [2] 최은만, 객체지향 소프트웨어 공학, 사이텍미디어, 2005.
- [3] Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice Second Edition, Addison Wesley, 2003.
- [4] Mary Show, David Garlan, Software Architecture : Perspectives on an Emerging Discipline, Pierson Education, 1996.
- [5] Len Bass, Mark Klein, Felix Bachmann, " Quality Attribute Design Primitives and the Attribute Driven Design Method" , 4th International Wotkshop on Product Family Engineering Bilbao, Spain, 3-5 October 2001.
- [6] Mugurel T. Ionita, Dieter K. Hammer and Henk Obbink, " Scenario-Based Software Architecture Evaluation Methods: An Overview" , Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering, 2002.
- [7] Mario R. Barbacci, Mark H. Klein, Charles B. Weinstock "Principles for Evaluating the Quality Atributes of a Software Architecture" Technical Report CMU/SEI, 1997.