P10. *Quality Attribute Design Strategies*

2014

Sungwon Kang

10. 품질속성 설계전략

10.1 성능 설계 전략

10.2 변경용이성 설계 전략

10.3 품질속성 설계 전략의 정의 절차

10. 품질속성 설계전략

- **패턴(pattern)** : 반복적으로 발생하는 문제에 대한 미리 만들어놓은 솔루션으로 보통 여러 개의 힘의 충돌을 해결

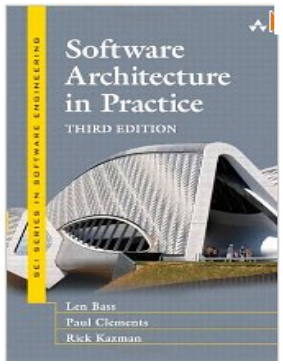
- 패턴이 문제를 해결해 주지 못하면 어떻게 할 것인가?
  - ☛ 품질속성 설계전략을 이용할 수 있다.
    - 품질속성 설계전략은 광범위하게 다룰 수 있는 해결의 틀을 제시한다.

- **품질속성 설계전략:** 단일 품질속성응답을 제어하는데 영향력 있는 설계결정

10. 품질속성 설계전략

[Bass 13] provided quality design tactics for the following quality attributes:

(1) availability
(2) interoperability
(3) modifiability
(4) performance
(5) security
(6) testability
(7) usability



10.1 성능 설계 전략

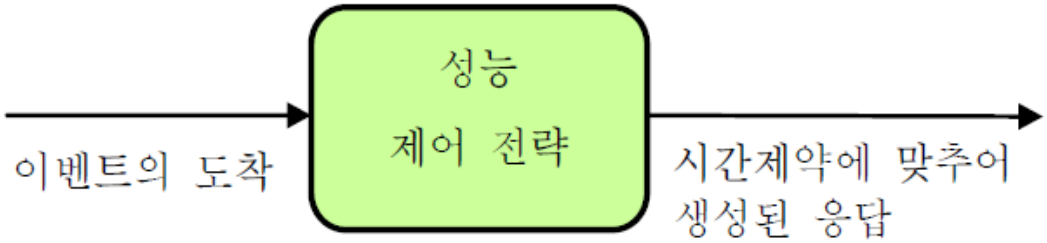


그림 10-1. 성능 설계 전략의 역할 (출처: [Bass 13]p. 136)

표 10-1. 성능 설계전략[112] (출처: [Bass 13]p. 141)

자원 수요의 제어	자원의 관리
– 샘플링 빈도 조절 – 이벤트 응답 속도 관리 – 이벤트의 우선순위 부여 – 처리 오버헤드(overhead) 감소 – 실행시간 상한선 설정 – 자원 효율성의 증대	– 가용 자원의 증대 – 병행성(concurrency) 도입 – 복수의 컴포넌트 복제본(copy) 유지 – 복수의 데이터 복제본 유지 – 큐 크기의 상한선 설정 – 자원의 스케줄링

10.2 변경용이성 설계 전략

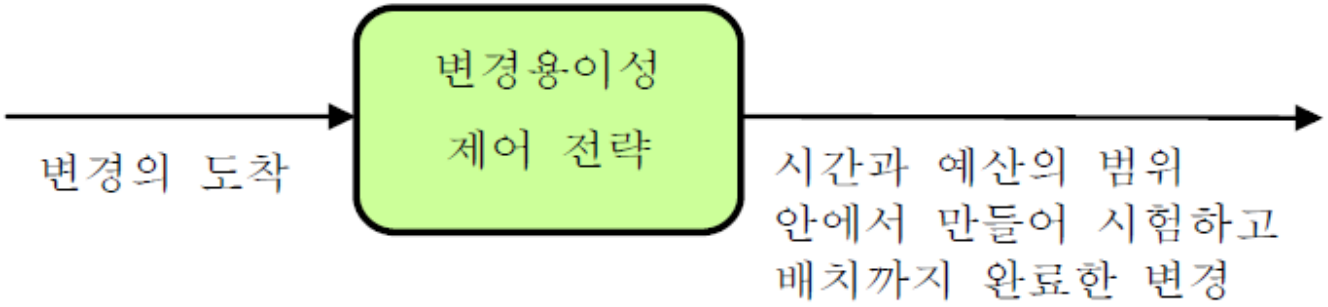


그림 10-2. 변경용이성 설계 전략의 역할 (출처: [Bass 13]p. 106)

표 10-2. 변경용이성 설계전략 (출처: [Bass 13]p. 122)

A) 모듈크기 축소	B) 응집도 증대	C) 결합도 축소	D) 바인딩의 지연
– 모듈의 분리	– 의미적 정합성(coherence)의 증대	– 캡슐화 – 중개자(intermediary)의 사용 – 의존관계의 제한 – 리팩토링 – 공통서비스의 추상화	

10.3 품질속성 설계 전략의 정의 절차

- 구체적인 품질속성 설계 전략의 정의 절차 [Bass 13]:
  1. 관련 품질속성을 위한 분석모델로 시작한다.
  2. 그 모델의 파라미터를 식별한다.
  3. 그 모델의 파라미터를 조정하기 위한 아키텍처적 기법을 식별한다.

  See Ch. 11 for an example

10.3 품질속성 설계 전략의 정의 절차

- **품질속성의 일반화 <–> 검증가능한 시나리오 ?**
  - 검증 가능한 QA 시나리오를 도출하는 이유:
    - 일반적인 품질속성의 요구가 불명확
    - 일반적인 품질속성의 세분화 분류를 통하여 요구를 규정짓는 것이 성공적이지 못하였기 때문
  - 이런 상황에서 품질속성에 대한 일반적인 설계전략을 제시하는 것이 타당한가?

☛ 품질속성의 설계전략은 일반적인 가이드라인을 주어,
특정시나리오들을 충족시키는 설계를 지원하고자 함

  - 품질속성설계전략이 설계를 지원할 수 있는지에 대한 궁극적인 판단은 설계자의 몫.

Lab 3. 아키텍처 설계

- **아키텍처 문서의 D와 E를 작성**
- "E. 아키텍처 설계 결과"는 "D.3 아키텍처 설계절차의 정의"에서 정의된 설계절차를 준수
- 뷰의 표현 언어는 적절히 만들어 사용
  - 프로젝트가 완료된 후 UML을 사용한 아키텍처 뷰의 기술 학습

아키텍처 설계 결정표

설계결정	해결조건/방법	적용점
ADS1 (클라이언트- 서버 아키텍처 스타일의 적용)	중앙의 데이터를 많은 사용자가 공유해야 한다.	E.1절의 개념뷰
ADS1 (load balancing)	J2EE는 서버의 수를 늘리고 서버들 사이에 부하를 균형 있게 만들 수 있는 메커니즘을 갖고 있다.	config7.xml 의 변수 _noservers … 등
. . .	. . .	. . .

Questions?