

로써 아키텍처의 복잡도[7]를 측정하는 방법을 제안했다. 이런 여러 가지 분석을 통한 아키텍처 검증방법이 제시되었지만, 이들 중 아키텍처의 다양한 면을 함께 고려하는 것은 많지 않았다.

2.2 요구사항 관리도구

소프트웨어의 대형화, 복잡화에 따라 요구사항의 중요성 또한 많이 인식되었지만, 아직까지 분석, 설계에 비하면 요구공학에 대한 인식은 낮은 편이다. 현재는 미국, 유럽 등지에서 요구사항 관리를 위해 많은 도구가 나와 요구사항의 수집과 관리 기능을 중심으로 많은 연구가 행해지고 있다[8].

현재 많이 사용중인 요구사항 관리도구들로는 DOORS, RTM, RequisitePro, SACHER, Core, PROTEUS 등이 있다. 이런 도구들은 PC 기반이거나 플랫폼에 종속적인 경우가 많기 때문에, CSCW(Computer Supported Cooperative Work)가 보편화되어가고 있는 실정에는 잘 맞지 않다. 최근에 DOORS, RTM, RequisitePro 등이 부분적으로 웹을 지원하고는 있지만, 아직은 보조적인 수준에 머무르고 있다.

3. Architecture Tradeoff Analysis Method(ATAM)

ATAM은 변경가능성, 성능, 안정성, 보안성 등 복수의 quality attributes를 함께 고려함으로써 아키텍처 수준의 디자인을 평가하는 방법이다. ATAM은 좀더 구체적인 상황을 제공할 수 있는 시나리오를 사용함으로써 여러 quality attributes 사이의 trade-off 포인트를 잡아내고, 이들간의 적당한 타협점을 찾아낼 수 있게 해준다. 즉, ATAM은 아키텍처의 다양한 면을 함께 고려함으로써 발생 가능한 위험을 초기에 제거하고, 아키텍처 자체를 더 잘 이해할 수 있게 해 준다. 다음은 ATAM을 실행하는 순서이다[9].

- | |
|--|
| 0. Planning/Information exchange
1. Scenario brainstorming
2. Architecture presentation
3. Scenario coverage checking
4. Scenario grouping and prioritization
5. Map high priority scenarios onto architecture
6. Perform quality attribute-specific analysis
7. Identify trade-off points
8. Consolidate findings and develop action plan |
|--|

그림 1. ATAM의 수행순서

4. Web-based Scenario Management System (WSMS)

WSMS는 웹기반의 시나리오 관리도구이다. ATAM이 시나리오를 이용한 아키텍처 분석 및 검증 방법임에도 불구하고, 시나리오를 관리하는 체계적인 방법을 가지고 있지 않다. 이를 보완하기 위해서 본 논문에서 ATAM에 최적화된 시나리오 관리도구인 WSMS를 제안한다.

4.1 기능

4.1.1 시나리오 추출

시간과 장소에 구애받지 않고, 웹을 통해서 사용자가 제안하는 시나리오를 받아들인다.

4.1.2 시나리오 분류

모든 시나리오들은 해당되는 quality attributes에 따라 분류될 수 있고, 또한 시나리오의 수가 많을 경우에 자연어 처리 엔진을 이용하여 관련된 시나리오들을 보다 정확히 분류할 수 있게 한다.

4.1.3 시나리오 넘버링과 우선순위 설정

새로운 시나리오가 들어오면 WSMS는 각각의 시나리오에게 유일한 식별 번호를 부여해서, 시나리오의 관리를 체계적으로 할 수 있게 한다. 또한, 각각의 시나리오에 우선순위를 부여할 수 있게 하여 보다 중요한 시나리오를 선택할 수 있게 해 준다.

4.1.4 시나리오 변경 관리

시나리오에 변경이 발생하는 경우에 WSMS는 새로운 시나리오를 데이터베이스에 추가하고, 이전 시나리오를 추적할 수 있는 정보를 유일한 번호와 함께 저장한다. 즉, 시나리오의 삭제나 변경이 해당 시나리오를 데이터베이스에서 완전히 제거하는 것을 의미하는 것이 아니고, 이전 시나리오들은 물론 변경의 이유와 이전 시나리오들로 추적할 수 있는 정보를 유지한다는 것이다.

4.2 시스템 아키텍처

WSMS는 크게 시나리오 추출, 시나리오 우선순위화, 시나리오 분류, 변경관리, 버전관리와 같은 다섯 가지의 컴포넌트로 구성된다. 이들은 다음의 그림 2에서 볼 수 있다.

a. 웹 환경에서의 시나리오 추출: WSMS는 시스템 요구사항을 충족시키기 위해 시나리오를 추출하고, 유일한 번호를 제공하며, 계층적인 입출력을 가능하게 한다.

b. 시나리오에 우선순위를 부과: 모든 시나리오를 만족시킬 수 있는 아키텍처는 구현하기 힘들기 때문에 가장 중

요한 시나리오를 골라내기 위해 우선순위를 부과한다.

c. 자연어 처리에 의한 시나리오의 분류: attribute에 의해서 뿐만이 아니라 자연어 처리를 통해서 관련된 시나리오를 분류해낸다.

d. 시나리오의 변경관리: 변경된 시나리오들에 관한 정보를 가지고 있어서, 변경 전의 시나리오, 변경 이유 등을 추적할 수 있다.

e. 시나리오의 버전관리: WSMS는 시나리오들의 버전을 관리할 수 있다.

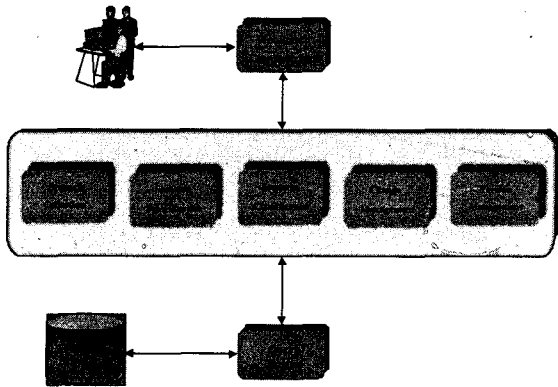


그림 2. WSMS의 시스템 아키텍처

4.3 구현

WSMS는 지난 연구의 결과물이었던 웹기반의 요구사항 관리도구를 기반으로 개발되고 있다. 본 도구는 자바 언어를 사용하고 있으며, UML을 사용하여 객체지향방법에 따라 개발되고 있다. 서버 부분은 Windows-NT에서 작동하고, MS-SQL 7.0을 DBMS로 사용하고 있다. 클라이언트 부분은 자바 애플릿을 다운받아서 실행되기 때문에 플랫폼에 상관없이, 인터넷에 연결되어 있다면 실행될 수 있다. 자바 보안상의 이유로 클라이언트인 애플릿이 직접 DBMS에 접근하는 것이 아닌, 클라이언트/서버 형태로 도구가 개발되고 있다.

5. 결론

본 논문에서는 시나리오 추출, 시나리오 우선순위화, 시나리오 분류, 변경관리, 버전관리 등의 기능을 수행하며 ATAM을 지원하는 WSMS에 대하여 설명하고 있다. 소프트웨어의 대형화, 복잡화는 아키텍처의 대형화, 복잡화란 결과를 만들어냈다. 따라서, 아키텍처를 분석, 검증하는 ATAM에서 아키텍처를 기술하는 시나리오들의 효과

적인 관리는 매우 중요하다 할 수 있다. 이 시점에서 웹을 통한 시나리오의 관리는 새로운 가능성으로 대두되고 있다. WSMS를 통해서 ATAM에서 시나리오의 효과적인 관리뿐만이 아니라 소프트웨어의 개발, 검증 단계에 까지 시나리오의 이용이 가능해 질 것이다.

향후에는 요구사항과 시나리오를 통합하여 관리하는 방안을 모색할 것이다. 또한, 체계적인 요구사항, 시나리오 관리를 통해 시스템 전체를 기술하는 시나리오를 개발하여, 아키텍처의 검증, 테스트에 이용할 수 있는 방법을 연구할 것이다. ATAM과 WSMS를 이용하여 에이전트의 아키텍처를 분석, 검증할 수 있는 연구를 또한 진행할 것이다.

6. 참고문헌

- [1] Len Bass, Paul Clements, Rick Kazman, Software Architecture In Practice, Addison Wesley, 1998
- [2] Kazman, R., Abowd, G., Bass, L., Clements, P., "Scenario-Based Analysis of Software Architecture", IEEE Software, Nov. 1996, 47-55
- [3] Rick Kazman, Mark Klein, Mario Barbacci, "The Architecture Tradeoff Analysis Method", Proceedings of ICECCS '98, (Monterey, CA), August 1998, 68-78
- [4] P. Inverardi and A. L. Wolf, "Formal Specification and Analysis of Software Architectures using the Chemical Abstract Machine Model," IEEE Transactions on Software Engineering, 21(4):373--386, April 1995
- [5] Nico Lassing, D. Rijsenbrij, and H. Vliet, "Towards a broader view on software architecture analysis of flexibility," Proc. the Asian-Pacific Software Engineering Conference (APSEC'99), 1999
- [6] J.A. Stafford, D.J. Richardson, and A. L. Wolf, "Aladdin: A Tool for Architecture-level Dependence Analysis of Software Systems," University of Colorado Technical Report, CU-CS-858-98, 1998
- [7] J. Zhao, "On Assessing the Complexity of Software Architectures," Proc. 3rd International Software Architecture Workshop (ISAW3), pp.163-166, ACM SIGSOFT, Orlando, Florida, USA, November 1998
- [8] 이원우, 황만수, 박수용, 류성열, "요구사항 관리를 위한 웹 기반 모델 설계", 소프트웨어공학회지 11권 4호, 1998.12
- [9] R. Kazman, M. Barbacci, M. Klein, S.J. Carriere, "Experience with Performing Architecture Tradeoff Analysis", Proceedings of ICSE99, (Los Angeles, CA), May 1999, 54-63