자동 코드 생성 기법을 이용한 원전 소프트웨어 개발 프로세스의 구조적 평가

A Structural Assessment of the Software Development Process for the Nuclear Power Plant using Auto-code Generation Technique

장훈선, 정재천, 김도연, 김재학, 남상구, 장영우
한국전력기술주식회사, 대전광역시 유성구 덕진동 150
hschang@kopec.co.kr

1. 서 론

산업자원부 전력산업 연구개발 과제인 "발전소 계측제어시스템용 소프트웨어 통합 검증 환경 구축"의 일환으로 그림 1 과 같은 컴포넌트 기반 설계 개념 및 환경을 시험적으로 구축하였다. 본 개념은 반복적인 소프트웨어 개발 수명주기의 개발에 따른 위험을 식별하고, 제거하는 것에 제 1 의 목표를 두고 있으며, 재사용을 고려한 설계 개념을 도입하고 있다. 이를 위해 인터페이스 단위의 조립을 통한 개발 환경을 구축하여 생산성 향상도 아울러 이룩할 수 있도록 하고 있다. [1]

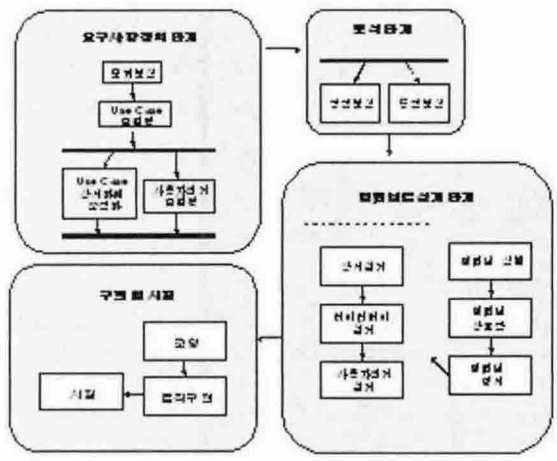

그림 1. 컴포넌트 기반 소프트웨어 설계 개념도

2. 컴포넌트 기반의 소프트웨어 설계개념

프로세스 진행은 요구사항의 정의, 분석 및 설계 단계와 구현 및 시험 단계를 거친다.

2.1 모델링 도구에 의한 소프트웨어 구현[2]

요구 사항의 정의 단계에서는 요건 분석을 통해 개발할 시스템 범위를 정하고, 시스템에 대한 사용자의 요구사항을 체계적으로 정의하고, 시스템 범위를 결정한다. 요건분석을 통해 식별된 시스템 요건은 유즈케이스 다이어그램을 이용한 모델링을 이용하여 사용자 입장에서 시스템의 요구 사항을 알기 쉽도록 하고 컴포넌트의 식별이 가능토록 한다.

분석 단계에서는 중요한 기능을 정립하기 위하여 클래스 다이어그램을 이용하여 정적 분석을 먼저 수행한 후 UML 의 상관 연계 모델 기법인 액티비티 다이어그램과 시컨스 다이어그램을 이용 하여 동적 분석을 수행토록 한다. 정적 분석시 클래스 식별기법으로는 MVC (Model View Control) 을 사용한다.[3]

설계단계에서는 분석된 모델을 가지고 컴포넌트 식별을 한 후, 초기 컴포넌트 아키텍처를 모델링 하도록한다. 컴포넌트의 상호작용 모델링을 통해서 초기 컴포넌트 아키텍처를 검증 및 보완한 후 완성된 컴포넌트 아키텍처를 기반으로 컴포넌트 명세 작업을 한다. 식별된 컴포넌트들은 상세 설계를 통해 구체적인 설계 요소 (클래스 등)를 식별하고 식별된 설계요소를 이용하여 상세 동적 설계 작업을 하게 된다.

구현단계에서는 동적 설계단계에서 식별된 각 컴포넌트에 대해 시컨스 다이어 그램으로부터 소스 코드가 자동 생성된 후 프로그래머가 최종 조정하는 절차를 수립한다. 최종 소스 코드는 Visual C++언어로 구현된다.

2.2 컴포넌트 기반의 소프트웨어 품질평가

코드 품질관리 기능을 이용하여 개발된 시스템 코드가 사용자 정의형 스타일과 표준 코딩 지침등에 적합한지 여부를 검사하고, 시스템의 정량적 평가 데이터를 측정하였다. 매트릭 요소(예 CBO : 객체간의 결합도) 결과는 Kiviat 그래프 등을 이용하여 출력하였다.[4]

원자력발전소 소프트웨어 통합 개발 프로세스에 대한 적용성 여부를 평가하기 위해 가압 중수로형 원전의 제 2 안전정지 계통을 축약하여 본 논문에서 제시된 컴포넌트 기반 설계 기법을 이용, 요건분석 단계로부터 설계, 구현 및 시험 단계까지를 구현하였다. 식별된 컴포넌트 중 중성자속 신호측정 컴포넌트의 NEUTRONICS 클래스 LogPower 부분 구현을, 시컨스 다이어그램 으로부터 자동 코드 생성 결과를 개발 경력이 많은 프로그래머가 작성한 코딩과 시컨스 다이어그램을 숙지한 경력이 적은 프로그래머가 코딩한 결과를 비교하였다.

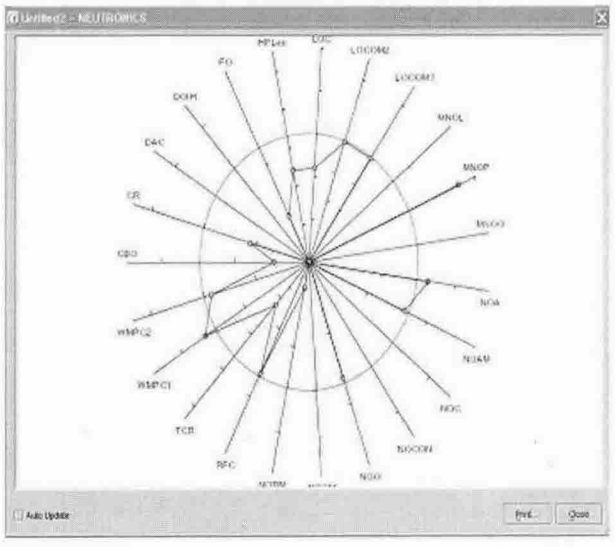

그림 2. 구현된 코드에 대한 품질 관리결과 (KIVIAT 그래프)

시컨스 다이어그램을 통한 모델 데이터를 소스 자동 생성을 통해 구현한 부분과 시컨스 다이어그램 모델 데이터를 근간으로 수동으로 구현한 부분을, 코드 품질관리 도구를 이용하여 그림 2 와 같이 실행한 결과, 매트릭 요소 결과에서 코드의 전체 라인수만 차이가 있을 뿐 다른 부분은 일치하는 결과를 얻을 수 있었다.

표 1 에서 보는 바와 같이 설계 단계에서 모델링한 시컨스 다이어그램을 이용하여 구현된 소스 생성 결과는 프로그래머의 알고리즘 이해도에 따라 차이가 있다.

표 1. 자동 코드 생성 기법 사용의 경우와 매뉴얼 코딩시 비교

코딩 방법	코드 자동생성 (검증단계 후처리 코딩)	시컨스 다이어그램 수작업
라인수	14	23
함수생성	자동생성	수동
관련변수	자동생성	수동

3. 결론

분석결과, 품질감사 요소는 프로그래머의 개발 능력에 따라 값이 달라지고 상세 설계의 동적 모델을 가지고 구현을 했다면 개발자의 개발 성향에 따른 프로그램 볼륨 부분만 차이가 있을 뿐 매트릭 요소 결과 값들은 거의 같다는 결론을 도출하였다. 이에 따라 코드 자동생성에 의한 정형적 코딩 방식을 사용하는 것이 원전용 소프트웨어 개발에 좀더 바람직한 방법론일 것으로 사료된다.

4. 참고문헌

[1] CRIEPI, "Software Fault Reduction Using Comp uter-Aided Software Engineering (CASE) Tools", EPRI TR-105989-V1, Jun. 1995.
[2] Ivar Jacobson et al., "The Unified Modeling Lan guage User Guide", Addison Wesley, 1999.
[3] TogetherSoft, "Together Quick-start for UML W orkbook", 2002
[4] Jeff Tian, "Quality-Evaluation Models and Measurements", IEEE Software, May 2004.