원전 분야를 위한 KAIST 소프트웨어 개발 방법론

조재명, 오윤주, 유준범, 차성덕
한국과학기술원 전자전산학과 전산학전공
{jmcho, yjoh, jbyoo, cha}@salmosa.kaist.ac.kr

KAIST Software Development Framework for Nuclear-Domain (KSDFN)

Jaemyung Cho, Younju Oh, Junbeom You, and Sungdeok Cha
Dept. of EECS, Div. Of CS, Korea Advanced Institute of Science and Technology

요 약

원자력 발전소 제어시스템은 그 특성으로 인하여 안정성이 크게 중요시되며, 소프트웨어의 오류가 많은 인적 물질적 피해를 줄 수 있는 시스템이므로 높은 신뢰도를 가지고 있는 소프트웨어가 요구된다. 높은 신뢰도를 뒷받침하기 위해 내장되는 소프트웨어는 적합한 정형명세기법과 정형검증기법의 적용이 필요하다. 소프트웨어 개발시 적용되는 정형화 기법은 명세의 모호성을 제거하고 검증을 수행하기에 용이하게 하므로, 높은 신뢰도를 요구하는 원자력 분야의 소프트웨어 개발에 유용하다. 그러므로, 본 논문에서는 원자력 분야의 소프트웨어 개발시 유용하게 적용할 수 있는 정형기법을 기반으로 한 소프트웨어 개발 방법론을 제시하고자 한다.

1. 서 론

본 논문에서 대상으로 하고있는 시스템은 원자로 보호계통(Reactor Protect System)의 계측제어 시스템 소프트웨어로서 시스템의 오류 발생시 큰 인적 물질적 피해를 초래할 수 있기 때문에 시스템을 구성하는 복잡한 소프트웨어에 대한 안전성, 신뢰성, 확실성, 무결성 등이 이들 소프트웨어 개발시 반드시 고려되어야 할 사항이다. 이러한 시스템을 Safety-Critical System[1]이라고 하며, 항공기 관제시스템, 인공위성 제어 시스템, 원자력 발전소 통제시스템 등이 그 좋은 예이다.

일반적으로 소프트웨어는 요구사항 분석, 설계, 구현, 테스팅, 유지보수 등의 주기에 따라 개발된다. 이 중에서 요구사항 분석단계는 소프트웨어의 품질과 안전성에 가장 중요한 역할을 하는 것으로 알려져 있다. 자연어로 되어있는 요구사항 명세는 많은 모호성을 갖고 있으므로 개발자들 사이에 혼란을 야기할 수 있고 검증을 수행하기 힘들 다는 단점이 있다. 많은 사고들이 소프트웨어의 요구사항 오류에 기인하고 있으며, 요구사항 단계에서 생기는 오류는 구현 등의 단계에서 생기는 오류보다 매우 큰 수정 비용을 필요로 한다.

위와 같은 문제점의 해결을 위해 요구사항에 대한 엄격한 명세와 검증을 위해 명세의 모호성을 제거하고, 검증을 수행하기에 용이하게 할 수 있는 정형기법(Formal method)을 Safety-Critical System인 원자력 분야에서의 소프트웨어 개발시 적용한다면 이상적일 것이다.

정형기법은 정형적인 의미(Formal semantics)를 가지는 정형 명세언어(Formal specification language)와 그 언어를 기반으로 시스템의 성질을 검증할 수 있는 정형 검증방법(Formal verification method)을 제공한다.

현재 원자력 분야에서는 기술 국산화와 함께 소프트웨어의 신뢰도를 높이려는 많은 노력이 진행되고 있다. 그러나 현재까지 신뢰성 있는 시스템을 구현하기 위하여 부분적으로 정형기법을 이용한 사례는 있었으나, 소프트웨어 개발 전 단계에 걸쳐 적용한 사례는 없었다. 이에 소프트웨어의 신뢰성 향상을 위하여 소프트웨어의 개발시 전 단계에 걸쳐 정형기법을 이용하는 개발방법론을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 원자력 분야에서의 소프트웨어 개발에 있어서 기존에 적용되었던 기법에 대해 설명하고, 3장에서는 정형기법을 이용한 원자력 분야에서의 소프트웨어 개발 방법론을 제시하고, 이를 효과적으로 지원 할 수 있는 도구를 제안하였으며, 4장에서는 결론 및 향후 연구 방향을 제시하고 있다.

2. 관련연구

2.1. 원자력 분야의 소프트웨어 개발절차

원자력 분야의 계측제어 시스템의 소프트웨어는 내장시스템(Embedded System)이며, 소프트웨어 개발절차는 그림1과 같이 진행된다.

최초 소프트웨어의 요구사항 중 가장 기본적인 요소를 기술한 FRS(Functional Requirement Specification)를 작성하고, FRS를 바탕으로 요구사항을 보다 구체적으로 기술한 소프트웨어 요구사항 명세서인 SRS(Software Requirement Specification)을 작성한다. 작성된 SRS를 바탕으로 구현단계를 고려하여, 소프트웨어를 어떻게 만들 것인가를 기술한 소프트웨어 설계서인 SDS(Software Design Specification)를 작성한다.

작성된 소프트웨어 요구사항과 관련된 문서들을 이용하여 하드웨어에 내장되는 소프트웨어를 구현하고

테스팅을 통해 오류를 발견하는 노력을 수행하게 되며, 완성된 내장시스템은 유지보수를 실시하게 된다.

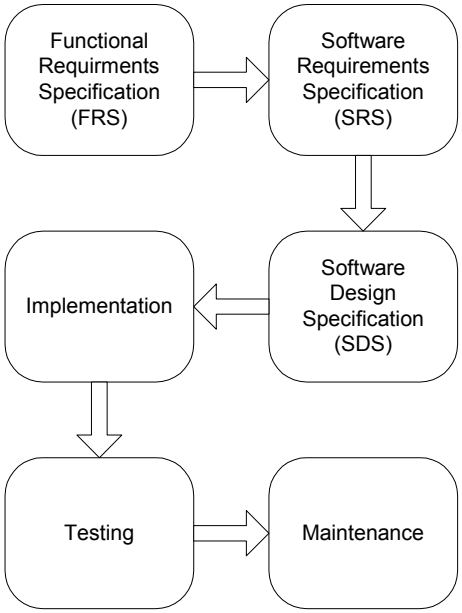

그림1 원자력 분야의 소프트웨어 개발절차

하지만, 위의 개발절차에서는 두 가지 문제점을 가지고 있다. 첫째, 요구사항을 문서화하여 나타낸 FRS, SRS, SDS 모두 자연어로 기술되어 명세의 모호성을 내포하고 있기 때문에, 요구사항이 맞는지, 일치 하는지, 구현시 제대로 반영이 되었는지 검증을 수행하기가 어렵다.

둘째, 요구사항이 명확히 반영되어 있지 않거나, 검증이 제대로 되어있지 않은 상태에서 테스팅 만을 거쳐 시스템을 완성하였을 경우, 테스팅은 모든 가능한 상황을 검사할 수 없으므로 시스템의 오류발생 확률이 높아질 것이다. 이러한 이유로 유지보수시 소요되는 노력이 매우 커질 것이며, 특히 Safety-Critical System에서는 큰 문제점으로 작용할 것이다.

2.2. 정형기법 적용사례

월성 2.3.4호기에 설치되어 운용되고 있는 비상정지 시스템의 경우 정형기법에 의해 명세 및 검증을 수행하려는 노력이 있었으며, 이를 AECL(The Atomic Energy of Cnanda Limited) 방법론이라고 한다. [2]

AECL 방법론은 자연어로 기술되어있는 FRS를 정형명세 언어인 SCR(Software Cost Reduction)을 기반으로 한 표기법을 이용하여 정형화된 언어로 나타내었다. 그러나, 요구사항 명세 단계만 정형기법을 적용하였으며, V&V절차는 정형기법을 적용하지 못하였다.[3]

3. 정형기법을 이용한 원자력 분야의 소프트웨어개발 방법론 KSDFN

3.1 개요

소프트웨어를 개발함에 있어 여러 가지 방법론이 제시되었고, 이를 적절히 선택하여 개발시 활용하는 것이 중요하다. 그러나, 특히 안정성이 중요시되고 높은 신뢰도가 요구되는 시스템의 특성상 정형기법에 기반을 둔 본 연구에서는 KSDFN (KAIST Software Development Framework for Nuclear-domain)이라는 방법론을 제안한다.

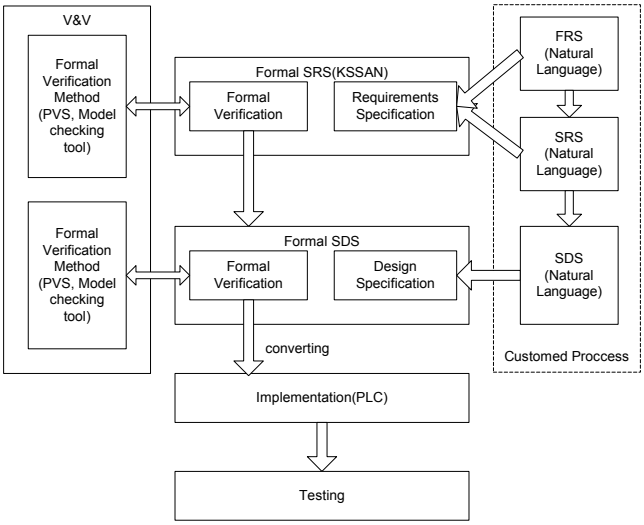

그림 2 KSDFN의 절차

그림 2는 KSDFN의 절차이다. 자연어로 작성되어 있는 SRS를 정형화된 언어로 명세하고, 정형검증 방법을 이용하여 검증을 수행한다. 검증된 결과는 차후 정형화된 SDS를 만드는데 유용한 정보를 제공하여 SRS와 SDS의 일치성을 제고한다.

또한, 자연어로 작성되어 있는 SDS를 정형화된 언어로 명세하고, 정형검증 방법을 이용하여 검증을 수행한다. 검증된 결과는 구현시에 사용될 PLC(Program Logic Controller)의 언어로 전환하는데 유용하게 사용될 수 있다. 정형기법이 사용되어 구현된 시스템은 최종적으로 테스팅을 거쳐 완성된다.

정형 검증시에는 정형화된 언어로 요구사항이 명세 되어 있으므로 정형 검증 방법인 정리증명(Theorem Proving)방법의 도구인 PVS (Prototype Verification System), 모델 체킹(Model Checking) 방법의 도구인 SMV(Symbolic Model Checker), SPIN 등의 도구를 이용해 검증할 수 있다. 또한 필요에 의해서는 자체 검증기를 구현하여 검증할 수 있다.

KSDFN은 요구사항 명세단계부터 구현 단계까지 정형화 기법이 적용되며, 다음 단계로의 전환시 정형 검증단계를 거치므로 일치성, 명확성, 완전성이 보장될 수 있다. 현재까지의 연구는 SRS의 정형화 명세와 PVS를 이용한 정형 검증단계 까지 수행되고 있으며, SDS의 정형화 명세는 진행중이다.

3.2 SRS 정형화

SRS의 정형화를 위해서는 정형화 명세를 위한 접

근방법으로서 제안된 KSSAN(KAIST Software Specification Approach for Nuclear Domain)을 사용한다. KSSAN은 기존의 AECL 방법론에 기초한다. ACEL 방법론은 시스템의 기능을 명세하기 위해서 DFD의 일종인 FOD(Function Overview Diagrams)와 condition-event 테이블인 SDT(Structured Decision Table)을 사용하며, 모든 시스템의 행위를 함수로서 표현하므로, 예전의 상태들이나 타이밍 부분은 명세에 어려움이 있다.

KSSAN은 시스템의 행위 중 기능과 관련된 부분만 함수로서 명세하고, 명세를 보다 명확하게 하기 위하여 예전의 상태들과 타이밍으로 명세해야 하는 부분은 오토마타를 이용하여 표현하는 방법론이다.

3.3 KSSAN 지원 도구 (KSSAN 에디터)

정형화된 요구사항을 위한 방법론인 KSSAN을 지원하기위해 지원 도구를 구현하였다. 현재 자연어로 기술된 요구사항 명세를 정형화하여 기술할 수 있도록 GUI환경의 에디팅 기능을 제공하므로 수정과 편집을 용이하게 할 수 있다. KSSAN 에디터는 그림3과 같은 인터페이스로 되어있다.

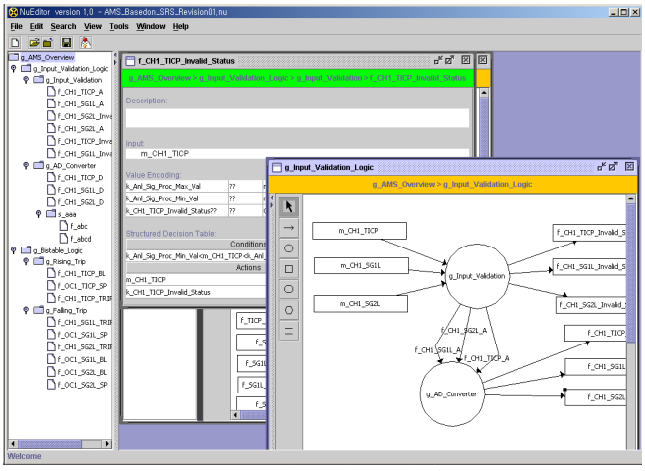

그림3 KSSAN 에디터 인터페이스

또한, 정형화된 명세를 정리증명기인 PVS를 이용하여 검증할 수 있도록 자료호환의 용이성을 제공한다. 함수는 원, 입출력은 직사각형, 히스토리는 육각형, 각 노드의 관계는 화살표로 연결하여 작성한다.

그림4는 KSSAN에디터를 사용하는 절차를 나타내고 있다. 자연어로 작성된 명세는 에디팅 기능을 이용하여 사용자가 직접 모델링을 하면서 입력 하며, 입력된 명세는 사용자의 수정 및 확인을 거친 후, 검증도구를 이용해 검증이 수행된다.

KSSAN 에디터의 입출력 자료는 XML형식으로 되어 있으므로 검증 및 다음 단계의 절차 수행시 유용하게 활용될 수 있다. 현재 KSSAN 에디터로 작성된 정형명세는 PVS로 검증을 수행할 수 있으나,[4] 모델체킹 도구인 SMV 또는 SPIN을 이용하거나 자체 검증기를 내장하여 검증하는 방안은 계획중에 있다.

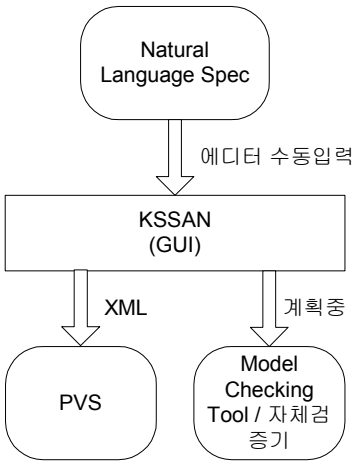

그림4 KSSAN 에디터 사용절차

4. 결론 및 향후 연구방향

본 논문에서는 정형기법을 이용한 원자력 분야에서의 소프트웨어 개발 방법론을 제시하였으며, 요구사항 명세의 정형화 방법과 이를 지원하는 도구를 제안하였다.

향후 과제로서는 정형화되어 있는 명세를 이용하여 V&V를 효과적으로 수행하기 위해 모델 체킹 도구를 사용하는 방안과, 자체 내장된 검증도구를 개발하는 방안에 대한 연구가 필요하다.

또한, 본 과제의 시스템은 정형기법을 이용한 V&V를 거친후 최종적으로 PLC로 구현된다. PLC는 내장시스템 개발시 사용되는 도구로서 소프트웨어 구현을 위한 다양한 언어를 제공한다. 주로 사용되는 언어는 시각적인 언어로서 FBD(Function Block Diagram), LD(Ladder Logic) 등이 있는데 정형화된 SDS에서 이를 부분적으로 지원해 줄 수 있는 기능을 지원해 주어야 할 것이다.

참고문헌

[1] N.G. Leveson, “Safeware – System Safety and Computers,” Addison-Wesley, 1995.

[2] WolsongnNPP 2/3/4, “Software Work Practice Procedure for the Specification of SR for Safety Critical Systems,” *Design Document no. 00-68000-SWP-002, Rev. 0*, Sept. 1991.

[3] WolsongnNPP 2/3/4, “Software Verification Report Code Verification, SDS2 PDC,” *Design Document no. 86-68350-SVR-002, Rev. 0*, June 1994.

[4] T. Kim, D. Stringer-Calvert, and S. Cha, “Formal Verification of Functional Properties of an SCR-Style Software Requirements Specification Using PVS,” *TACAS 2002, LNCS 2280*, pp. 205-220, 2002.