연동소프트웨어의 안정성 확보를 위한 시뮬레이션 기법
Simulation Technique for Secure Inter-locking Software

황 종 규*, 이 종 우*, 오 석 문*, 김 영 훈*
Jong-Gyu Hwang, Jong-Woo Lee, Suk-Moon Oh, Young-Hun Kim

ABSTRACT

Recently, the μ-processor based-controlled systems instead of conventional relays circuitry are widely used to industrial applications, and also those technology is available to railway signalings which are safety-critical systems. However, the safety and reliability of software for those systems are harder to demonstrate than in traditional relays circuitry because the faults or errors can not be analyzed and predicted to those systems. So, the safety problems are crucial more and more in μ-processor based-controlled system. In this paper, the Grafcet language, the graphical and mathematical form, is used to obtain the high-level safety and reliability of software control logic. The general description for Grafcet notation are provided. And some partial of interlocking logic are formally modeled and simulated by Grafcet language and graphical windows.

1. 서론

최근들어 마이크로프로세서를 이용한 제어시스템의 설계 및 제작기술이 전자기술의 발달에 힘입어 산업계 각 분야에서 실용화되어지고 있으며, 높은 레벨의 안정성·신뢰성이 요구되는 철도분야에서도 기존의 계전기를 이용한 논리회로 구성에서 마이크로프로세서를 이용한 제어시스템으로 대체되어가고 있다.

기존의 계전기 등은 소자 자체의 Fail-safe 동작 특성을 이용하여 높은 레벨의 안전성과 신뢰성을 확보할 수 있었지만 마이크로프로세서를 이용하여 제어시스템을 구성할 경우 기존의 계전기를 사용할 경우만큼 높은 레벨을 확보하기가 어렵다. 이 높은 레벨의 요구사항을 만족할 수 있도록 제어시스템을 이중 또는 삼중 시스템으로 구성하는 등 하드웨어에 의한 안전성 확보방안들이 다양하게 제안 및 적용되고 있지만, 상당부분 소프트웨어에도 그 역할을 요구하고 있다.

이에 따라 고장이나 오류의 발생을 최대한으로 줄일 수 있도록 소프트웨어가 개발되어져야 하며, 이를 위해서는 제어시스템 소프트웨어의 동작특성에 대한 사양을 엄격하게 정하여 로직상의 오류를 최대한으로 줄일 수 있도록 하여야 한다. 이 부분의 신뢰성 향상이 제어시스템 전체 소프트웨어의 신뢰성을 확보하는데 중요한 부분이 된다. 이러한 노력의 하나로 철도 선진국에서는 높은 안정성을 요구하는 제어시스템의 소프트웨어 개발에 정규기법(FM : Formal Method)을 적용

* 한국철도기술연구원, 정회원

하려는 연구가 진행 중에 있다[7][8]. 이는 소프트웨어 로직 및 그 동작특성을 엄격한 논리체계를 바탕으로 정의 및 표현하고 이를 검증함으로서 소프트웨어의 높은 신뢰성을 확보하는 것이다. 하지만 이러한 방법은 많은 개발기간을 필요로 하고 개발비용이 과다해지며 대규모의 전체시스템에 적용하기에는 다소 무리가 따른다. 특히 철도신호분야에도 철도 선진국에서는 연구가 진행 중에 있으나 아직은 연구차원에 머무르고 있다[1][2][9]-[12].

본 논문에서는 이에 따라 안전성과 신뢰성이 절대적으로 요구되는 분야의 소프트웨어 오류를 최대한 줄이기 위해서 표준화된 제어언어인 GRAFCET(*GRA*phe *F*onctionnel de *C*ommande *E*tape/*T*ransition)을 사용하여 제어로직을 모델링하고 시뮬레이션을 수행하여 소프트웨어의 신뢰성을 확보할 수 있는 방법에 대해 검토하고자 한다[4]-[6]. 특히 철도 신호제어시스템 중 열차의 안전운행을 최종적으로 책임지는 전자연동장치의 연동로직을 그 대상으로 하였다.

2. 소프트웨어의 안정성 확보를 위한 방안 검토

철도신호제어분야에서 기존의 계전기 로직에서 마이크로프로세서로 대체되어가면서 부피, 가격, 확장성 등 시스템의 많은 부분에서 장점이 있지만, 상대적으로 로직에 대한 안정성 및 신뢰성에 대한 부분은 더욱 중요하게 되었다. 이는 제어시스템이 전자화되어감에 따라 고장의 분석 등에 어려움이 있기 때문이다. 이에 따라 안정성이 절대적으로 요구되는 제어시스템에서는 하드웨어를 이중 또는 삼중으로 구성하는 Fault-tolerant 시스템으로 구성하는 등 다양한 방법들이 적용되어져 오고 있고, 소프트웨어에서도 Recovery Block 기법, N-version 프로그래밍 등 많은 결함허용 기법들이 적용되어지고 있다.

하지만 이러한 결함허용 기법들과 병행하여 중요하게 다루어야 할 것은 고장이나 오류의 발생을 최대한 줄일 수 있는 높은 레벨의 안정성과 신뢰성이 확보된 소프트웨어 로직을 확보하는 것이다. 이러한 절대안정성을 요구하는 제어시스템을 위해서 철도선진국에서는 정규기법의 적용에 대한 연구가 진행중에 있다. 이 정규기법은 모든 로직을 확실하고 엄격한 동작특성이 보장되는 수학적인 형태로 표현하는 것으로서, 높은 레벨의 안정성이 요구되는 부분의 동작사양과 기능을 이러한 정규기법에 의해 모델링하고 분석함으로서 로직의 오류확률을 줄여주고자 하는 것이다[7][8].

이러한 정규기법들에는 Z-notation, VDM, LOTOS 등 매우 많은 방법들이 개발되어지고 있고, RTRI, AEA Technology 등 철도선진국에서 신호제어시스템 분야에 적용하려는 연구가 진행 중에 있다[1][2][9]-[12]. 정규기법의 적용에 대한 연구와는 동시에 이를 검증하고 확인할 수 있는 방법들에 대한 연구도 동시에 필요로 한다. 이 검증을 위한 방법의 하나로 제어시스템에 많이 사용되고 있는 Petri-net이 있다. 이 페트리 네트는 제어의 흐름과 상태의 변화를 모델링하는데에 많이 이용되고 있으며, 이러한 모델링을 그래픽 형태와 수학적인 형태로 정규화시켜 표현할 수 있어 신호제어시스템의 시뮬레이션 및 검증을 위한 방법으로 이용되어져 오고 있다[1][3].

또한 80년대 후반 프랑스에서 이 페트리 네트의 기능을 그대로 가지면서 좀더 쉽게 표현할 수 있는 Grafcet이 소개되었다[4]-[6]. 이 Grafcet은 IEC1131-3(International Electrotechnical Commission) 표준으로 채택되었다. 다음 절에서는 이 Grafcet의 기본적인 표현방법을 설명하고 이것을 어떻게 전자연동장치의 연동로직의 모델링 및 시뮬레이션에 적용할 지를 검토하고자 한다. 이러한 표준화된 언어를 바탕으로 시뮬레이션을 수행함으로서 좀더 구체적이고 엄격한 동작사양을 얻을 수 있을 것으로 기대된다.

3. Grafcet의 기본표현

Grafcet은 Petri-net에서 파생되어진 모델링 도구로서 몇가지 규칙만을 이용하여 시스템 모델링을 할 수 있으며 이미 프랑스를 비롯한 유럽 등의 여러 국가에서 실제의 복잡한 제어시스템 분야에 많이 이용되어지고 있다. 이 Grafcet은 제어시스템의 상태변화, 제어의 흐름을 그래픽적으로 모델링 할 수 있고, 또한 제어의 흐름을 나타내기 쉽게 되어있어 복잡한 제어시스템의 분석 및 모델링에 적합하다. 이 Grafcet은 1970년대 프랑스에서 처음 소개되었으며 1987년 IEC에 의해 표준언어로 채택되었다.

Grafcet은 일련의 제어동작을 여러 개의 스텝으로 분할하여 프로그램 실행순서와 실행조건을 명확히 표현 가능하도록 한 제어로직의 동작사양을 모델링하고 표현할 수 있는 언어로서 스텝(Step), 천이(Transition) 링크(Link), 제어액션(Control Action) 등으로 구성되어지며 그 기본구성이 그림1에 나타나 있고, 그래프의 표현방법은 페트리 네트와 거의 유사하다.

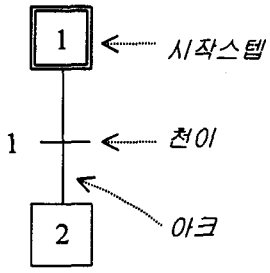


그림 1. Grafcet의 기본구조

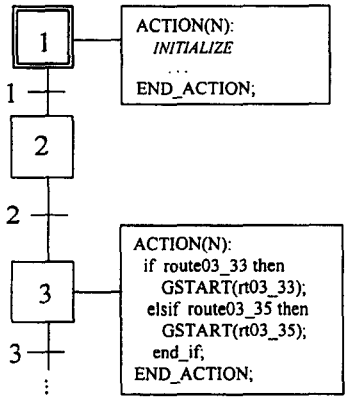
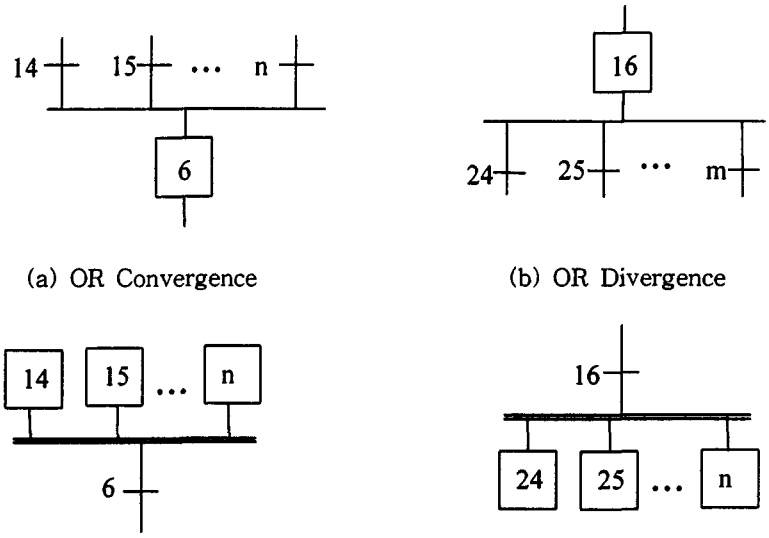


그림 2. 제어액션

스텝은 활성화 상태와 비활성화 상태를 가지며 제어토큰이 표시된 스텝이 활성화 스텝이 된다. 스텝들 중 프로그램이 시작될 때 맨 처음 시작하는 스텝을 초기스텝이라하며 이중 사각형으로 표시한다. 제어액션은 스텝에 대응하는 것으로 스텝이 활성화되면 수행되어지는 부분으로 각 스텝 옆에 사각형 내에 수행할 내용들을 표시하게 된다. 이 제어액션은 IEC에서 표준으로 정한 ST, IL, LD 등의 어느 언어로도 표현이 가능하며, 모든 스텝들이 제어액션을 가지는 것은 아니다. 천이는 스텝들을 활성화 또는 비활성화 시키도록 하는 것으로 이 천이의 점화조건을 만족하면 바로 전 스텝을 비활성화 시키고 다음 스텝을 활성화시키도록 하는 역할을 한다. 즉, 이 조건에 따라 제어토큰이 한 스텝에서 다른 스텝으로 이동하게 된다. 이 천이의 점화조건은 논리변수나 논리 연산자, 타이머 등의 여러 가지 형태로 표현이 가능하다. 제어로직의 동작사양 표현시 이 천이의 점화조건을 어떻게 정하느냐 하는 것이 프로그램의 제어흐름을 결정하는 중요한 부분이 된다.

또한 Grafcet은 좀더 효율적으로 로직의 제어흐름을 표현할 수 있도록 결합OR(OR Convergence)와 분산OR(OR Divergence) 그리고 결합AND(AND Convergence) 와 분산AND(AND Divergence)에 의해 선택 시컨스와 병렬 시컨스가 가능하도록 하고 있다. 결합OR는 여러 개의 천이 조건들 중 하나이상이 만족하면 다음 스텝이 활성화되는 것이고, 분산 OR는 스텝의 출력이 여러 개의 천이조건으로 분기하는 것으로 여러 개의 천이조건들 중 하나를 만족하면 그 스

텝으로 분기하게 된다. 이 결합 및 분산 OR는 여러 가지로 분기하고자하는 선택적인 로직 프로세스에 효과적으로 이용될 수 있다. 마찬가지로 결합AND는 모든 스텝이 활성화되어야 만이 제어토큰이 다음 천이로 넘어갈 수 있으며 천이가 점화되면 모든 입력 스텝들이 동시에 비활성화 된다. 그림3 (d)에서 분산AND는 천이16이 점화되면 동시에 다음 스텝 24, 25, ..., n 이 활성화되어 진다. 이러한 형태의 선택 및 병렬 시컨스는 제어흐름을 표현하는데 매우 유용하게 이용될 수 있을 것이다.



(a) OR Convergence (b) OR Divergence

(C) AND Convergence (d) AND Divergence

그림 3. 선택 시컨스 및 병렬 시컨스

그림4와 같은 매크로스텝(Macro Step)은 복잡한 제어로직을 계층적인 간략화된 그래프로 표현할 수 있도록 해 준다. 즉 복잡한 그래프를 매크로 스텝으로 표시하면 전체 제어흐름을 이해하기에 편리할 것이다. 그림 5는 Grafcet 그래프의 전개와 타이밍 차트를 동시에 비교한 하나의 예로서, 스텝 16, 17, .., n이 활성화되어 있다가 천이조건 'exp'가 점화되면서 제어토큰이 같은 분산 AND에 연결된 모든 스텝들이 동시에 비활성화되는 것을 나타낸다.

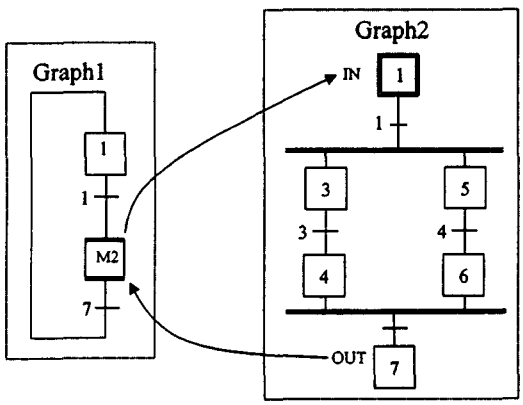


그림 4. 매크로스텝의 표현

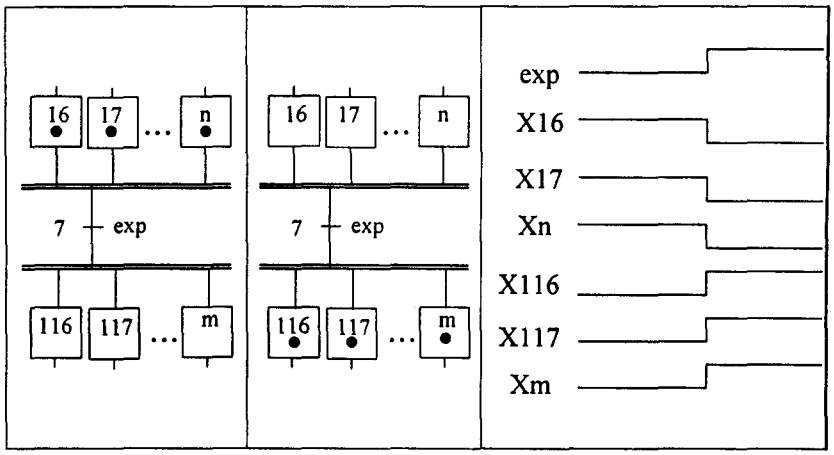


그림 5. Grafcet의 전개와 타이밍 차트

시스템의 제어 프로세스에 Grafcet을 적용할려면 다음과 같이 제어로직의 기능과 동작특성을 Grafcet으로 표현하고 그 로직을 검증하는 1단계와 이를 실제 I/O 하드웨어와 연계시켜 동작시키는 2단계로 구분될 수 있다.

- 1 단계(Functional Level)
  - Step 1 : 시스템에 대한 전체적인 파악을 하며 제어목적, 제어동작의 특성을 파악
  - Step 2 : 시스템의 입·출력에 대한 정의
  - Step 3 : 각 스텝에서 각 이벤트들에 대한 동작을 제어액션에 프로그래밍
  - Step 4 : 천이의 점화조건들을 표시
  - Step 5 : 제어흐름을 표현하고 로직의 검증 및 수정
- 2 단계(Operational Level)

4. 연동로직 분석을 위한 시뮬레이션

전자연동장치는 열차집중제어장치나 지역제어조작반(LCP : Local Control Panel)으로 부터의 제어명령을 수신하여 선로변에 진로제어 신호를 발생시키고 또한 안전한 진로의 확보를 위한 각종 쇄정로직을 처리하는 열차의 안전운행을 마지막으로 책임지는 장치이다. 이러한 높은 안정성과 신뢰성이 요구되는 연동로직의 확보를 위해 앞장에서 설명한 Grafcet을 이용하여 로직의 모델링 및 시뮬레이션을 수행하였다.

전자연동장치의 주요 기능은 진로제어, 감시기능, 보호기능 등으로 분류될 수 있으나 연동장치 고유기능은 안전한 진로제어기능이다. 이 진로제어에는 진로설정, 진로해정, 전철기 제어, 신호기 제어 등으로 다시 분류될 수 있는데 이러한 기능들을 수행함에 있어서 안전한 진로의 확보 및 보호가 되도록 접근쇄정, 수렴쇄정, 대향진로쇄정, 진로쇄정, 전철기 쇄정, 철사쇄정 등의 각종 안전로직들을 필요로 한다. 우선적으로 진로설정 프로세스를 살펴보면 열차집중제어장치나 LCP로부터 진로제어요청을 하게되면 요청된 진로가 제어가능한지를 우선적으로 검사하게 된다. 이를 진로의 가용성 검사라고 하며 이 가용성 검사가 완료되면 진로상의 모든 전철기들의 위치를 확인하고 원

하는 위치를 제어하게 된다. 그리고 나서 설정된 진로의 보호를 위해서 진로쇄정, 대향진로쇄정, 수렴쇄정 등의 쇄정을 하게된다. 이러한 일련의 과정이 모두 마무리되면 요청된 진로의 사용이 유효하게 되고 신호기나 마커로 게이트 개방신호를 전송하게 된다. 진로를 해정할 경우에도 궤도회로의 점유상태, 진로의 상태, 열차의 위치 등 모든 상태들에 따라 해정조건을 달리하게 된다.

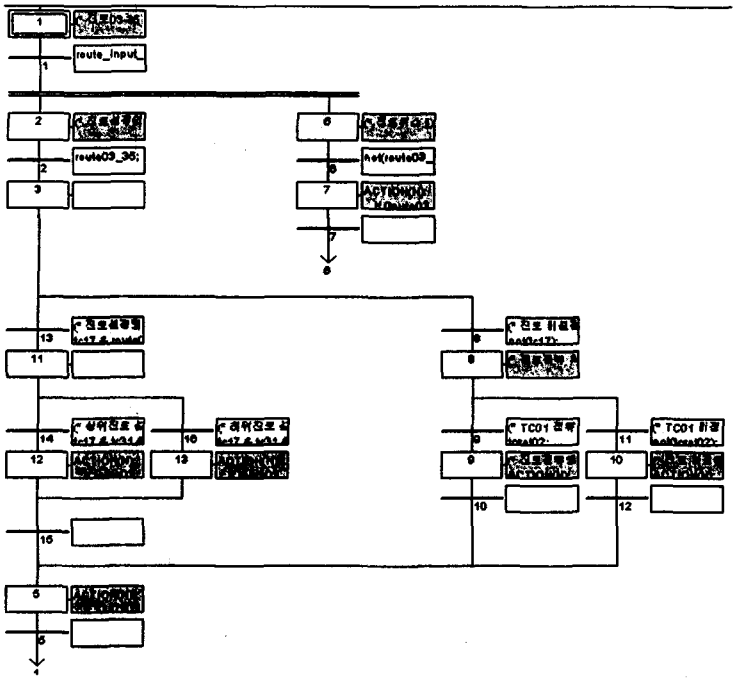


그림 6. Grafcet에 의한 모델링 예

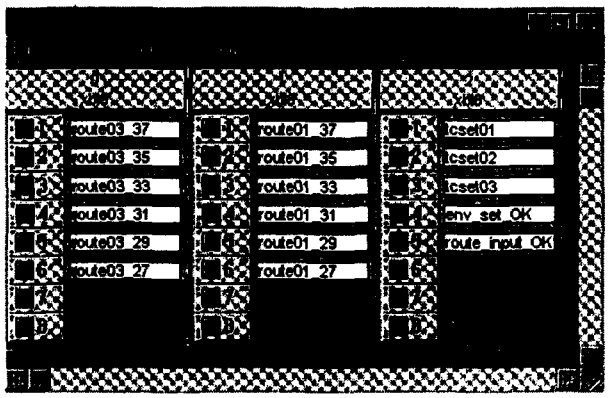


그림 7. 시뮬레이션 I/O 윈도우

시뮬레이션에서는 Grafcet으로 연동로직의 일부를 구성하였고 효과적인 분석을 위해 궤도회로와 진로 등의 상태가 표시되는 그래픽 화면을 구성하였고, 진로요청이나 선로변의 상태 등을 입력할 수 있도록 시뮬레이션 I/O 윈도우를 이용하였다. 아직은 연동로직 전체를 Grafcet으로 표현하지

못하였고 진로설정 프로세스의 일부를 시뮬레이션 할 수 있도록 하였다.

그림6은 하나의 예로서 진로요청이 있을 경우 진로의 가용성을 검사하여 요청된 진로가 설정가능한지를 검사하는 Grafcet 표현이다. 그림7은 시뮬레이션 I/O 윈도우로서 연동로직의 분석을 위한 각종 입력 윈도우이다. 여기에서 'tcset01', 'tcset02', 'tcset03' 등 임의로 각 궤도회로의 점유상태를 입력할 수 있도록 한 것으로 진로의 가용성 검사가 제대로 되는지를 분석하기 위함이다. 또 진로요청을 위해 각각 'route03-37' 등의 입력변수를 설정하였다. 그림8 (a)는 궤도회로 'tc01'이 점유된 상태에서 'route03-33'을 입력한 경우의 그래픽 화면으로 진로제어가 불가능함을 나타내고 있고, (b)는 같은 조건에서 'route03-35'의 진로요청을 하였을 경우의 그래픽 윈도우를 나타낸 것이다.

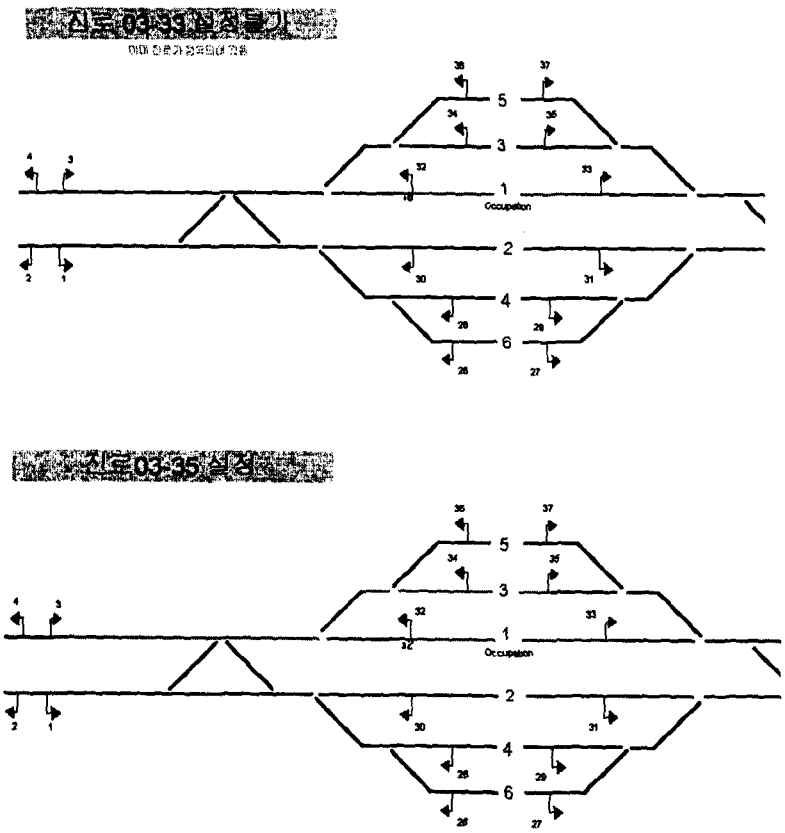


그림 8. 연동로직 분석을 위한 그래픽 윈도우

5. 결론

높은 레벨의 안정성과 신뢰성이 요구되는 전자연동장치의 연동로직의 분석을 위하여 Grafcet이라는 언어를 사용하였다. 이 언어는 페트리 네트에서 파생되어진 모델링 수단으로서 몇 가지 규칙만을 이용하여 시스템 모델링을 할 수 있다. 즉, 이 Grafcet은 제어시스템의 상태변화 및 제어의 흐름을 그래픽적으로 모델링 할 수 있고, 또한 제어의 흐름을 이해하기가 쉽게 되어있어 제어로직의 분석 및 동작사양을 표현하는데 적절하다. 또한 그래픽 윈도우와 연계하여 시뮬레이션을 수행

함으로서 제어로직의 분석에 용이하다. 본 논문에서 이 언어의 기본적인 표현방법을 설명하였고 이 언어를 이용하여 연동로직의 일부분을 시뮬레이션 하였다. 향후 이러한 언어를 이용하여 연동로직들을 모델링할 예정이며 제어감시용 전용 그래픽 툴과 연계하여 연동로직 분석용 시뮬레이션 프로그램을 개발할 계획이다.

<참고문헌>

[1] 福岡 博, 福田 光芳, 'ペトリネットによる連動仕様の検証', RTRI Report Vol. 9, No. 11, pp. 19-24, 1995.

[2] 土屋 陵田, 福田 光芳 and etc., '高安全性システムのための要求分析技術', RTRI Report Vol. 11, No. 8, pp. 25-30, 1997.

[3] J. Sagoo, J. Boardman, 'Formalisation of an Order Processing Soft Systems Model Using Petri nets', Proceedings of INCORSE'98, pp. ??-??, 1998.

[3] Rene David, 'Grafcet : A Power Tool for Specification of Logic Controllers', IEEE Trans. on Control System Technology, Vol. 3, No. 3, pp. 253-268, 1995.

[4] 한승수, 'GRAFCET을 이용한 프로그램형 제어기의 제어기능 설계 및 모니터링에 관한 연구', 석사학위논문, 연세대학교 전기공학과, 1989.

[5] P. Baracos, 'Tutorial Reference Guide to the Grafcet Automation Language : Grafcet Step by Step', Famic Technologies 2000 In., 1992.

[6] J. Rushby, 'Formal Methods and the Certification of Critical Systems', Technical Report CSL-93-7, pp. 14-105, December 1993.

[7] D. L. Dill and etc., 'Specification and Automatic Verification of Self-Timed Queues', IEEE Computer Society Press Tutorial on Formal Verification of Hardware Design, pp. 225-248, 1990.

[8] R. Tsuchiya and etc., 'Computer-supported Requirements Analysis for the Design of Safety-Critical Systems in Railways', Proceedings of WCRR'97, pp. 431-440, Nov. 1997.

[9] A. Cimatti and etc., 'Formal Validation & Verification of Software for Railway Control and Production Systems : Experimental Applications in ANSALDO', Proceedings of WCRR'97, pp. 467-473, Nov. 1997.

[10] R. Rawlings, 'Mathematical Techniques and the Development Lifecycle', Report of an IRSE Seminar, 15th. April 1996.

[11] Ian Mitchell and etc., 'Formal Mathematics for Signalling - A Tutorial Example', Report of an IRSE Seminar, pp. 17-30, 15th. April 1996.