

PtolemyII의 CCS도메인

황혜정⁰, 김윤정^{**}, 남기혁^{*}, 김일곤^{*}, 최진영^{*}

*고려대학교 컴퓨터학과
(hjhwang⁰,khnam igkim, choi)⁰@formal.korea.ac.kr

** 서울여자대학교 컴퓨터학과
(yjkim@swu.ac.kr)

CCS domain in PtolemyII

Hye-Jung Hwang, Ki-Hyuk Nam, Il-Gon Kim, Jin-Young Choi
Dept. of Computer Science and Engineering, Korea University

YoonJeong kim
Dept. of Computer Science, Seoul Women's University

요 약

PtolemyII는 내장형 시스템과 같이 이질적 성질을 가진 병렬 시스템을 모델링하고 디자인 할 수 있는 환경을 지원해준다. PtolemyII의 CSP 도메인은 병렬 시스템을 효과적으로 명세할 수 있는 프로세스 알제브라 언어인 CSP를 구현한 것이다. CCS도 프로세스 알제브라 언어로써 정형적으로 병렬시스템을 명세하고 검증할 수 있다. 이 두 프로세스 알제브라 언어는 비슷한 목적으로 가지고 개발되었으나 통신의 세부적인 부분의 개념에 있어서 차이가 존재한다. 본 논문에서는 CCS 기반의 통신이 이루어 질 수 있는 CCS 도메인을 구현하여 PtolemyII에 추가하였으며, PtolemyII에 구현된 CCS 도메인의 커널구조와 구현 방법을 설명하겠다. CCS도메인 구현은 기존의 개발자들이 actor-oriented 디자인을 하는 PtolemyII의 디자인 방법을 따라 기존의 프로세스 알제브라 언어들의 문법을 익히지 않고도 모델링 하고 시뮬레이션 할 수 있다는 것을 의미한다. 또한 PtolemyII에 CCS도메인을 추가함으로써 정형적 명세의 다양성을 높였다.

1. 서 론

내장형 시스템은 범용적인 목적을 가진 워크스테이션이나 데스크탑 컴퓨터 등과는 달리 특정 기능을 수행하기 위해, 컴퓨터 하드웨어와 소프트웨어가 조합된 전자제어 시스템을 일컫는다. 내장형 시스템 중 실시간 내장형 시스템(realtime embedded system)의 경우는 시스템의 주변 환경이 요구하는 속도에 맞추어서 지속적으로 환경과 상호작용을 해야만 한다. 이 시스템은 성능보다는 정확성과 안전성이 더욱 요구되어 지는 환경에서 자주 쓰인다. 예를 들면 원자력 발전소나 항공기 제어 시스템과 같은 것이 이에 해당한다. 이러한 시스템의 경우는 안전성이나 정확성이 만족되어지지 않을 때는 심각한 재해를 가지고 오게 된다. 이러한 시스템에 대한 정형 검증(formal verification)과 자동화 도구를 사용한 구현은 일반적인 테스트보다는 안전성을 보장하는 좋은 방법이다. 따라서 시스템이 하드웨어와 소프트웨어로 나누기 전, 추상화된 높은 레벨에서 시스템의 행위를 묘사할 때는, 반드시 하나 이상의 정형 모델(formal model)을 기반으로 해야한다[1]. 즉 정형 명세(formal specification)하는 과정이 요구된다. PtolemyII는 추상화된 레벨로 시스템을 명세할 수 있게 해주는 도구이다.

기존의 오토마타는 병렬 시스템(concurrent system)을 정확히 표현하는데 한계가 있었다. 이러한 한계 때문에 병렬적으로 통신하는 시스템을 묘사하고 설계하기 위해

CSP(Communicating Sequential Processes)[2], CCS(A Calculus of Communicating System)[3]와 같은 시간개념이 없는 여러 프로세스 알제브라 언어가 개발이 되었다. 이러한 프로세스 알제브라 언어는 프로토콜이나 시스템을 보다 정확하게 정형적으로 명세할 수 있도록 의미론과 문법을 제공한다. 그러나 이러한 언어는 각 언어의 전문가적 지식을 가지고 있지 않은 개발자는 다시 이 언어들의 문법과 의미론을 익힌 뒤야 정형적 명세를 할 수 있다. 그러나 프로세스 알제브라 언어인 CSP를 톨레미의 도메인으로 구현하여 톨레미만의 actor-oriented program[4]을 사용하여 복잡한 정형적 명세언어를 익히지 않고도 CSP를 명세할 수 있다.

본 논문에서는 프로세스 알제브라 언어인 CCS를 PtolemyII의 정형적 MoC(model of Computation)로 선택하여 java로 CCS 의미론을 구현하여 PtolemyII에 CCS 도메인을 추가하였다. 본논문의 구성은 다음과 같다. 2절에서 PtolemyII에 대한 소개를 3절에서 중요한 커널 부분의 구현 설명 그리고 마지막 4절에서 예제를 보도록 하겠다. 자세한 세부 부분은 참고문헌 [5]를 참조하기 바란다.

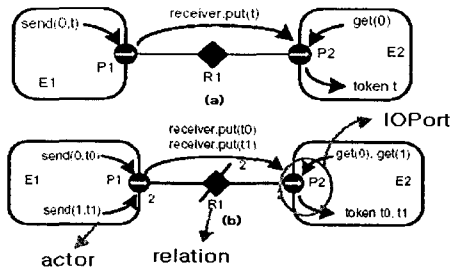
2. 본론

2.1 PtolemyII(PtolemyII)소개 및 장점

U.C 버클리에서 내장형 시스템을 정형적인 이론을 기반으로

로 하여 설계하고, 이질적인 의미들을 이용하여 설계된 내장형 시스템을 통합하여, 디자인 할 수 있는 도구인 PtolemyII를 개발하였다[6]. PtolemyII는 여러 가지 다른 의미론을 가진 이른바 도메인이라고도 불려지는 여러개의 MoC를 가지고 있으며 이중에서도 프로세스 알제브라 언어의 하나인 CSP[2]를 가장 전형적인 MoC중 하나로 채택하여 구현하였다. 모델링이라는 것은 시스템이나 하위 시스템을 정형적으로 표현하는 행위를 일컫는다. 반면 디자인은 시스템이나 하위 시스템을 정의하는 행위를 일컫는 것이다[7]. 어떠한 시스템을 설계하는 과정에서 디자인과 모델링 하는 과정은 필수적이며 또한 개발 단계시에 중요한 작업이다. 보통 시스템을 개발할 때 높은 레벨에서 시스템의 행위를 디자인 할수록 시스템의 행위가 많이 추상화 되어 진다. 즉 추상화가 많이 되어질 수록 개발자나 사용자의 이해가 쉬워진다. 또, 모델을 개발하는 데 드는 비용은 추상화 레벨이 낮아 질수록 차츰 더 많이 들게 되어진다. 이는 개발시에 모델의 요구사항들이 여러 요인들로 인해 변경되어질 수 있기 때문이다[7]. 이러한 개발 단계에 생길 수 있는 문제점들을 보완하고 보다 정확하고 효율적으로 시스템을 디자인하고 모델링하기 위해 만들어진 것이 PtolemyII이다.

PtolemyII의 큰 장점은 여러가지 프레임워크를 하나의 도구에서 구현할 수 있고, 각각 다른 프레임워크로 디자인된 모델을 계층적으로 연결하여 시스템을 하나의 도구를 사용하여서 디자인 하고 모델링하며 시뮬레이션 까지 설계과정에서 할 수 있다는 것이다. 현재 PtolemyII에서는 총 11개[6]의 프레임워크를 제공하고 있으며 현재 매뉴얼에 추가되지 않고 도구에만 추가된 도메인 2개를 합하여 총 13개의 다른 의미론을 가진 프레임워크들이 존재한다. 프레임워크는 다른 말로 '도메인'이라고도 일컬어지는데, 이러한 여러 도메인들은 각 구성 요소들 사이에 계산을 지배하는 물리적인 규칙을 말하는 'Model of computation'이다. 시스템의 이질적인 특성을 가진 여러 부분들을 각각의 단위에 최적의 MoC을 적용함으로써 시스템 디자인의 절적인 부분의 향상을 높일 수 있다. 톨레미는 Vergil이라는 도구를 사용하여 명세가 가능할 뿐만이 아니라 PtolemyII의 커널의 클래스들을 자바의 API처럼 사용하여 applet을 사용하여 모델을 명세 할 수 있다. 본문에서는 PtolemyII의 CCS도메인의 CCS커널부분의 구현세부 사항을 설명하고 톨레미의 두가지 모델 명세 방법중에서 applet을 사용한 예제를 보도록 하겠다.



<그림 1.(a) PtolemyII에서 하나의 channel을 가지고 있는

relation (b) 두개의 channel을 가지고 있는 relation >

3. CCS 도메인 커널의 구현 세부사항

3절에서는 톨레미II의 CCS구현 세부사항을 설명하기 위해서 PtolemyII의 각각의 도메인의 아키텍처중에서 중요한 커널부분인 Receiver[6]와 Director[6] 그리고 IOPort[6]를 중심으로 설계방법을 설명하도록 하겠다.

3.1 CCS Receiver

Receiver는 PtolemyII에서 출력 port로부터 데이터를 받아서 저장하는 일종의 저장소나 저장된 데이터를 주고 받는 메소드가 구현되어 있다[5]. 랑데뷰를 기반으로 하는 모델은 동시적이며 순차적인 프로세스들로 구성되어 있다. 프로세스들은

동기화 시점(synchronization points)에서 서로 통신한다. 즉, 동기화 시점에 도달해야 통신을 할 수 있다. 프로세스 알제브라의 언어들은 앞에서 언급한 랑데뷰 통신을 기본으로 한 모델을 사용하는데 이러한 대표적인 프로세스 알제브라 언어로는 CCS와 CSP가 여기에 속한다.

CCS의 경우는 랑데뷰를 기반으로 하여 프로세스간 통신을 하지만은 tau와 restriction의 발생 여부에 따라서 각각 프로세스들은 동기화 시점에 이를 수도 있고 이르지 못할 수도 있다.[2] 일반적인 랑데뷰알고리즘은 tau와 restriction의 확인 없이, 통신하고자 하는 두 프로세스가 통신할 준비가 되었다면은 랑데뷰에 성공하게 된다. 일단 통신하고자 하는 프로세스의 restriction은 CCSIOPort에서 시스템을 디자인 할때 개발자가 restriction의 값을 설정해 주며 설정이 안되었을 경우는 기본적으로 거짓 값을 가진다. restriction에 대해서는 CCSIOPort에서 설명하도록 하겠다. get이 일단 호출되면 통신을 하려고 기다리고 있는 상대방 프로세스가 있는지를 확인한다. 이것은 putwaiting이라는 flag를 통해 알수 있는데 put 메소드를 호출한 sender는 자신이 도착했음을 나타내기 위해 flag를 참값으로 바꾼다 그리고 일단 대기 상태로 들어간다. putwaiting flag가 참일 경우는 서로 랑데뷰 통신을 하고자 준비를 한다. 준비하는 과정은 아직 랑데뷰가 이루어지지 않았음을 나타내는 RendezvousComplete를 거짓 값으로 바꾼다음 대기하고 있는 모든 프로세스들을 깨운다. 이때 대기하고 있던 쓰레드는 통신을 하기위해 깨어난다. 그리고 대기상태가 해제된 get() 메소드를 호출한 상대방 프로세스는 자신의 flag의 값이 변경이 되었는지를 확인한다음에 자신과 통신하고자하는 프로세스의 restriction 정보를 확인한다. 자신과 통신하는 프로세스가 조건 객체(conditional object)인[8] conditionalReceiver나 conditionalSend일 수도 있고 조건 객체가 아닌 일반 Actor일 수도 있다. 따라서 자신이 통신하려는 프로세스가 조건 객체일 경우는 조건 객체의 각 클래스 (예를 들면 conditionalReceive, conditionalSend 클래스) 안에서 통신하려는 상대방 프로세스의 restriction을 먼저 확인하고 자신과 상대방의 restriction이 참인 경우와 tau flag의 값을 모두 먼저 확인하기 때문에 CCSReceiver에서는 통신하고자 하는 프로세스가 조건 객체인지의 여부만 확인한다. 이미 CCSReceiver안으로 들어온 조건 객체들은 restriction이 true이거나 tau flag가 참인 경우이다. 그리고 조건 객체가 아닌 객체와 통신할 경우는 자신의 restriction과 상대 프로세스의 restriction 값을 확인하고 그 값에 따라 랑데뷰를 수행한다. 또 restriction이 설정이 되지 않았으나 tau의 값이 참이 될 경우도 랑데뷰를 수행한다. 그러나 tau의 값이 거짓일 경우는 두 메소드(get()과 put())는 null을 반환하게 되고 RendezvousComplete는 거짓 값을 갖게 된다.

3.2 CCS Director

일반적으로 Director는 전체 actor들의 실행 순서를 스케줄링 하고, 전체 actor들간의 통신 방법을 제어한다. CCSDirector에서는 기존의 CSPDirector[8]의 시간 개념을 그대로 유지하였다. 여기서 시간의 개념이란 timed CSP와는 조금 다르다. 전체적인 실행 시간이 아니라 랑데뷰 수행시 상대방 프로세스가 통신 준비가 되어 있지 않을 경우 일정 시간 동안 잠서 금지되어질 수 있는 시간을 의미한다. 만약 시간을 설정하지 않으면 기본적인 CSP의 의미론을 기반으로 하여 통신할 수 있게 된다. 이러한 CSP의 시간개념을 CCS에서도 그대로 유지 하였다. 그러나 restriction이나 tau의 값을 확인할 때 단지 actor만 block list에 저장되었다. 전의 CSP와는 달리 조건 객체(conditional object)인 쓰레드도 블록 시킬때 block 된 쓰레드의 list를 생성함으로써 인해서 각각의 조건 객체가 아닌 것 뿐만 아니라 조건 객체까지도 블록 리스트 생성이 됨으로써 조건 객체가 가지고 있는 restriction과 그 외의 정보를 사용할 수 있도록 하였다. 이것은 프로세스를 제어하는 간접적인 방법으로는 get()과

put()함수를 들어가기 전 자기의 정보를 리스트에 넣고 restriction값까지 넣음으로 해서 get()과 put()메소드 안에 일단 들어가면은 그때 자신의 정보를 찾아서 restriction값을 가지고 파라미터화 할 수 있게 되어진 get()과 put() 메소드에 들어가게 된다.

3.3 CCS IOPort

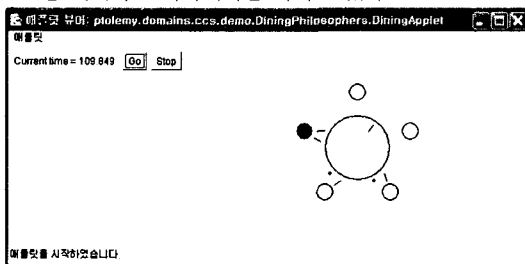
PtolemyII에서의 IOPort는 actor간의 데이터를 전송하고 받는 입구이다. 모든 Port는 입력 채널과 출력 채널들의 집합이다. 하나의 채널은 통신하고자 하는 두개의 프로세스의 연결을 의미한다. 즉 소비자 actor와 생산자 actor가 가지고 있는 각각 port의 연결을 말한다. 입력port와 출력 port를 연결하는 채널은 CCS의 의미론으로 설명이 되는데, 예를 들면 입력 port는 a라는 action을 말하고 출력 port는 \bar{a} 라는 action을 의미한다. 이 채널을 통해서 token이 주고 받아 진다면 이것은 tau나 restriction값에 의해서 랑데뷰 통신이 발생한 것을 의미한다. Port는 반드시 자신을 포함하고 있는 컨테이너를 가지는데 이것은 actor를 의미한다. port는 두개의 타입을 가지는데 하나의 채널만을 가질수 있는 단일 port와 여러개의 채널을 가질수 있는 다중 port가 있다. 채널은 두개의 port가 서로 통신이 가능하도록 연결된 것을 말한다. 또 Port들은 relation에 의해서 묶여질 수 있다. CCSIOPort는 기존의 IOPort와 그 하위 클래스인 TypedIOPort를 상속하였으며, restriction 파라미터를 가지고 있다.

restriction 파라미터는 시스템 디자이너가 CCS 도메인을 사용하여 디자인 했을 경우 이 값을 정해줌으로써 CCS 기반의 통신에서 restriction을 표현할 수 있게 하였다. 또한 이렇게 정해진 값인 restriction은 여러 메소드를 통해 값이 CCSIOPort에 저장되고 사용되어 진다. 이것은 랑데뷰가 이루어지는 해당 Receiver에서 랑데뷰통신 전에 확인된다.

4. 예제

DiningPhilosopher 문제는 1965년 Dijkstra에 의해서 처음 제시되었다. 이것은 concurrent programming에서 두개의 기본적인 Property인 Liveness와 Fairness에 대해 설명할 때 주로 사용하는 예제이다. 이 문제는 두가지 속성을 가지고 있는데 Liveness와 Fairness이다. Liveness를 묻는 것은 Deadlock을 피할 수 있게 디자인 할 수 있는 지의 여부를 묻는 것을 의미한다. 이와 같이 Fairness를 묻는 것은 starvation을 피할 수 있게 디자인 할 수 있는 지의 여부를 묻는 것이다.

구현에서 각각의 철학자는 임의적으로 첫 번째 젓가락을 집도록 되어 있다. 모든 철학자가 같은 비율로 생각하고 음식을 먹는다. 각각의 철학자와 젓가락은 하나의 프로세스들로 설계된다. 각각의 젓가락은 CDO를 사용하여 조건 객체(conditional object)로 설계된다. 일단 철학자가 젓가락을 잡으면 임의의 시간동안 먹는다. 각각의 프로세스들은 다른 프로세스들과 CCSIOPort를 사용하여서 연결이 되는데 CCSIOPort는 restriction을 의미하는 파라미터를 가지고 있다.



[그림 2. Dining Philosopher 예제]

DiningPhilosopher 문제에서 하나의 젓가락은 각각의 양옆의

철학자와 연결된 port를 2개씩 가진다. 즉 양쪽의 철학자와 통신을 위한 입출력 port를 가진다. 예를 들면 Leftout은 젓가락 프로세스가 왼쪽 철학자와 통신하기 위한 port이다. 이렇게 젓가락은 각각 두개씩의 입출력 port를 가지고 있다. 세부적으로 설명을 하자면, 이 때 젓가락은 두개의 입력 port를 가지고 있는데 양 옆의 철학자가 하나의 젓가락을 동시에 사용하고자 할 때 젓가락 안에서는 조건 객체인 ConditionalDO[8]를 사용하여서 그 안에서 ConditionalSend를 생성하게 된다.

이 때 chooseBranch함수를 호출하여 두개의 쓰레드중에서 통신을 할 쓰레드가 정해지고 이 때 어떠한 철학자와 통신할지 결정되어 지게 된다.

5. 결론 및 향후 연구 방향

본 논문에서는 PtolemyII에 CCS 도메인을 추가하고 구현하였다. CCS 도메인을 설계하여 PtolemyII에서 디자인을 설계할 수 있는 정형 도메인의 다양성을 가져왔을 뿐만이 아니라, PtolemyII를 사용하는 개발자가 기존의 PtolemyII에서 시스템을 명세하고 디자인 하는 패턴인 actor-oriented Programming 방식과 또 톨레미 커널을 사용하여 applet으로 모델을 시뮬레이션할 수 있는 두가지 방법으로 그대로 사용하여서 정형적 명세가 되어진 시스템을 개발할 수 있게 한다. 나아가서는 시간과 자원 그리고 우선순위를 정형적으로 설계 및 검증할 수 있도록 PtolemyII에 ACSR(Algebra of Communicating Shared Resources)[9] 도메인을 구현 및 추가할 수 있는 기반을 마련하였다. 그러나 아직 다른 도메인과의 계층적 연결에 있어 기존의 CSP가 가진 한계를 극복하지 못했다.

향후 연구방향은 CCS를 기반으로 한 ACSR을 PtolemyII에 도메인을 구현 추가하는 것이다.

6참고문헌

- [1] Edward A. Lee 외 4명 Design of Embedded Systems: Formal Models, Validation and Synthesis. *Proceedings of IEEE* VOL.85 No.3 March, 1997 pp.366-390
- [2] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall International, 1985
- [3] Robin Milner, *Communication and Concurrency*, Prentice Hall international Ltd, 1989
- [4] Edward A. Lee 외 3명 Actor-Oriented Control System Design : A Responsible Framework Perspective *IEEE Transactions on control systems technology*. VOL.X No. Y Month-Z 2003, Revision.
- [5] H.-J. Hwang, K.H. Nam, I.-G. Kim, J.-Y. Choi, PtolemyII의 CCS 도메인, 2003년 정보과학회 논문지 심사중
- [6] Edward A. Lee 외 17명 Heterogeneous Concurrent Modeling and Design in Java, UCB/ERL M01/12, University of California, Berkeley, March 15, 2001
- [7] Edward A. Lee, "Embedded Software," *Advances in Computers* (M. Zelkowitz, editor), Vol. 56, Academic Press, London, 2002
- [8] Neil Smyth, *Communicating Sequential Processes in Ptolemy II*, University of California, Berkeley, December 15, 1998.
- [9] I. Lee, H. Ben-Abdallah, and J.-Y. Choi, A Process Algebraic Method for Real-Time Systems, *Formal Methods for Real-Time Computing* C. Heitmeyer and D. Mandrioli (eds), John Wiley & Sons Ltd, 1996