

조업자 교육시스템의 사용자 모듈과 OM모듈간의 전송 데이터 구조

박선용, 조성일, 문 일

연세대학교 화학공학과

Transmitting Data Structures between User and OM Module in OTS

Seon Yong Park, Seong Il Cho, Il Moon

Department of Chemical Engineering, Yonsei University

서론

OTS(Operator Training System)의 구성요소중 교육자(instructor)와 조업자(operator)의 사용자 환경은 효과적인 교육을 위해서 체계적이고 유연한 환경으로 구성되어야 한다. 특히 교육자의 경우 조업자의 전체적인 훈련상황을 통제할 수 있어야 하기 때문에 조업자의 훈련 과정 선택, 훈련과정의 조정, 결과의 표시 및 조업자의 상태 분석 등 감독자로서의 기능을 수행할 수 있어야 한다¹⁾. 교육 단계가 정상조업 훈련단계(normal operation training session)일때 조업자가 다중으로 접속하여 훈련이 이루어지는 경우 각 조업자의 입력과 조업자에 대한 교육자의 간섭으로 인한 입력등으로 OM(OTS Manager)로 전달되는 데이터와 그에 따라 OM이 응답하는 정보들은 매우 복잡한 과정을 거치게 된다. 따라서 OM과 각 사용자 모듈중에서 이러한 동작을 수행하기 위해 상호간에 전달하고 실행하는 연결부분의 알고리즘이 OM과 사용자간의 동작을 원활하게 하는데 핵심적인 부분이 된다. 본 고에서는 OM에 연결된 사용자로부터 입력된 신호를 해석하고 이에따라 적절한 기능을 수행하는 OM과 사용자 접속부분의 알고리즘을 화학공정의 정상조업 훈련단계를 중심으로 기술한다.

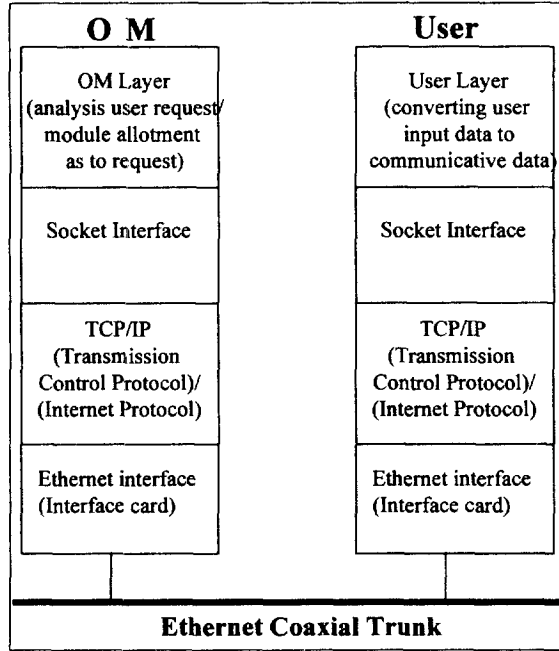
사용자 모듈과 OM모듈간의 인터페이스

설명 단계(tutorial session)와는 달리 정상조업 훈련단계에서는 조업자가 표준 조업작업에 따라 공정을 진행시킨다. 이때 교육자는 이를 모니터하며 오류가 진행될 때 이를 간섭할 수 있으며 초기에 교육자가 훈련할 내용을 선택할 수 있어야 한다. OM모듈과 사용자모듈의 정보교환 개괄도²⁾는 Fig 1과 같다. OM과 Solver가 위치해 있는 워크스테이션과 사용자들이 사용하는 이 기종 컴퓨터간의 통신을 위해서 양방향 통신기능이 있는 소켓(socket)을 이용하였다. 초기화단계에서 OM서버는 비활성소켓(passive socket)을 생성하고 사용자의 연결요청이 들어올때까지 기다렸다가 연결을 완성한다. 상호간의 통신이 가능해진 이후에는 OM은 다음의 과정을 따라 연결된 사용자에 필요한 자원들(resources)을 배당한다.

- 연결이 이루어진 사용자를 교육자와 조업자로 구분하여 확인한다
- 교육자와 조업자가 모두 등록된 이후에는 필요한 자원(공유 메모리, 교육자 데이터, 조업자 데이터, 필요한 Solver의 갯수, 모사 데이터, 평가 데이터)을 할당한다.
- 교육자의 입력과 조업자의 입력을 받아들여 이를 분석하여 필요한 모듈을 실행시키거나 명령이나 결과를 사용자모듈에 전달한다.

사용자 데이터 구조

교육자의 경우 전체 시스템을 관리하는 포괄적인 기능을 담당하므로 초기 등록과 정상조업시 사용자가 전달하는 데이터가 세분화 되므로 사용자중에서 교육자 전달데이터 구조를 중심으로 살펴보면 표 1과 같다. 표중 굵은선 안의 데이터는 교육자에게 추가되는 데이터이다. 초기 접속시 사용자는 사용자 확인에 필요한 데이터를 OM으로 보내고 OM은 자신의 데이터 베이스와 비교하여 사용자의 등록, 갱신 등을 수행한다. 이후의 과정에서는 교육자는 조업자 훈련단계(training session)의 선택, 훈련과정(training course)의 선택 등 교육에 대한 선택을 임의로 할 수 있어야 하며, 훈련도중 각 조업자의 상태를 임의로



조정할 수 있어야 한다. 따라서 OM으로 전달되는 데이터는 교육훈련에 필요한 데이터와 공정모사에 관계된 데이터로 구분된다. OM에서 시용자에게 보내는 데이터는 대상이 교육자인 경우 조업자의 현재 상태를 표시하는 정보가 포함되게 되며 공정상태에 대해서는 조업자에 따른 복수의 데이터가 존재하게 된다.

sending data from user to OM		sending data from OM to User	
user identification	user id, password	operator identification (instructor only)	operator id
	description, date		present training session
	change to operator (instructor only)		present training course
training session	present training session	process status	time
	present training course		model status
process status	time		var1
	simulation time increment (Δt)		var2
	process status		var3
	unit num	...	
	task num		
	specified value		

Table 1. Transmitting data structure between OM and user modules

OM의 사용자 입력처리 알고리즘

정상조업 훈련단계를 예로 교육자 모듈 1개와 조업자 모듈 2개가 연결된 경우

OM의 사용자 알고리즘은 Fig 2와 같다.

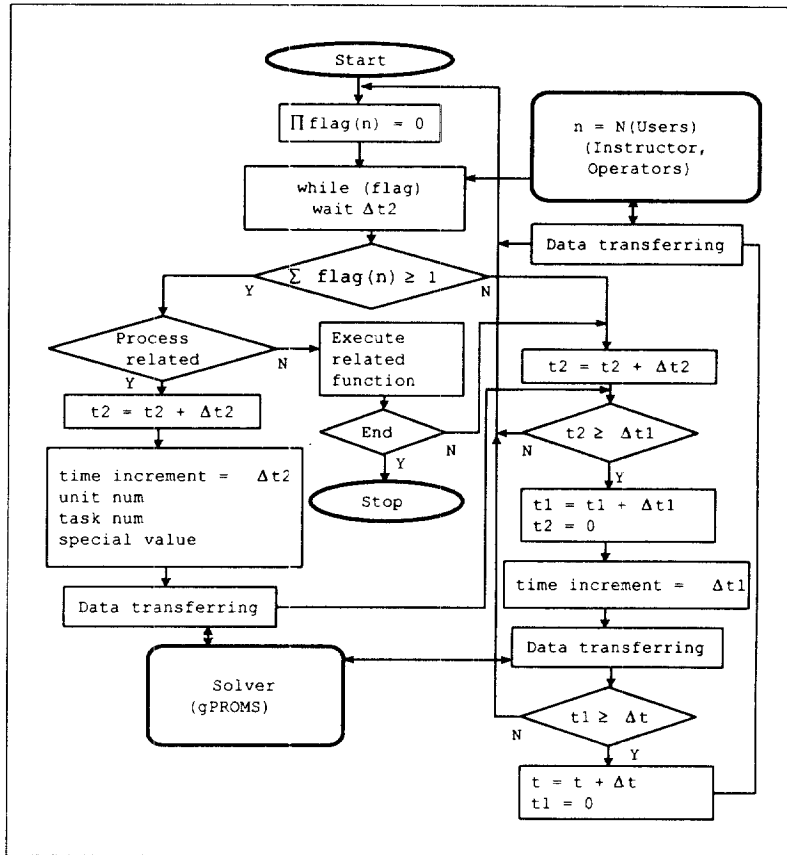


Fig 2. OM interface algorithms for adjusting time

여기서 Δt , Δt_1 , Δt_2 은 시간의 증분을 의미한다. Δt 는 사용자쪽에 계산된 데이터가 표시되는 시간간격을 의미하며 Δt_1 은 OM이 외부 이산사건이 생성되지 않았을 때 공정을 계산하는 시간간격을 Δt_2 는 비분질적인 이산사건³⁾이 발생했거나 사용자측의 입력을 기다리는 시간간격을 의미한다. 사용자측에서 입력이 들어오면 각 사용자에게 해당하는 데이터 구조체의 신호(flag)값이 참(true)로 설정되며 각 신호값들의 합이 1보다 큰 경우 사용자측의 입력에 따라 계산이 필요한 과정인가 아닌가로 분기하며 신호값이 0인 경우 Δt_1 만큼 시간을 증가시켜서 계산을 하고 사용자측에 의해 정해진 출력시간간격 Δt 에 따라 사용자측에 데이터를 전송한다. 전체과정이 끝나지 않은 경우 루프를 반복해서 수행하게 되며 반복전에 신호값들을 모두 0으로 재설정한다.

시간분할 결정

위에서 제시한 알고리즘은 사용자에 의해 화면에 표시되는 시간간격 Δt ,과 OM서버가 입력을 기다리는 시간간격 Δt_2 이 있으며 이 시간은 상대적으로 Δt_1 에 비해 결정적이다. 그 사이에 일정한 시간간격으로 Solver가 계산하는 시간간격 Δt_1 이 있게 된다. Δt_1 이 Δt 을 m 개로 Δt_2 는 Δt_1 을 n 개로 분할한 것이라면 다음과

같은 관계가 성립한다.

$$\Delta t_1 > \sum_i^n (\Delta t_2) = n\Delta t_2 \quad (1) \qquad \Delta t > \sum_i^m (\Delta t_1) = m\Delta t_1 \quad (2)$$

전체 동작의 수행이 원활하려면 시스템의 계산 간격 Δt_1 과 n , m 이 결정되어야 하는데 한명의 조업자에 대해 각 기능의 수행 시간을 정리하면 아래와 같다.

Δt_1 동안 발생하는 조작의 개수

- 공정 관련 조작 : α 개 - 공정의 기능 실행 : β 개
- 조작 없는 경우 : $n - (\alpha + \beta)$ 개
- Δt_1 간격동안 계산하는데 걸리는 수행시간 : δ_1
- Δt_2 간격동안 계산하는데 걸리는 수행시간 : δ_2
- 공정의 기능 실행시 걸리는 시간 : γ_1
- 사용자에게 데이터 전송시 걸리는 시간 : γ_2

따라서 Δt_1 은

$$\sum_i^n (\delta_2) + \sum_i^n (\Delta t_2) + \sum_i^n (\gamma_1) + \sum_i^n (\Delta t_2) + \sum_i^{(n-(\alpha+\beta))} (\Delta t_2) + \delta_1 + \gamma_1$$

이 되며 (2)로 부터

$$\begin{aligned} \Delta t > \Delta t_1 &= \sum_i^m \left(\sum_i^n (\delta_2) + \sum_i^n (\Delta t_2) + \sum_i^n (\gamma_1) + \sum_i^n (\Delta t_2) \right. \\ &\quad \left. + \sum_i^{(n-(\alpha+\beta))} (\Delta t_2) + \delta_1 + \gamma_1 \right) \\ &= \sum_i^m (n\Delta t_2 + \sum_i^n (\delta_2) + \sum_i^n (\gamma_1) + \delta_1 + \gamma_1) \end{aligned} \quad (3)$$

과 같이 표현된다. 이 관계로 부터 Δt , Δt_2 , n , m 과 δ_1 , δ_2 , γ_1 , γ_2 에 따른 시간 분할 값의 범위를 결정할 수 있게 된다.

결론

OTS가 전체적인 다수의 사용자가 동시에 사용할 수 있기 위해서는 OM과 각 사용자간의 인터페이스 문제를 해결하여야 한다. 또한 교육자가 각 조업자의 훈련상황에 개입하고 조절할 수 있어야 하므로 OM과 사용자간의 정보교환은 필요한 데이터 구조와 각 데이터에 따른 효율적인 처리 알고리즘을 가지고 있어야 한다. 본 고에서는 인터페이스에 필요한 데이터와 알고리즘을 제시하였으며 현재 이를 YOTS(Yonsei OTS)에 적용하여 그 효율성을 입증하였다. 특히 위에서는 각 시간값과 무관하게 사용자에게 의해 발생하는 비본질적인 이산사건(extrinsic discrete event)을 시간값을 분할하여 처리함으로써 각 사용자에게서 들어오는 입력을 순서나 시간에 상관없이 처리할 수 있으므로 서버의 성능만 충분하면 복수의 사용자를 동시에 교육시킬 수 있게 된다.

감사의 글

이 연구는 지능자동화시스템연구소의 목적기초연구 연구비지원에 의해 이루어졌으며 지원해주신 연구센터에 감사드립니다.

참고문헌

1. 박선용, 문 일, 화학공업과 기술, 12(6), 22(1994)
2. S. Rajagopal, M. Rafian-Naini, J. Lake, C. Velo, A. Turke, P. Sigari, IEEE Trans. on Power Systems, 9(4), 1980(1994)
3. 박선용, 오 민, 문 일, "화학공정 동적모사에서 발생하는 이산사건의 분류 및 특성", 화학공학, 수록예정(1996)