대기행렬 페트리넷을 이용한 웹서비스 시스템 분석 방법

심규박[†], 임재걸^{††}, 이계영^{†††}, 김진석^{††††}

요 약

소프트웨어 시스템 개발의 초기 단계에서 시스템의 처리 시간이나 비용을 예측하는 것이 매우 중요하다는 것은 널리 알려진 사실이다. 많은 논문에 처리 시간을 예측하는 페트리넷 방법이 소개된 바 있다[1,2]. 한편, 컴퓨터 과학 분야의 다른 영역에서는 응용 시스템 개발의 효율성을 증진시키는 방법으로 웹서비스가 깊이 있게 연구되고 있다[3,4]. 이러한 두 가지 현상을 고려하여, 본 논문은 웹 서비스로 구성된 응용 소프트웨어 시스템의 처리시간을 분석하는 페트리넷 방법을 소개한다. 웹 응용 시스템의 처리시간을 어떤 상수로 표현하 기는 대단히 어렵다. 따라서 본 논문은 대기행렬을 갖는 페트리넷으로 웹 응용 시스템을 표현할 것을 제안한다. 본 논문의 핵심은 이러한 대기행렬을 갖는 페트리넷의 분석 방법을 소개한 것이다.

A Web Service System Analysis Method Using Petri Net with Queue

Kyubark Shim[†], Jaegeol Yim^{††}, Gyeyoung Lee^{†††}, Jinseog Kim^{††††}

ABSTRACT

It is well known that estimating the system processing time and cost at the early stage of the software system development is very important, and many Petri net methods of analysing processing time have been published [1,2]. In another field of computer science, web service has been widely studied as a key of an efficient application software development method [3,4]. Considering these two phenomena, this paper introduces a Petri net method of analysing the process time of application software consisting of web services. It is not plausible to represent the process time of a web application system as a constant. Therefore, we propose to use "Petri net with queues" to represent a web application system. The main contribution of this paper is to introduce a method of analysing a "Petri net with queues".

Key words: Web Service(웹서비스), Petri Net(페트리넷), Queue(대기행렬), System Analysis(시스템 분석), SOA(서비스지향구조), Formalization(정형화)

1. 서 론

SOA(Service Oriented Architecture)[3,4]는 웹 상에서 서비스를 효율적으로 공유하는 방법이고 개 발자들에게 이미 존재하는 웹 서비스들을 조합함으 로써 새로운 업무 처리 시스템을 효율적으로 구축하는 방법을 제공한다. 조합된 시스템이 원래 의도한 대로 작동하는 것을 보장하기 위하여, 요구사항 분석 단계나 설계 단계에서 웹서비스들 간의 정보 흐름과 상호 작용을 정형화하여 표현하고, 이것을 분석하는

 [※] 교신저자(Corresponding Author): 심규박, 주소: 경북 경주시 석장동 707 동국대학교(780-714), 전화: 054)770-2245, FAX: 054)770-2520, E-mail: shim@dongguk.ac.kr 접수일: 2011년 4월 25일, 수정일: 2011년 8월 29일 완료일: 2011년 10월 7일

^{*} 정회원, 동국대학교 과학기술대학 정보통계학과 교수 ** 종신회원, 동국대학교 과학기술대학 컴퓨터공학부 교수 (E-mail: yim@dongguk.ac.kr)

^{****} 정회원, 동국대학교 과학기술대학 컴퓨터공학부 교수 (E-mail: lky@dongguk.ac.kr)

^{*****} 정회원, 동국대학교 과학기술대학 정보통계학과 조교수 (E-mail: jinseog.kim@gmail.com)

[※] 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2011-0006942).

일을 수행한다.

[5]는 웹 응용시스템의 제어 흐름의 모형을 구축하는 페트리넷(Petri Net) 방법을 소개한다. 제어 흐름의 페트리넷 모형이 구축되면 그들이 소개하는 대수적 방법으로 웹 응용시스템의 신뢰도를 증명할 수있다. 그러나 [5]는 시간 관리 부분을 다루지 못하고향후 연구 과제로 남겨두었다. 이 논문은 [5]가 향후연구 과제로 남겨둔 시간 관리를 위하여 대기행렬페트리넷 방법을 제안한다. 특히 고객들의 도착이Poisson 과정으로 일어나고 서비스 시간들이 동일하지만 독립적으로 지수분포를 따르는 M/M/1 대기행렬시스템을 사용한다.

2. 관련 연구

요구사항을 만족하지 못하는 시스템은 분쟁의 소지만 될 뿐 아무런 소용이 없기 때문에, 개발 중인시스템이 사용자 요구사항을 만족하는지 검증하는 것은 매우 중요한 일이다. 그래서 웹서비스를 조합하여 구축하는 응용시스템을 검증하는 정형화 방법들이 많이 연구되었다. 예를 들어 [6]은 조합넷이라는 페트리넷을 제안한다. 조합넷에서는 웹서비스들 간의 비호환성이 교착상태(deadlock)로 나타난다. 이논문의 가장 흥미 있는 공헌은 교착상태를 제거하는 정책을 제시한 것이다. 이 정책을 적용함으로써 웹서비스들 간의 비호환성 문제를 자동적으로 해결할 수있다.

[기은 응용시스템을 구축하기 위하여 웹서비스를 조합하는 것을 표현하는 페트리넷 방법과 이러한 페트리넷의 도달성 분석 방법을 제안한다. 이 논문의 단점은 웹서비스 조합 유형이 세 가지밖에 되지 않아모든 조합유형을 다루고 있는지 의문이라는 것과 도달성 분석에 걸리는 시간이 너무 길다는 것이다. [8]은 웹서비스 시스템의 데이터 흐름을 칼라 페트리넷으로 표현할 것을 제안한다. 이들도 시스템 분석 방법으로 페트리넷의 도달성 검사를 사용한다. 도달성검사의 약점인 실행 시간을 절약하기 위하여 이들은도달성 검사에 앞서 축소(reduction)를 선행한다.

IBM과 마이크로소프트는 이미 모바일 웹서비스 프레임워크를 제안한 바 있다. [9]는 모바일 웹서비스에서 메시지와 메소드의 흐름을 나타내는 일반화 통계적 고급 페트리넷 모형을 제안하였다. 그리고 이

모델을 분석하여 시스템에 내재된 교착상태를 검출하는 방법도 소개하였다. 소개된 방법은 [8]과 비슷하게 도달그래프(reachability graph)를 이용한다. 도달그래프는 초기마킹을 뿌리노드로 하여 초기마킹으로부터 격발가능한 모든 트랜지션들의 격발 결과를 자식노드로 추가하여 작성하는 그래프로, 시스템의 초기상태로부터 도달 가능한 모든 상태를 낱낱이열거하는 것과 동일하여 작성 시간이 매우 길다는단점이 있다. 즉, 도달그래프의 임의의 노드는 일반적으로 여러 자식노드를 갖기 때문에 도달그래프의 노드의 총 수는 격발가능 트래지션의 수를 지수로하는 수가 된다. [9]는 모바일 웹서비스의 페트리 넷모형을 제시한 초기 논문으로 가치가 크지만 효율적인 분석 방법까지는 제시하지 못하였다.

본 논문은 [5]의 결과를 확장하여 통계적 시간 분석이 가능하도록 한다. 그래서 공서비스, 원자서비스, 순차조합, 택일조합, 비순서순차 조합, 반복조합, 통신이 있는 병렬서비스, 식별조합, 선택조합, 정제조합 등 [5]에서 소개된 조합 연산들 각각을 대기행렬로 확장한다. 즉, 이들 각각에 대한 시간 분석 방법을 제안한다.

3. 대기행렬 페트리넷

본 논문은 웹응용 시스템의 통계적 분석 방법을 제안한다. 웹응용은 다수의 원자 웹서비스로 구성된다. 원자 웹서비스는 정확히 하나의 입력 플레이스와 하나의 출력 플레이스를 갖는다. 입력 플레이스에 토큰이 들어오는 것은 바로 그 원자 웹서비스에 대한서비스 요구가 있다는 것을 나타내고, 출력 플레이스에 토큰이 놓이는 것은 해당 웹서비스의 수행이 완료되었음을 나타낸다. 각 원자 웹서비스의 처리시간은서비스 제공자에 의하여 이미 공표되었다. 그런데 원자 웹서비스의 입력 플레이스에 토큰이 도착하는 사건은 무작위로 발생한다. 그래서 입력 플레이스와 출력플레이스에 토큰이 나타나는 사건을 M/M/1 대기행렬 시스템으로 나타낼 수 있다.

고객들의 도착이 Poisson 과정으로 일어나고 서비스 시간들이 동일하지만 독립적으로 지수분포를 따르는 M/M/1 시스템은 다음과 같은 출생 및 사멸률을 갖는 출생사멸과정이다.

출생률 : $\lambda_n = \lambda \quad (n \ge 0)$

사멸률 : $\mu_n = \mu \quad (n \ge 1)$

출생률과 사멸률 사이의 비율을 ρ 라 하자. ρ 는 단위시간동안 도착하는 고객들에 의해 서버에게 주어지는 평균부하량인데, $\rho=\lambda/\mu$ 이다. 이 때, $\rho<1$ 인데, 단위시간동안 서버가 처리할 수 있는 최대능력보다작아야 하는데, 그렇지 않으면 처리하지 못하는 일의양이 계속 쌓여서 언젠가는 고객의 수가 무한대로증가할 것이기 때문이다.

원자 웹서비스의 대기행렬 페트리넷 모형을 그림으로 표현하면 그림 1과 같다. 입력 플레이스인 플레이스 i에는 토큰들의 대기행렬이 있다. 그리고 대기행렬은 출생률 λ 를 갖는다. 원자 웹서비스 S의 실행은 플레이스 i에 토큰이 도착할 때 시작하고 플에이스 o에 도착할 때 종료한다. 출력 플레이스 o는 사망률 μ 를 갖는다. 이때, 출생사멸과정의 초기상태와 안정상대는 다음과 같다.

$$P_{0} = \left[1 + \sum_{n=1}^{\infty} \left\{ \frac{\prod_{i=0}^{n-1} \lambda_{i}}{\prod_{i=1}^{n} \mu_{i}} \right\} \right]^{-1}$$

$$= \left[\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^{n}\right]^{-1} = 1 - \frac{\lambda}{\mu} = 1 - \rho$$
(1)

$$P_{n} = \frac{\prod_{i=0}^{n-1} \lambda_{i}}{\prod_{i=1}^{n} \mu_{i}} P_{0} = \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^{n}$$

$$= (1 - \rho) \rho^{n}, \quad (n \ge 1)$$
(2)

즉, 안정상태의 고객수 분포는 기하분포(geometric distribution)를 따른다. 임의시점에서의 평균고 객수를 L이라하면 L은 다음과 같다.

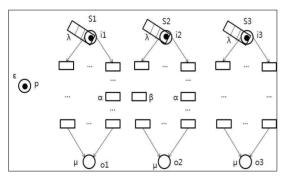


그림 1. 공서비스 ε과 원자 웹서비스 S1, S2, S3.

$$L = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n \rho^n (1 - \rho)$$

$$= \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}$$
(3)

서비스를 받고 있는 평균고객수는 실행로드와 같으므로 ρ 이다. 또한, 평균 대기고객수를 L_q 라 하면 L_z 는 다음과 같다.

$$L_{q} = \sum_{n=0}^{\infty} (n-1)P_{n} = \frac{\rho^{2}}{1-\rho} = \frac{\lambda^{2}}{\mu(\mu-\lambda)}$$
 (4)

평균대기시간을 W_q 라 하면 W_q 는 Little의 법칙으로부터 다음과 같다.

$$W_q = \frac{L}{\lambda} = \frac{1}{\mu - \lambda} = \frac{1}{\mu(1 - \rho)} \tag{5}$$

Case 1) 순차조합

웹서비스 조합 연산 중 가장 기본적인 연산은 S1 다음에 S2를 연결 수행하는 순차조합이다. 순차조합을 나타내는 대기행렬 페트리넷은 그림 2에 보인다. 고객의 수가 각각 n_1 과 n_2 인 두 개의 M/M/1 시스템이 독립적으로 연결되어 있을 경우, 순차조합 시스템의 고객수 분포는 다음과 같다.

$$\begin{split} P_{n1} + P_{n2} &= \sum_{i=1}^{2} \left(1 - \frac{\lambda_{i}}{\mu_{i}} \right) \left(\frac{\lambda_{i}}{\mu_{i}} \right)^{n_{i}} \\ &= \sum_{i=1}^{2} (1 - \rho_{i}) \rho_{i}^{n_{i}}, \quad (n_{i} \ge 1, i = 1, 2) \end{split} \tag{6}$$

① 순차조합 시스템에서 임의시점에서의 평균고 객수를 L_1 이라 하면, 식 (3)과 식 (6)에 의해 다음과 같이 계산할 수 있다.

$$L_{1} = \sum_{i=1}^{2} \frac{\rho_{i}}{1 - \rho_{i}} = \sum_{i=1}^{2} \frac{\lambda_{i}}{\mu_{i} - \lambda_{i}}$$
 (7)

2 순차조합 시스템에서 평균 대기고객수를 L_{q1} 이라 하면, 식 (4)에 의해 다음과 같이 계산할 수 있다.

$$L_{q1} = \sum_{i=1}^{2} \frac{\rho_i^2}{1 - \rho_i} = \sum_{i=1}^{2} \frac{\lambda_i^2}{\mu_i(\mu_i - \lambda_i)}$$
 (8)

③ 순차조합 시스템에서 평균대기시간을 W_{q1} 라하면, 식(5)에 의해 다음과 같이 계산할 수 있다.

$$W_{q1} = \sum_{i=1}^{2} \frac{1}{\mu_i - \lambda_i} = \sum_{i=1}^{2} \frac{1}{\mu_i (1 - \rho_i)}$$
 (9)

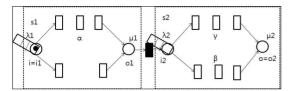


그림 2. 순차조합을 나타내는 대기행렬 페트리넷

Case 2) 택일조합

조합 연산 중 택일조합은 그림 3과 같이 웹서비스 S1과 S2 중 하나만 선택적으로 실행하는 조합이다. 고객의 수가 각각 n_1 과 n_2 인 두 개의 M/M/1 시스템 이 각각 weight w_1 과 w_2 를 가진 채 연결되어 있을 경우, 택일조합 시스템의 고객수 분포는 두 시스템의 평균으로서 다음과 같다. 단, $w_1+w_2=1$ 이다.

$$P_{n1} + P_{n2} = \frac{1}{2} \sum_{i=1}^{2} w_i \left(1 - \frac{\lambda_i}{\mu_i} \right) \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}$$

$$= \frac{1}{2} \sum_{i=1}^{2} w_i (1 - \rho_i) \rho_i^{n_i}, (n_i \ge 1, i = 1, 2)$$
(10)

① 택일조합 시스템에서 임의시점에서의 평균고 객수를 L_2 라 하면, 식 (3)과 식 (10)에 의해 다음과 같이 계산할 수 있다.

$$L_{2} = \sum_{i=1}^{2} \frac{w_{i} \rho_{i}}{1 - \rho_{i}} = \sum_{i=1}^{2} \frac{w_{i} \lambda_{i}}{\mu_{i} - \lambda_{i}}$$
(11)

② 택일조합 시스템에서 평균 대기고객수를 L_{q2} 라 하면, 식 (4)과 식 (10)에 의해 다음과 같이 계산할수 있다.

$$L_{q2} = \sum_{i=1}^{2} \frac{w_i \rho_i^2}{1 - \rho_i} = \sum_{i=1}^{2} \frac{w_i \lambda_i^2}{\mu_i (\mu_i - \lambda_i)}$$
 (12)

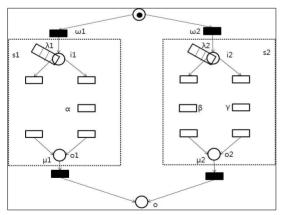


그림 3. 택일조합의 대기행렬 페트리넷 모형

③ 택일조합 시스템에서 평균대기시간을 W_{q2} 라하면, 식 (5)와 식 (10)에 의해 다음과 같이 계산할수 있다.

$$W_{q2} = \sum_{i=1}^{2} \frac{w_i}{\mu_i - \lambda_i} = \sum_{i=1}^{2} \frac{w_i}{\mu_i (1 - \rho_i)}$$
 (13)

Case 3) 비순서순차 조합

비순서순차 조합에서는 웹서비스 S1 다음에 S2가 실행되거나 혹은 S2 다음에 S1이 실행된다. 이러한 상황을 나타내는 페트리넷 모형은 그림 4와 같다. 고객의 수가 각각 n_1 과 n_2 인 두 개의 M/M/1 시스템이 각각 가중값 w_1 과 w_2 를 가진 채 연결되어 있을 경우, 비순서순차 조합 시스템의 고객수 분포는 두 시스템의 평균으로서 다음과 같다. 단, $w_1 + w_2 = 1$ 이다.

$$\begin{split} P_{n1} + P_{n2} &= \frac{1}{2} \sum_{i=1}^{2} w_{i} \left(1 - \frac{\lambda_{i}}{\mu_{i}} \right) \left(\frac{\lambda_{i}}{\mu_{i}} \right)^{n_{i}} \\ &= \frac{1}{2} \sum_{i=1}^{2} w_{i} (1 - \rho_{i}) \rho_{i}^{n_{i}}, (n_{i} \ge 1, i = 1, 2) \end{split} \tag{14}$$

① 비순서순차 조합 시스템에서 임의시점에서의 평균고객수를 L_3 이라 하면, 식 (3)과 식 (14)에 의해 다음과 같이 계산할 수 있다.

$$L_3 = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i \rho_i}{1 - \rho_i} = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i \lambda_i}{\mu_i - \lambda_i}$$
 (15)

② 비순서순차 조합 시스템에서 평균 대기고객수 를 L_{q3} 이라 하면, 식 (4)과 식 (14)에 의해 다음과 같이 계산할 수 있다.

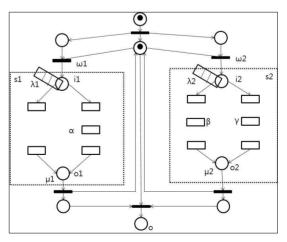


그림 4. 비순서순차조합의 대기행렬 페트리넷모형

$$L_{q3} = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i \rho_i^2}{1 - \rho_i} = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i \lambda_i^2}{\mu_i (\mu_i - \lambda_i)}$$
(16)

③ 비순서순차 조합 시스템에서 평균대기시간을 W_{q3} 이라 하면, 식 (5)와 식 (14)에 의해 다음과 같이 계산할 수 있다.

$$W_{q3} = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i}{\mu_i - \lambda_i} = \frac{1}{2} \sum_{i=1}^{2} \frac{w_i}{\mu_i (1 - \rho_i)}$$
 (17)

Case 4) 반복조합

반복조합에서는 그림 5와 같이 웹서비스 S가 주어 진 횟수만큼 반복 실행된다. 고객의 수가 각각 n_i , $i=1,\cdots,n$ 인 n의 M/M/1 시스템이 독립적으로 연결되어 있을 경우, 반복조합 시스템의 고객수 분포는 시스템이 n번 독립적으로 반복되므로 다음과 같다.

$$kP_{n} = k \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^{n} = k(1 - \rho)\rho^{n}, \tag{18}$$

$$(n \ge 1)$$

① 반복조합 시스템에서 임의시점에서의 평균고 객수를 L_4 라 하면, 식 (3)과 식 (18)에 의해 다음과 같이 계산할 수 있다.

$$L_{4} = \sum_{i=1}^{n} \frac{\rho_{i}}{1 - \rho_{i}} = \sum_{i=1}^{n} \frac{\lambda_{i}}{\mu_{i} - \lambda_{i}}$$
(19)

② 반복조합 시스템에서 평균 대기고객수를 L_{q4} 라 하면, 식 (4)와 식 (18)에 의해 다음과 같이 계산할 수 있다.

$$L_{q4} = \sum_{i=1}^{n} \frac{\rho_i^2}{1 - \rho_i} = \sum_{i=1}^{n} \frac{\lambda_i^2}{\mu_i(\mu_i - \lambda_i)}$$
 (20)

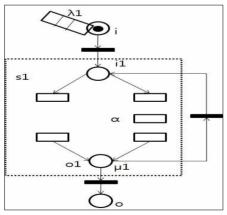


그림 5. 반복조합을 나타내는 페트리넷

③ 반복조합 시스템에서 평균대기시간을 W_{q4} 라하면, 식 (5)와 식 (18)에 의해 다음과 같이 계산할수 있다.

$$W_{q4} = \sum_{i=1}^{n} \frac{1}{\mu_i - \lambda_i} = \sum_{i=1}^{n} \frac{1}{\mu_i (1 - \rho_i)}$$
 (21)

Case 5) 통신이 있는 병렬

"통신이 있는 병렬" 조합에서는 두 웹서비스 S1과 S2가 그림 6처럼 독립적으로 실행하다가 데이터를 주고받는다. 고객의 수가 각각 n_1 과 n_2 인 두 개의 M/M/1 시스템이 서로 데이터 교환을 하기도하면서 병렬로 연결되어 있을 경우, "통신이 있는 병렬" 조합시스템의 고객수 분포는 다음과 같다.

$$P_{n1} + P_{n2} = \sum_{i=1}^{2} I(x, A) \left(1 - \frac{\lambda_i}{\mu_i} \right) \left(\frac{\lambda_i}{\mu_i} \right)^{n_i}$$

$$= \sum_{i=1}^{2} I(x, A) (1 - \rho_i) \rho_i^{n_i}, (n_i \ge 1, i = 1, 2)$$
(22)

여기서, I(x,A)는 지시함수(indicator function)로서, x를 표본공간 A에서 주어진 사상이라할 때 S안의 모든 점에 대하여 다음과 같이 정의된다.

$$I(x,A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$
 (23)

① "통신이 있는 병렬" 조합 시스템에서 임의시점에서의 평균고객수를 L_5 라 하면, 식 (3)과 식 (22)에 의해 다음과 같이 계산할 수 있다.

$$L_{5} = \sum_{i=1}^{2} I(x,A) \frac{\rho_{i}}{1 - \rho_{i}}$$

$$= \sum_{i=1}^{2} I(x,A) \frac{\lambda_{i}}{\mu_{i} - \lambda_{i}}$$
(24)

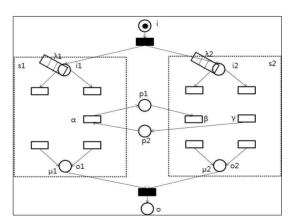


그림 6. "통신이 있는 병렬" 조합의 페트리넷 모형

② "통신이 있는 병렬" 조합 시스템에서 평균 대기 고객수를 L_{q5} 라 하면, 식 (4)와 식 (22)에 의해 다음과 같이 계산할 수 있다.

$$\begin{split} L_{q5} &= \sum_{i=1}^{2} I(x, A) \frac{\rho_{i}^{2}}{1 - \rho_{i}} \\ &= \sum_{i=1}^{2} I(x, A) \frac{\lambda_{i}^{2}}{\mu_{i}(\mu_{i} - \lambda_{i})} \end{split} \tag{25}$$

③ "통신이 있는 병렬" 조합 시스템에서 평균대기 시간을 W_{q5} 라 하면, 식 (5)와 식 (22)에 의해 다음과 같이 계산할 수 있다.

$$W_{q5} = \sum_{i=1}^{2} I(x, A) \frac{1}{\mu_{i} - \lambda_{i}}$$

$$= \sum_{i=1}^{2} I(x, A) \frac{1}{\mu_{i} (1 - \rho_{i})}$$
(26)

Case 6) 식별조합

식별조합은 그림 7에 보이는 바와 같이, 고객의수가 각각 n_i , i=1,2,3인 3개의 M/M/1 시스템이처음 2개의 시스템 중 먼저 끝나는 시스템이 3번째시스템을 작동 시키고, 작동된 3번째 시스템과 처음두 번째 시스템 중 늦게 끝난 시스템이 끝이 나야시스템이 멈추는 경우이다. 식별조합 시스템의 고객수 분포는 시스템이 n번 독립적으로 반복되므로 다음과 같다.

$$\begin{split} &(P_{n1}+P_{n2}\,)P_{n3} \\ &= \Bigg(\sum_{i=1}^2 \bigg(1-\frac{\lambda_i}{\mu_i}\,\bigg)\!\bigg(\frac{\lambda_i}{\mu_i}\bigg)^{n_i}\Bigg) \bigg(1-\frac{\lambda_3}{\mu_3}\,\bigg)\!\bigg(\frac{\lambda_3}{\mu_3}\bigg)^{n_3} \end{split}$$

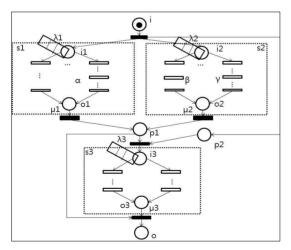


그림 7. 식별조합연산 나타내는 페트리넷모형

$$= \left(\sum_{i=1}^{2} (1 - \rho_i) \rho_i^{n_i}\right) (1 - \rho_3) \rho_3^{n_3}$$

$$(n_i \ge 1, i = 1, 2, 3)$$
(27)

① 식별조합에서 임의시점에서의 평균고객수를 L_6 이라 하면, 식 (3)과 식 (27)에 의해 다음과 같이 계산할 수 있다.

$$L_{6} = \left(\sum_{i=1}^{2} \frac{\rho_{i}}{1 - \rho_{i}}\right) \left(\frac{\rho_{3}}{1 - \rho_{3}}\right)$$

$$= \left(\sum_{i=1}^{2} \frac{\lambda_{i}}{\mu_{i} - \lambda_{i}}\right) \left(\frac{\lambda_{3}}{\mu_{3} - \lambda_{3}}\right)$$
(28)

② 식별조합에서 평균 대기고객수를 L_{q6} 이라 하면, 식 (4)와 식 (27)에 의해 다음과 같이 계산할 수있다.

$$L_{q6} = \left(\sum_{i=1}^{2} \frac{\rho_{i}^{2}}{1 - \rho_{i}}\right) \left(\frac{\rho_{3}^{2}}{1 - \rho_{3}}\right)$$

$$= \left(\sum_{i=1}^{2} \frac{\lambda_{i}^{2}}{\mu_{i}(\mu_{i} - \lambda_{i})}\right) \left(\frac{\lambda_{3}^{2}}{\mu_{3}(\mu_{3} - \lambda_{3})}\right)$$
(29)

③ 식별조합에서 평균대기시간을 W_{q6} 이라 하면, 식 (5)와 식 (27)에 의해 다음과 같이 계산할 수 있다.

$$W_{q6} = \left(\sum_{i=1}^{2} \frac{1}{\mu_{i} - \lambda_{i}}\right) \left(\frac{1}{\mu_{3} - \lambda_{3}}\right)$$

$$= \left(\sum_{i=1}^{2} \frac{1}{\mu_{i} (1 - \rho_{i})}\right) \left(\frac{1}{\mu_{3} (1 - \rho_{3})}\right)$$
(30)

Case 7) 선택조합 선택조합은 그림 8에 보이는 바와 같이 여러 웹서

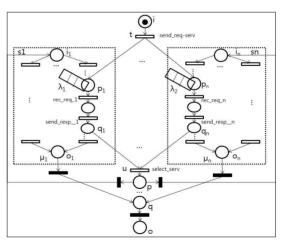


그림 8. 선택조합연산을 나타내는 페트리넷 모형

비스에 작업 의뢰 메시지를 보낸 후 답장을 취합하여 가장 적당한 웹서비스에 작업을 의뢰하는 현상을 나타내는 조합연산이다. 선택조합 시스템은 선택 서버에 도착한 n개의 M/M/1 시스템 가운데, 임의로 1개의 시스템을 선택하여 작동시킴으로 선택조합 시스템의 고객수 분포는 다음과 같다.

$$Rand(P_{n1}, \dots, P_{nn})$$

$$= \left(1 - \frac{\lambda}{\mu}\right) \left(\frac{\lambda}{\mu}\right)^{n_i} = (1 - \rho) \rho^{n_i},$$

$$i = 1, 2, \dots, n$$
(31)

① 선택조합 시스템에서 임의시점에서의 평균고 객수를 L_7 이라 하면, 식 (3)과 식 (31)에 의해 다음과 같이 계산할 수 있다.

$$L_7 = \frac{\rho_i}{1 - \rho_i} = \frac{\lambda_i}{\mu_i - \lambda_i}$$

$$i = 1, 2, \dots, n$$
(32)

② 선택조합 시스템에서 평균 대기고객수를 L_{q7} 이라 하면, 식 (4)와 식 (31)에 의해 다음과 같이 계산할수 있다.

$$L_{q7} = \frac{\rho_i^2}{1 - \rho_i} = \frac{\lambda_i^2}{\mu_i(\mu_i - \lambda_i)}, i = 1, 2, \dots, n$$
 (33)

③ 선택조합 시스템에서 평균대기시간을 W_{q7} 이라 하면, 식 (5)와 식 (31)에 의해 다음과 같이 계산할수 있다.

$$W_{q7} = \frac{1}{\mu_i - \lambda_i} = \frac{1}{\mu_i (1 - \rho_i)}, i = 1, 2, \dots, n$$
 (34)

Case 8) 정제조합

정제조합은 계층페트리넷에서 볼 수 있는 바와 끝이 추상화된 트랜지션을 실제 작업을 실행하는 웹서비스로, 그림 9에 보이는 바와 같이, 대치하는 연산이다. 따라서 고객의 수가 각각 n_i , i=1,2,3인 3개의M/M/1시스템의 관계가 2개시스템의 합으로 하나의 시스템이 이루어지는 경우를 정제조합 시스템이라 하며,이 시스템의 고객수 분포는 두 개시스템의합으로 이루어지므로 다음과 같다.

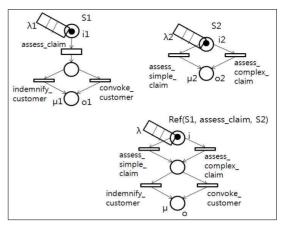


그림 9. 정제조합을 나타내는 페트리넷 모형. S1의 assess_ claim을 S2로 정제함

여기서, 정제는 여러 개의 서비스 중 적당한 서비 스를 택하여 실행한다는 의미이다.

① 정제조합 시스템에서 임의시점에서의 평균고 객수를 $L_{\rm s}$ 이라 하면, 식 (3)과 식 (35)에 의해 다음과 같이 계산할 수 있다.

$$\begin{split} L_8 &= \frac{\rho_1}{1-\rho_1} + \, \, \, \overline{\wedge} \, \, \, \overline{\wedge} \\ & \left(\frac{\rho_2}{1-\rho_2} \, , \, \cdots \, , \, \frac{\rho_n}{1-\rho_n} \, \right) \\ &= \frac{\lambda_1}{\mu_1 - \lambda_1} + \, \, \overline{\wedge} \, \, \, \overline{\wedge} \\ & \left(\frac{\lambda_2}{\mu_2 - \lambda_2} \, , \, \cdots \, , \, \frac{\lambda_n}{\mu_n - \lambda_n} \right) \end{split} \tag{36}$$

② 정제조합 시스템에서 평균 대기고객수를 L_{q8} 이라 하면, 식 (4)와 식 (35)에 의해 다음과 같이 계산할수 있다.

$$\begin{split} L_{q4} &= \frac{\rho_1^2}{1-\rho_1} + \ \text{Re} \ \text{Re} \\ & \left(\frac{\rho_2^2}{1-\rho_2} \ , \cdots \ , \frac{\rho_n^2}{1-\rho_n} \right) \end{split}$$

$$=\frac{\lambda_1^2}{\mu_1(\mu_1-\lambda_1)}+\stackrel{\sim}{\bowtie}\stackrel{\sim}{\bowtie}$$

$$\left(\frac{\lambda_2^2}{\mu_2(\mu_2-\lambda_2)},\cdots,\frac{\lambda_n^2}{\mu_n(\mu_n-\lambda_n)}\right) \tag{37}$$

③ 정제조합 시스템에서 평균대기시간을 W_{q8} 이라 하면, 식(5)와 식(35)에 의해 다음과 같이 계산할수 있다.

4. 시스템 분석 사례

옥내 위치기반 콘텐츠 출력 응용 시스템은 박물 관, 도서관, 백화점과 같은 건물 내부에서 사용자의 현재 위치를 바탕으로 알맞은 콘텐츠를 출력하여주 는 시스템이다. 예를 들어, 박물관 관람자의 현재 위 치가 금제완 전시부스라면 금제완 관련 콘텐츠를 출 력하여 준다. 이러한 옥내 위치기반 콘텐츠 출력 시 스템은 응용영역의 도면을 출력하고, 사용자의 위치 를 측정하고, 현재위치가 어느 부스인가를 파악하고, 부스에 연합된 콘텐츠를 찾아 출력하여 주는 과정을 거친다. 그런데 이러한 과정들은 응용영역이 박물관 이든, 도서관이든 관계없이 모든 옥내 위치기반 콘텐 츠 출력 시스템들이 실행해야할 사항들임으로 웹서 비스로 제공하는 것이 바람직하다. 그래서 본 연구에 서는 도면출력 웹서비스, 옥내측위 웹서비스, 사용자 가 현재 위치하고 있는 부스를 찾아주는 부스 웹서비 스, 부스 아이디로 관련된 콘텐츠의 URI를 찾아 출력 하여주는 웹서비스를 구현하고, 이들을 이용하여 콘 텐츠를 출력하는 시스템을 구현하였다. 구현한 시스 템의 실행 화면 예는 그림 10에 보인다. 이 그림에서 는 사용자의 위치가 금제완 부스임을 판별하고 금제 와 영상을 출력하고 있다.

구현한 옥내 위치기반 콘텐츠 출력 응용 시스템을 본 논문이 제안하는 대기행렬 페트리넷으로 표현하

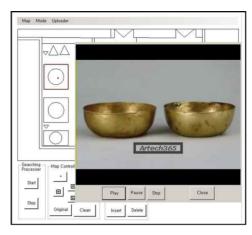


그림 10. 위치기반 콘텐츠 출력 시스템 실행 화면 예

면 그림 11과 같다. 처리과정은 위에서 설명한 바와 동일하며, 네 개의 웹서비스들이 순차조합으로 연결되었다. 또한 "rendering drawing", "location", "booth?" 및 "content display"는 각각 그림 12부터 그림 15가 정제조합된 것이다. 응용시스템의 고객은 이 시스템의 사용자이고, 각 웹서비스의 고객은 그

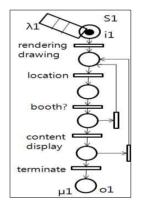


그림 11. 옥내위치기반 콘텐츠 출력 시스템의 페트리넷 모형

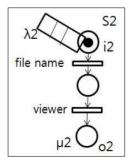
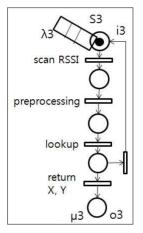


그림 12. 도면출력 웹서비스

서비스를 호출하는 여러 응용시스템들이다. 그림 11 부터 그림 15까지 각각의 출생률과 사멸률을 각각 λ_i , μ_i , i=1 , \cdots , 5 라고 하면 정제조합에 의한 응용시스템의 평균고객수는 $L=\frac{\lambda_1}{\mu_1-\lambda_1}+$ 정제 $\left(\frac{\lambda_2}{\mu_2-\lambda_2}$, \cdots , $\frac{\lambda_5}{\mu_5-\lambda_5}\right)$ 이고, 평균대기시간은 $W=\frac{1}{\mu_1(1-\rho_1)}$ + 정제 $\left(\frac{1}{\mu_2(1-\rho_2)}$, \cdots , $\frac{1}{\mu_5(1-\rho_5)}\right)$ 가 된다.

예를 들어, 그림 11부터 그림 15까지 Poisson 과정에 따른 각각의 출생률과 사멸률을 참고문헌 [10]의 자료를 응용하여 표 1이라고 하면, 각 응용시스템의 고객수와 대기시간은 표 2가 되고 정제조합에 의하여 응용시스템의 평균고객수와 대기시간은 각각



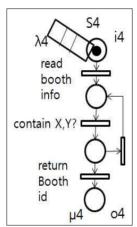


그림 13. 옥내측위 웹서비스

그림 14. 부스 판별 웹서비스

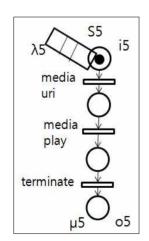


그림 15. 콘텐츠 출력 웹서비스

표 1. Poisson과정에 따른 각 시스템의 출생률(λ)과 사멸률(μ) 의 예

i	1	2	3	4	5
λ_i	0.22	0.13	0.27	0.17	0.25
μ_i	0.81	0.75	0.85	0.79	0.89

표 2. 표 1의 자료를 사용했을 때 각 응용시스템의 고객수와 대기시간

i	1	2	3	4	5
고 객 수	0.3729	0.2079	0.4655	0.2742	0.3906
대기시간	1.6949	1.6129	1.7141	1.6129	1.5625

1.7129 와 8.1973 이 된다.

이와 같은 분석의 실행 시간을 살펴보면, 웹서비스를 나타내는 페트리 넷 (그림 12부터 15) 각각을 분석하기 위하여 해당 페트리 넷의 구성요소 수만큼의 사칙연산을 수행하였고, 최종적으로 그림 11의 페트리 넷을 비슷한 방법으로 분석하여 시스템 전체를 분석하였다. 즉, 분석에 필요한 사칙연산의 수는 페트리 넷을 구성하는 요소의 수에 비례한다.

5. 결 론

웹 응용 시스템 분석을 위한 다양한 페트리넷 모형이 제안된 바 있다. 그러나 대부분의 기존의 연구는 시스템이 교착상태 없이 실행되는지를 보이는데 중점을 두고 있다. 교착상태가 없어야 하는 것은 시스템이 충족해야할 기본적인 기능적 조건이지만, 나아가서 시스템 실행 시간이 사용자가 요구한 시간이 내라야 하는 것도 반드시 검증되어야할 사항 중 하나이다. 그럼에도 불구하고 웹서비스를 이용한 응용시스템의 실행시간 분석에 대한 연구는 미미한 수준이다. [8]이 웹서비스 시스템의 시간분석을 다루었는데이 방법은 모든 도달 가능한 상태를 나열함으로써실행시간을 구하기 때문에 페트리넷의 크기가 증가함에 따라 분석에 필요한 시간이 폭발적으로 증가하는 단점이 있다.

이러한 기존의 방법의 단점을 극복하기 위하여 본 논문은 대기행렬을 이용하여 웹 응용 시스템의 처리 시간을 분석하기 위한 페트리넷을 제안한다. 즉, 제 안한 방법은 [5-7]에 소개된 페트리넷 표현에 대기행 렬을 추가함으로써 교착상태뿐 아니라 시간분석도 가능하도록 확장하였다. 또한 [8,9]에 제안된 시간 분 석 방법이 모든 도달 상태를 나열함으로써 실행에 소요되는 시간이 페트리넷 구성 요소의 수를 지수로 하는 식이 되는 단점을 보이는데 반하여 본 논문은 시스템을 구성하는 각 웹서비스의 대기행렬 분석결 과를 통합하는 연산을 제안함으로써 응용시스템의 대기행렬 분석 시간이 구성요소의 수에 비례한다.

옥내위치기반 콘텐츠 출력 시스템은 사용자의 위치를 고려하여 가장 적당한 콘텐츠를 출력하여주는 시스템이다. 이러한 응용시스템의 예로 박물관 관람객에게 현재 관람중인 전시물 관련 콘텐츠를 출력하여 주는 서비스를 들 수 있다. 옥내위치기반 콘텐츠 출력 시스템은 일반적으로 네 개의 웹서비스를 조합하여 구축한다. 본 논문은 박물관 관람객을 위한 위치기반 콘텐츠 출력 응용시스템을 제안한 페트리넷으로 분석한 결과를 제공하여 본 논문이 제안하는 대기행렬 페트리넷 방법의 타당성을 보인다.

참 고 문 헌

- [1] Tadao Murata, Jaegeol Yim, Huabei Yin, and Ouri Wolfson, "Petri-Net Model and Minimum Cycle Time for Updating Moving Objects Database," *International Journal of Computer Systems Science & Engineering*, Vol.21, No.3, pp. 211–217, 2006.
- [2] Jaegeol Yim, Jae-Hun Joo, and Young-Chan Lee, "Timed Net With Choice Probability and Its Minimum Cycle Time-The Case of Location Based Service," *Expert Systems With Applications* Vol.31, No.2, pp. 221–230. 2006.
- [3] M.P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing," *Communications ACM*, Vol.46, No.10, pp. 25–28, 2003.
- [4] J. Zhang, C.K. Chang, L.J. Zhang, and P.C.K.

- Hung, "Toward a Service-Oriented Development Through a Case Study," *IEEE Trans. Syst., Man, Cyber. A, Syst. Humans*, Vol.37, No.6, pp. 955–969, 2007.
- [5] R. Hamadi and B. Benatallah, "A Petri Net-based Model for Web Service Composition," 14th Australasian database conference, Vol. 17, pp. 191–200, 2003.
- [6] P. Xiong, Y. Fan, and M. Zhou, "A Petri Net Approach to Analysis and Composition of Web Services," *IEEE Transactions on Sys*tems, Man and Cybernetics, Part A: Systems and Humans, Vol.40, Issue 2, pp. 376–387, 2010.
- [7] Y. Han, C. Jiang, and X. Luo, "Modeling and Analysis of Semantic Web Services with Petri Nets," IEEE Third International Conference on Semantics, *Knowledge and Grid*, pp. 122–127, 2007.
- [8] Y. Han and X. Luo, "Composition and Reduction of Web Service Based on Dynamic Timed Colored Petri Nets," *IEEE Internatio*nal Symposium on Parallel and Distributed Processing with Applications, pp. 659–663, 2009.
- [9] H. Yin, Z. Tan, C. Lin, Y. Jiang, and F. Qiu, "Modeling Mobile Web Services Flow Using Generalized Stochastic High-Level Petri Net," IEEE International Conference on Systems, Man and Cybernetics, Vol.5, pp. 4953-4958, 2004.
- [10] 김종호, 이기성, 확률과정입문, 2010, 자유아카 데미.
- [11] 이호우, 대기행렬이론 확률과정론적 분석, 시 그마프레스. 1998.



심 규 박

1986년 동국대학교 대학원 통계 학과 이학석사

1993년 동국대학교 대학원 통계 학과 이학박사

1994년~현재 동국대학교 과학기 술대학 정보통계학과 교수

관심분야: 전산통계, 신뢰도검정, 통계자료분석.



임 재 걸

1981년 동국대학교 전자계산학과 졸업

1987년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 석사

1990년 일리노이대학교 시카고 캠퍼스 컴퓨터과학 박사

1992년~현재 동국대학교 과학기술대학 컴퓨터공학부 교수

관심분야: 시스템 설계 및 분석, 인공지능, 페트리 넷 이 론 및 응용.



이 계 영

1980년 동국대학교 전자계산학과 졸업

1983년 동국대학교 대학원 전자 계산학과 석사

1991년 단국대학교 대학원 전자 공학과(컴퓨터공학 전공) 전자공학 박사

1985년~현재 동국대학교 과학기술대학 컴퓨터공학부 교수

관심분야: 운영체제이론, 멀티미디어 이론, IPTV 이론.



김 진 석

1992년 서울대학교 계산통계학과 졸업

1994년 서울대학교 대학원 계산 통계학과(통계전공) 석사 2003년 서울대학교 대학원 통계

학과 이학 박사

2007년~현재 동국대학교 과학기술대학 정보통계학과 조교수

관심분야: 데이타마이닝, 통계계산, 네트워크자료분석