

상황인지 컴퓨팅 환경에서 복합서비스를 제공하는 서비스시스템의 성능분석

남진규^{1*} · 허 선² · 주국선² · 신동민²

¹안보경영연구원 / ²한양대학교 정보경영공학과

Performance Analysis of a Composite Service Providing System in a Context-Aware Computing Environment

Jingyu Nam¹ · Sun Hur² · Kuksun Joo² · Dongmin Shin²

¹Security Management Institute, Seoul, 135-871, Korea

²Department of Information and Industrial Engineering, Hanyang University, Ansan 426-791, Korea

There are a variety of users and devices in a context-aware computing environment. In this environment, the service provided to a user may be the composition of diverse services rather than one independent service. Before user's devices provide the composite service to the user, they should perceive user's needs by gathering related information segments from other surrounding devices and/or sensors. We consider a context-aware computing environment providing composite and adapted service to users and propose an information processing model that characterizes the device where the collected data should be processed through services and/or applications. Based on this model, we provide an analytical tool to obtain some performance measures of the context-aware computing environment.

Keywords: Analytical Tool, Composite Service, Service System, Performance Measure, Context-Aware Computing Environment

1. 서론

분산 시스템과 모바일 컴퓨팅 기술의 발전으로 컴퓨팅 기기는 사용자의 주변 환경에 스며들어 운용될 수 있을 만큼 점점 작아지고 강력한 성능을 지니게 되었다. 이 기기들은 컴퓨팅 능력과 통신 기능을 이용하여 사용자 주변에 존재하면서 적합한 액션 또는 서비스를 사용자에게 제공하는 상황인지 서비스 제공 환경을 조성한다(Satyanarayanan, 2001). 이러한 환경에서 컴퓨팅 기기는 사용자의 현재 상태나 주변 환경에 관한 정보를 자동으로 인지하고 이에 적합한 서비스를 사용자에게 제공할 수 있어야 한다.

한 사용자가 다른 사용자와 의사소통을 하는 경우, 두 사용자는 서로의 함축적인 상황 정보를 대화에 이용할 수 있다.

예를 들어, 두 사용자가 회의나 강의 중에 있다면 소리를 내어 대화를 하는 것이 어렵다는 것을 알고 있으므로, 간단한 메모를 통해 서로 의사를 전달할 수 있다. 그러나 기기와 사용자간의 상호작용에서는 사용자의 주변 상황을 기기에게 인지시키는 것이 쉽지 않다. 이러한 문제점을 극복하기 위해 인간과 컴퓨터의 상호작용의 관점에서 상황인식에 대한 많은 연구들이 이루어져 왔다(Dey and Abowd, 2000).

상황(context)은 사용자와 서비스를 제공하는 어플리케이션 사이에 존재하는 모든 개체의 상태나 주변 환경에 관한 정보를 의미하는 것으로 개체는 사람이나 물건, 특정 장소 등이 될 수 있다(Dey and Abowd, 2000). 상황은 사용자 주변 환경에 관한 물리적인 정보들을 얻음으로써 형성될 수 있는데, 이 정보들은 비용이 저렴하면서 널리 퍼져있는 센서들과 PDA나 휴대

본 논문은 2008년 추계 산업공학 학술대회 서비스 사이언스 특별세션에서 발표된 우수 논문입니다.

* 연락저자 : 남진규, 135-871, 서울특별시 강남구 삼성동 79-1 동흥빌딩 9층, Tel 02-544-5938, Fax 02-544-5939, E-mail : jgnam@hanyang.ac.kr
2008년 12월 18일 접수; 2009년 1월 19일 수정본 접수; 2009년 1월 21일 게재 확정.

전화와 같은 기기들을 통해 수집된 데이터를 기반으로 형성된다(Schilit and Theimer, 1994; Schilit *et al.*, 1994; Moran and Dourish, 2001). 이 상황 정보는 사용자에게 적합한 맞춤형 서비스를 제공할 때 사용되는 데, 이것은 상황인지 컴퓨팅(context-aware computing)의 중요한 특징이다(Dey and Abowd, 2000; Dey, 2001).

이러한 컴퓨팅 환경에서 사용자가 지닌 기기들은 사용자의 상황 정보와 관련된 데이터를 다른 기기나 센서들과의 커뮤니케이션을 통해 얻을 수 있다. 이러한 데이터는 개별적으로 의미가 없거나 부족하지만 여러 개가 모여 사용자의 상황과 관련된 하나의 정보를 형성한다. 이렇게 생성된 정보는 사용자에게 적합한 서비스를 제공하는 데 이용될 수 있다. 그러나 사용자의 상태나 주변 상황은 시간이 지남에 따라 변화할 수 있으므로 수집되어 있던 데이터 중 일부는 그 효용성이 소멸될 수도 있다(Hur *et al.*, 2007). 예를 들어, 날씨에 관련된 데이터는 시간이 지남에 따라 새로 업데이트되며 이전의 날씨 데이터는 불필요해진다. 그래서 기기는 사용자의 상황 정보를 생성할 때까지 데이터가 수집되고 소멸되는 과정을 거치게 된다. 이 과정을 거쳐 기기가 사용자의 상황과 관련된 데이터를 일정한 양 이상 모두 수집하면 그 데이터에서 사용자의 상태나 주변 상황을 나타내는 하나의 정보를 형성하게 된다.

주변 기기에서 수집된 데이터를 바탕으로 형성된 하나의 정보(이를 정보객체라고 하자)는 해당 데이터를 수집한 사용자의 다양한 요구에 맞는 서비스를 제공하는 데에 이용된다. 또한 대부분의 경우 한 사용자가 원하는 서비스는 단순한 알람 기능처럼 한 가지일 수도 있지만 일정 통보와 그 일정에 맞는 교통서비스와 같이 여러 서비스를 이어서 받는 복합적인 형태일 수 있다. 이 경우에는 여러 기기가 서로 협업하여 사용자에게 서비스를 제공한다(Baldauf *et al.*, 2007). 예를 들어, 사용자가 회사에서 회의를 마친 경우 스마트 PDA는 회의를 통해 사용자에게 주어진 업무를 파악하여 노트북의 일정 관리 프로그램에 업로드 시키고, 노트북은 업무에 필요한 정보를 검색하여 그 결과를 프린터를 통해 인쇄하도록 할 수 있다. 여기서 PDA가 음성인식 기능이나 다른 기기들을 통하여 사용자의 상황을 파악하는 것이 데이터를 수집하여 하나의 정보객체를 형성하는 과정이 되고, 업무를 업로드하고 관련 정보를 검색하는 것 등이 복합 서비스가 된다.

이러한 환경에서 각 사용자의 서비스에 대한 요구사항은 다양하므로, 사용자의 상황과 관련된 정보객체가 형성될 때까지 시간과 단순 또는 복합적인 서비스가 사용자에게 제공될 때까지 시간에 대한 요구 역시 각 사용자마다 다를 수 있다. 또한 제한된 성능을 지닌 기기들은 각 사용자의 요구를 한번에 하나씩 순차적으로 처리하게 되므로 사용자의 상황과 관련된 정보 객체는 이전에 축적된 정보 객체에 의해 진행되고 있는 서비스가 완료될 때까지 기다리게 된다(Hur *et al.*, 2007).

이러한 상황인지 컴퓨팅을 실현하는 서비스시스템에 대해서는 많은 연구들이 진행되고 있다. 이 연구들에서 제안되고

있는 서비스시스템의 구조(architecture)는 대부분 다음과 같다(Baldauf *et al.*, 2007; Schmidt *et al.*, 1999; Strang and Linnhoff-Popien, 2004; Schmidt, 2006). 서비스시스템은 센서 네트워크, 기기의 상태, 사용자 프로파일 등의 여러 소스들을 통해 개별적인 데이터를 수집한 후 사용자의 상황이나 주변 환경에 관한 정보를 추론한다. 서비스시스템은 이렇게 추론된 정보를 기반으로 사용자에게 적합한 서비스를 제공한다(Hur *et al.*, 2007). 이러한 연구들은 상황인지 컴퓨팅의 시스템적인 구조와 컴퓨팅 기기가 사용자의 환경을 인식하는데 필요한 상황모형(context model)을 설계하는 데 집중되어 있다(Baldauf *et al.*, 2007; Schmidt *et al.*, 1999; Strang and Linnhoff-Popien, 2004; Schmidt, 2006). 그러나 수집되는 데이터의 불확실성을 고려한 서비스시스템에 대한 연구는 미비한 실정이며, 상황인지 컴퓨팅에서 불확실성을 고려한 연구들은 대부분 수집된 데이터에서 사용자의 상황을 어떻게 정확하게 추론하는가에 중점을 두고 있다.

본 연구에서는 컴퓨팅 기기가 사용자의 상황을 인식할 때 발생하는 불확실성을 고려한 서비스시스템을 모형화하고 수리적으로 분석하여 다양한 성능척도를 제안한다. 또한 시뮬레이션과 비교하여 제안된 방법이 유용함을 검증한다. 이는 서비스시스템의 사전 분석과 이를 바탕으로 효과적인 시스템을 설계하는 데에 매우 중요한 역할을 차지한다. 특히 현재 정성적인 측면에서의 서비스사이언스 연구 분야에 정량적이고 수리적인 접근을 통해 보다 과학적이고 체계적인 서비스 모형의 개발에 기여할 것으로 기대된다.

본 연구의 구성은 다음과 같다. 제 2장에서는 사용자의 상황 정보를 처리하는 모형이 대기 네트워크로 구성될 수 있음을 설명하고, 제 3장에서는 사용자의 상태 또는 주변 상황 정보가 생성될 때까지의 시간의 확률분포함수를 구하는 방법을 제시한다. 제 4장에서는 제안된 모형의 성능척도를 유도하는 변형 평균치 분석방법을 설명한다. 제 5장은 제안된 방법과 시뮬레이션 결과를 비교 분석하고 제 6장에서는 결론과 추후 연구과제에 대해 설명한다.

2. 복합서비스를 제공하는 서비스시스템

앞서 언급한 대로, 상황인지 서비스 제공 환경에서는 많은 사용자들과 다양한 기기들이 존재하며, 각 사용자의 상태나 주변 상황과 관련된 정보 객체는 여러 기기들과 센서들을 통해 수집된 데이터에 의해 생성된다. <Figure 1>은 이러한 상황인지 서비스 제공 시스템의 한 응용 예제를 나타낸다. 이 시스템에서 헬스케어 기기는 두 사용자의 건강 상태에 대한 이상 징후 데이터를 수집한다. 이때 시간이 지남에 따라 사용자의 건강 상태는 변화할 수 있으므로 수집된 사용자의 건강 상태 데이터는 의미를 상실할 수 있다. 기기는 이 데이터 수집 과정을 거쳐 이상 징후에 대한 정보 객체가 생성되면 사용자의 건강

에 악화되었음을 파악하고 구급차와 병원에 연락한다. 그러면 구급차는 환자를 이송하는 서비스를 제공하고 병원은 환자의 건강 상태에 맞는 의료 서비스를 제공한다. 그러나 이 환경에서는 두 사용자의 상황 정보 객체들이 복합적인 서비스를 제공 받으려고 할 수 있으므로, 각 객체는 앞선 객체가 서비스를 완료 받을 때까지 해당 서비스를 기다리게 된다. 여기서 복합적인 서비스는 구급차의 이송 서비스와 병원의 의료서비스가 조합된 것으로 객체들은 각각의 서비스들을 하나씩 받는 것으로 볼 수 있다.

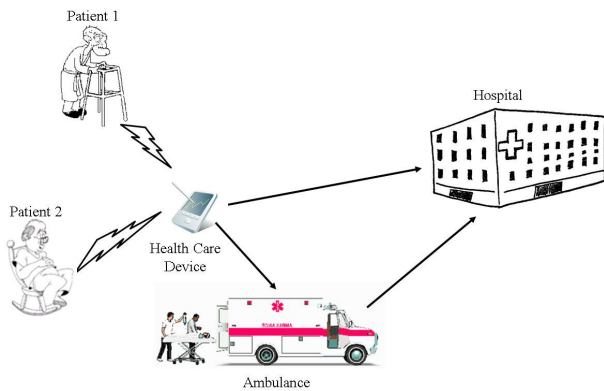


Figure 1. Context-aware computing environment

<Figure 2>는 서비스시스템을 샘플경로로 나타낸 것이다. 여기서 형태 1 사용자에게 대한 데이터가 l_1 개까지 수집되어 형태 1 사용자의 정보 객체가 먼저 형성된다. 그러면 형태 1 사용자는 두 가지의 서비스로 이루어진 복합적인 서비스를 제공받는다. 이후 형태 1 사용자가 두 번째 서비스를 받고 있을 때 형태 2 사용자에게 대한 데이터가 l_2 만큼 쌓이게 되면 형태 2 사용자는 첫 번째 서비스를 받게 되고 형태 1 사용자의 두 번째 서비스가 완료된 이후에 형태 2 사용자는 두 번째 서비스를 제공받는다.

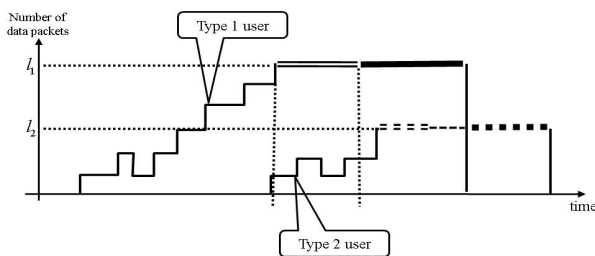


Figure 2. Sample path of a service system

이것을 대기행렬 네트워크로 나타내면 <Figure 3>와 같다. <Figure 3>에서 노드 0의 각 서버의 서비스 시간은 서로 다른 종류의 데이터가 쌓여서 각각 정보객체로써 완성되는데 걸리는 시간으로 볼 수 있으며, 노드 1과 노드 2에 존재하는 각 서

버의 서비스 시간은 각 정보 객체가 두 가지 서비스로 이루어진 복합적인 서비스의 첫 번째와 두 번째 서비스 시간으로 볼 수 있다.

본 논문에서는 보다 명료한 정보 제공 시스템 운영을 제안하기 위하여 두 명의 사용자와 두 종류의 서비스가 존재하는 상황인지 서비스시스템을 다음과 같이 모형화한다.

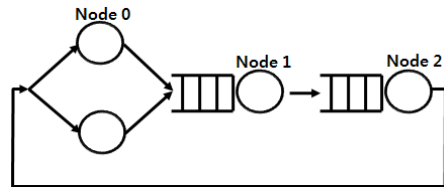


Figure 3. Closed tandem queue network

- (1) 두 가지 형태의 데이터는 서로 독립이며 각각 모수가 λ_1, λ_2 인 포아송 과정으로 도착한다.
- (2) 두 형태의 데이터가 시간의 흐름에 따라 의미를 상실 (obsolescence)하는 시간은 평균이 각각 $1/\alpha_1$ 과 $1/\alpha_2$ 인 지수 분포를 따른다. 그러므로 노드 1에서 각 기기의 서비스 시간은 데이터가 수집되거나 의미를 상실하는 과정을 거쳐 하나의 서비스 제공에 필요한 상황 정보객체가 형성될 때까지 걸리는 시간을 나타낸다.
- (3) 복합서비스를 구성하는 서비스는 두 종류가 존재하며 서로 독립이다.
- (4) 형태 $r(r=1, 2)$ 의 정보객체에 대한 $i(i=1, 2)$ 번째 서비스는 평균이 $1/\mu_{ir}$ 인 지수분포이다.

3. 정보 완성에 걸리는 시간의 분포

사용자의 상태나 주변 상황을 나타내는 정보 객체는 정해진 양의 데이터가 수집됨으로써 생성 될 수 있다. 또한 데이터는 시간이 지남에 따라 사용자의 상태와 관련된 의미를 잃어버릴 수 있으므로 그 수집된 데이터의 개수가 줄어드는 경우가 발생할 수 있다. 각 형태의 정보 객체 하나가 완성될 때까지 시간 $T_0^{(r)}(r=1, 2)$ 의 확률분포를 구하기 위해 다음과 같이 출생률 λ_k 과 사멸률 μ_k 를 가지는 출생사멸과정 $\{N(t), t \geq 0\}$ 를 고려하자.

$$\lambda_k = \begin{cases} \lambda, & k=0, 1, \dots, l-1 \\ 0, & \text{otherwise} \end{cases},$$

$$\mu_k = \begin{cases} k\alpha, & k=1, 2, \dots, l-1, \\ 0, & \text{otherwise.} \end{cases}$$

다음 <Figure 4>은 확률과정 $\{N(t), t \geq 0\}$ 의 전이율 다이어그램을 나타낸 것이다.

정리 1 : 출생사멸과정 $\{N(t), t \geq 0\}$ 에서 상태 $k(k=0, 1, \dots, l-1)$ 에서 출발하여 상태 l 로 흡수될 때까지의 시간 T_k 의 LST $\Phi_k(\theta)$ 는 다음과 같다.

$$\Phi_k(\theta) = \frac{b_k \cdots b_{l-1}}{B_k \cdots B_{l-1}} = \prod_{j=k}^{l-1} \frac{b_j}{B_j}, k=0, 1, 2, \dots, l-1, (1)$$

여기서, $b_k = \frac{\lambda}{\theta + \lambda + k\alpha}$, $k=0, 1, \dots, l-1$,

$$c_k = \frac{k\alpha}{\theta + \lambda + k\alpha}, k=0, 1, \dots, l-1,$$

$$A_k = b_k c_k, B_k = 1 - \frac{A_k}{B_{k-1}}, B_0 = 1.$$

증명 : 확률변수 X 를 평균이 $1/\lambda$ 인 지수분포라 하면 다음이 성립한다.

$$T_0 = X + T_1$$

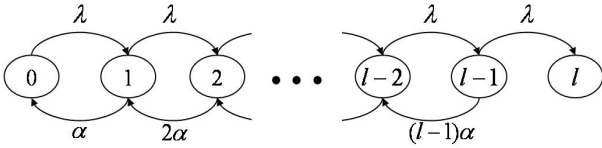


Figure 4. Rate-flow diagram of $\{N(t), t \geq 0\}$

이를 LST의 형태로 표현하면 다음 식과 같다.

$$\Phi_0(\theta) = \frac{\lambda}{\theta + \lambda} \Phi_1(\theta)$$

T_1 은 데이터가 하나 더 쌓이는 경우와 한 개의 데이터가 의미를 상실하여 손실되는 경우로 나누어 볼 수 있다. <Figure 3>에서 전자의 경우는 상태 1에서 상태 2로 전이하는 경우이고, 후자의 경우는 상태 1에서 상태 0으로 전이하는 경우이다. 이는 LST의 형태로 다음 식으로 표현할 수 있다.

$$\Phi_1(\theta) = \frac{\lambda}{\theta + \lambda + \alpha} \Phi_2(\theta) + \frac{\alpha}{\theta + \lambda + \alpha} \Phi_0(\theta).$$

T_k , $k \geq 2$ 인 경우도 위와 같이 <Figure 4>의 전이율 다이어그램과 첫 단계 분석법을 이용하여 다음과 같은 시스템 방정식을 구할 수 있다.

$$\Phi_0(\theta) = b_0 \Phi_1(\theta),$$

$$\Phi_k(\theta) = b_k \Phi_{k+1}(\theta) + c_k \Phi_{k-1}(\theta), k=1, 2, \dots, n-1,$$

$$\Phi_n(\theta) = 1.$$

위 식을 다시 정리하면, 다음과 같이 나타낼 수 있다.

$$\Phi_0(\theta) = b_0 \Phi_1(\theta),$$

$$\Phi_k(\theta) = \frac{b_k}{B_k} \Phi_{k+1}(\theta), k=1, 2, \dots, n-1,$$

$$\Phi_n(\theta) = 1.$$

여기서, $A_k = b_{k-1} c_k$, $B_k = 1 - \frac{A_k}{B_{k-1}}$, $B_0 = 1$.

다음으로 $\Phi_k(\theta)$ 에 $\Phi_{k+1}(\theta)$ 를 순차적으로 대입하면 $\Phi_k(\theta)$, $k=0, 1, \dots, l-1$ 의 식 (1)을 얻을 수 있다. \square

따름정리 2 : T_0 의 LST $\Phi_0(\theta)$ 는 다음과 같다.

$$\Phi_0(\theta) = \frac{b_0 b_1 \cdots b_{n-1}}{B_0 B_1 \cdots B_{n-1}} = \prod_{k=0}^{n-1} \frac{b_k}{B_k}, (2)$$

단, $A_k = b_{k-1} c_k$, $B_k = 1 - \frac{A_k}{B_{k-1}}$, $B_0 = 1$. \square

위의 출생사멸과정에서 λ 대신 λ_r ($r=1, 2$), α 대신, α_r ($r=1, 2$) 그리고 l 대신에 l_r ($r=1, 2$)을 사용하면 이는 각 형태 r 의 데이터가 시스템에 도착하고 l_r 개 쌓여서 형태 r 의 정보객체가 구성되는 과정을 나타내게 된다. 특히 여기서 상태 k 에서 상태 l_r 로 흡수될 때까지의 시간 $T_k^{(r)}$ 는 형태 r 의 데이터가 현재 k 개 쌓여 있는 때부터 형태 r 의 정보 한 개가 완성될 때까지의 시간을 의미한다.

이제 LST $\Phi_0^{(r)}(\theta)$ 를 역변환하면 $T_0^{(r)}$ 의 확률밀도함수와 분포함수 $f_{T_0^{(r)}}(t)$, $F_{T_0^{(r)}}(t)$ 를 구할 수 있다. 그러므로 사용자의 상황을 파악하는데 필요한 데이터의 수가 l_r 일 때, 정보객체를 생성하는데 걸리는 시간의 확률적 특성을 알아 낼 수 있다. 이것은 <Figure 2>의 노드 1에 있는 서버들의 서비스 시간의 분포로 사용된다.

4. 성능분석을 위한 근사적 알고리즘

제안된 모형은 <Figure 3>처럼 무한서버 형태의 한 노드와 두 개의 $\bullet/M/1/FCFS$ 형태의 노드들로 이루어진 닫힌 대기행렬 네트워크로 나타낼 수 있다. BCMP 정리에 따르면, 대기 네트워크는 다음과 같은 가정들을 만족하는 경우 곱 형태의 해를 가지는 대기 네트워크(product-form queueing network)로 볼 수 있다(Bolch *et al.*, 1998).

1. 무한서버 형태의 노드에 있는 각 서버의 서비스 시간의 p.d.f가 콕시안 계열의 LST, 즉, $T_0^{(r)}$, $r=1, 2$ 의 p.d.f가 유리함수 형태의 LST로 나타낼 수 있는 경우
2. $\bullet/M/m/FCFS$ 형태의 서비스 시간은 각 고객의 클래스와는 독립이어야 함(고객은 정보객체, 클래스는 정보 객체의 형태).

곱 형태의 해를 가지는 닫힌 대기 네트워크는 안정상태방정

식으로 표현될 수 있으며 안정상태확률을 계산하는 많은 알고리즘들이 알려져 있는데, 가장 유용한 방법 중의 하나가 평균치 분석(mean value analysis : MVA)이다. 이 방법론의 장점은 정규화 상수를 구할 필요 없이 직접적으로 대기 네트워크의 성능 척도들을 구할 수 있다는 것이다(Bolch *et al.*, 1998).

그러나 제안된 모델은 위의 가정에서 첫 번째는 만족하지만, 두 번째는 만족하지 않는다. 사용자들의 상황에 따라 생성된 각 정보객체는 서비스를 받을 때 서로 다른 서비스율로 받을 수 있으므로, $\bullet/M/m/FCFS$ 형태 노드들의 서비스율은 객체 형태에 따라 변화할 수 있다. 그러므로 본 연구에서는 $\bullet/M/m/FCFS$ 형태의 노드에서 서비스율이 고객 클래스마다 다른 경우에 성능척도들을 근사적으로 구할 수 있는 변형 평균치 분석 방법을 이용한다.

먼저, 다음과 같은 기호들을 정의한다.

κ_{ir} : 노드 i 에 존재하는 형태 r 정보객체의 수,

$$\sum_{i=1}^3 \sum_{r=1}^2 \kappa_{ir} = 2,$$

k_r : 네트워크에 존재하는 형태 r 정보 객체의 수, $r=1, 2$,

$$\text{즉 } k_r = \kappa_{1r} + \kappa_{2r}.$$

$\mathbf{k} = (k_1, k_2)$ (이를 인구벡터라 하자),

\mathbf{K} : 모든 형태의 정보 객체의 수, 즉, $\mathbf{K} = (1, 1)$,

μ_{ir} : 형태 r 정보객체의 노드 i 에서 서비스율,

$p_{ir,js}$: 노드 i 번째에 있는 형태 r 의 정보 객체가 노드 j 번째에 형태 s 의 객체로 전이될 확률,

e_{ir} : 노드 i 번째에 형태 r 의 정보 객체가 방문하는 평균 횟수,

$$e_{ir} = \sum_{j=0}^2 \sum_{s=1}^2 e_{js} p_{js,ir}, i=0, 1, 2, r=1, 2,$$

$A(\mathbf{k})$: 정보객체 수 벡터가 \mathbf{k} 일 때, 정보 객체의 전체 처리량,

$\bar{K}_{ir}(\mathbf{k})$: 정보객체 수 벡터가 \mathbf{k} 일 때, 노드 i 에서 형태 r 정보 객체의 평균 수,

$\bar{T}_{ir}(\mathbf{k})$: 정보객체 수 벡터가 \mathbf{k} 일 때, 노드 i 에서 형태 r 의 정보 객체의 평균체제시간.

<Figure 3>와 같은 닫힌 일렬 대기네트워크에서는 정보 객체가 한 노드에서 다른 노드로 이동할 때 객체의 형태가 변화하지 않는다. 그러므로 $j=i+1, r=s$ 인 경우에는 $p_{ir,js} = 1$ 이고, 다른 경우에는 $p_{ir,js} = 0$ 이다. 이를 이용하면 방문율 e_{ir} 은 다음과 같다.

$$e_{ir} = 1 (i=0, 1, 2, r=1, 2)$$

변형 평균치 분석 알고리즘은 다음과 같다.

Initialize $\bar{K}_{ir}(\mathbf{0}) = 0, i=0, 1, 2, r=1, 2$

Iteration: $\mathbf{k} = \mathbf{0}, \dots, \mathbf{K}$

For nodes 1 and 2 ($\bullet/M/1/FCFS$ nodes), compute

$$\bar{T}_{ir}(\mathbf{k}) = \frac{1}{\mu_{ir}} + \sum_{i=1}^3 \frac{1}{\mu_{ir}} \bar{K}_{ir}(\mathbf{k} - \mathbf{1}_r), i=1, 2, r=1, 2.$$

For node 0, compute $\bar{T}_{0r}(\mathbf{k}) = \frac{1}{\mu_{0r}}, r=1, 2$

$$\text{Compute } \lambda_r(\mathbf{k}) = \frac{k_r}{\sum_{i=1}^3 \bar{T}_{ir}(\mathbf{k})}, r=1, 2$$

Compute

$$\bar{K}_{ir}(\mathbf{k}) = \lambda_i(\mathbf{k}) \bar{T}_{ir}(\mathbf{k}), i=0, 1, 2, r=1, 2$$

단, $(\mathbf{k} - \mathbf{1}_r)$ 은 형태 r 의 정보 객체가 하나 적은 인구벡터를 의미한다. 무한 서버 형태 노드의 서비스율 μ_{ir} 은 제 3장에서 유도된 $T_0^{(r)}$ 의 평균에 역수를 취하여 구할 수 있다.

위의 알고리즘을 이용하여, $A_r(\mathbf{K}), \bar{K}_i(\mathbf{K}), \bar{T}_{ir}(\mathbf{K})$ 와 같은 성능척도를 유도할 수 있다. $A_r(\mathbf{K})$ 는 안정상태에서 각 노드에서 이탈하는 형태 r 의 정보객체의 비율을 의미하는 것으로 단위 시간당 기기가 처리하는 정보객체의 수를 나타낸다. $A_r(\mathbf{K})$ 와 μ_{ir} 을 이용하여 다음과 같이 서버 이용률 $\rho_{ir}(\mathbf{K})$ 를 구할 수 있다.

$$\rho_{ir}(\mathbf{K}) = \frac{A_r(\mathbf{K})}{\mu_{ir}}, i=0, 1, 2, r=1, 2$$

$\rho_{ir}(\mathbf{K})$ 는 i 번째 서비스를 가지는 기기가 형태 r 의 상황정보 객체에 의해 작동하게 되는 확률을 의미한다. 앞에서 말한 것처럼, 유비쿼터스 환경에서 기기는 제한된 성능을 가지고 있으므로, $\rho_{ir}(\mathbf{K})$ 는 단위 시간당 얼마나 많은 사용자의 상황 정보객체가 기기에 할당 되는지를 나타내는 기준이 될 수 있다. $\bar{K}_{ir}(\mathbf{K})$ 는 i 번째 서비스를 가지고 있는 기기에 존재하는 형태 r 정보 객체의 평균수를 의미하는 것으로 어느 노드가 복합적인 서비스를 제공받는데 지연 현상을 발생시키는 지를 파악하는 기준이 될 수 있다. $\bar{T}_{ir}(\mathbf{K})$ 는 i 번째 서비스를 제공하는 기기가 형태 r 정보 객체에게 서비스를 제공할 때까지 걸리는 시간을 나타내는 것으로 형태 r 의 한 객체가 두 개의 서비스로 구성된 복합적인 서비스를 모두 제공 받는데 얼마나 걸리는지에 대한 척도가 될 수 있다. 이러한 성능 척도들은 상황인지 컴퓨팅 환경을 설계하는데 사용된다.

5. 수치예제 및 시뮬레이션 비교분석

본 장에서는 변형 평균치 분석 알고리즘에서 얻어진 성능척도와 이산 사건 시뮬레이션을 통해 얻어진 성능척도를 비교하고, 제안된 방법이 유용하다는 것을 검증한다. <Table 1>은 두 가지 형태의 정보객체와 두 종류의 서비스가 존재하는 서비스 시스템에서 사용된 모수 값을 나타낸다.

따름정리 1과 모수 λ, α, l 을 이용하여 각 정보 객체의 평균 완성시간, 즉, 서비스 제공 정보 객체가 생성될 때까지 걸리는 평균 시간을 구할 수 있다. 이 평균 시간들의 역수가 노드 0에

서 각 서버들의 서비스율이 된다. <Table 1>에서 언급되지 않은 노드 2에서의 서비스율은 모두 1로 두었다. Case 1은 두 종류의 정보 객체가 노드 1에서 받는 서비스율이 같은 경우이고, Case 2는 서비스율이 정보 객체에 따라 2배, 경우 3은 10배, Case 4는 100배의 차이가 나게 정하였다.

Table 1. Data sets

λ	α	l	μ_{ir}			
			Case 1	Case 2	Case 3	Case 4
0.5	0.05	3	$\mu_{11} = 0.6$	$\mu_{11} = 1.2$	$\mu_{11} = 6$	$\mu_{11} = 60$
1	0.1	4	$\mu_{12} = 0.6$	$\mu_{12} = 0.6$	$\mu_{12} = 0.6$	$\mu_{12} = 0.6$

<Table 2>는 <Table 1>의 모수 값을 바탕으로 변형 평균치 분석 알고리즘으로 계산한 값과 시뮬레이션 값을 비교한 것이다. $A_1(\mathbf{K})$ 는 단위 시간당 서비스시스템이 처리하는 형태 1 정보 객체의 수, $\overline{K_1}(\mathbf{K})$ 는 첫 번째 서비스를 제공받으려는 평균 정보 객체의 수, $\overline{T_1}(\mathbf{K})$ 는 형태 1의 정보 객체가 형성되어 두 종류의 서비스로 이루어진 복합적인 서비스를 제공받는데 걸리는 평균시간, $\rho_1(\mathbf{K})$ 는 첫 번째 서비스를 제공하는 기기의 이용률을 나타낸다.

<Table 2>의 Case 1과 Case 2에서 변형 평균치 분석 알고리즘에 의해 구해진 성능척도와 시뮬레이션을 이용해 구해진 성능척도간의 상대오차가 4% 이내임을 알 수 있다. 이 두 경우는 첫 번째 서비스(●/M/m/FCFS 형태의 노드)에서 각 정보 객체별 서비스율의 차이가 없거나 비교적 적은(두 배) 경우이다. Case 3과 Case 4는 첫 번째 서비스에서 각 정보 객체의 서비스율 차이가 큰 경우(10배 혹은 100배)로 $\overline{K_1}(\mathbf{K})$ 을 제외한 나머지 성능척도들의 상대오차가 8% 이내임을 알 수 있다. 그러나 실제 상황인식 컴퓨팅 환경에서 Case 3과 Case 4와 같이 각 정보객체의 서비스율의 차이가 10배에서 100배정도 차이가 나는 경우는 드물다고 볼 수 있으므로, Case 3과 Case 4의 결과는 변형 평균치 분석 알고리즘이 각 정보 객체의 서비스율 차이가 큰 경우에도 유용함을 보여준다.

이러한 변형 평균치 분석법은 상황인지 컴퓨팅 환경에서 사용자의 상황이나 주변 환경과 관련된 데이터를 수집하여 사용

자에게 적합한 서비스를 제공하는 서비스시스템의 성능척도를 예측하는 데 사용할 수 있다. 이는 시스템을 실제 운영하기 전에 시스템의 성능을 사전에 파악하여 시스템의 설계 단계에서 활용할 수 있음을 의미한다. 예를 들어, 서비스시스템에서 각 사용자가 복합 서비스를 제공받는데 걸리는 시간이 일정 시간 τ 이내여야 한다면, 사전에 시스템의 성능을 알아볼 필요가 있다. 이 경우 변형 평균치 분석법을 사용하여 시스템에서 사용자가 복합서비스를 제공받을 때까지 걸리는 시간을 알 수 있다. 만약 이 시간이 정해진 시간 τ 를 만족하지 못한다면 사용자가 제공받는 서비스의 서비스율을 증가시켜 해결할 수 있다. 서비스율은 단위 시간당 서비스를 제공받는 사용자의 수를 의미하는 것으로 이것이 증가할수록 사용자가 서비스를 받는 데 걸리는 시간은 감소한다. 그러나 서비스율이 어느 정도 증가하면 사용자가 복합 서비스를 제공받을 때까지 걸리는 시간은 크게 감소하지 않는다. 그러므로 시스템이 사용자에게 시간 τ 이내에 복합 서비스를 제공하는 데 적합한 서비스율을 설계 단계에서 결정할 수 있다. 이 서비스율은 결국 서비스를 제공하는 컴퓨팅 기기의 성능을 결정하게 되므로 시스템 설계자는 컴퓨팅 기기가 갖추어야 하는 성능을 파악할 수 있다.

6. 결론

본 연구에서는 다양한 요구를 지닌 두 명의 사용자와 두 종류의 서비스를 제공하는 기기들이 존재하는 상황인지 서비스 제공 환경을 대기네트워크로 모형화하고 이를 변형 평균치 분석 알고리즘을 통해 여러 성능척도들을 구하였다. 이 성능 척도들은 서비스시스템에서 서비스 제공기기들을 설계하거나 개선하기 위한 기준이 될 수 있으며, 이산 사건 시뮬레이션과의 비교를 통해 제안된 접근방법이 효율적이고 유용함을 검증하였다. 특히 본 논문은 본 연구의 대상이 되는 상황인지 서비스 제공 시스템을 구성하고 있는 컴퓨팅 기기들의 협업을 보험, 금융 등 다양한 서비스 제공 객체로의 확장을 통하여 보다 일반적인 서비스 분야의 운영 성능에 대한 분석의 기초 도구로 활용될 수 있다.

추후 연구 방향의 일부로서 사용자의 수와 제공되는 서비스

Table 2. Comparison between modified MVA and Simulation

PM ¹⁾	Case 1			Case 2			Case 3			Case 4		
	MVA	Simul	R.E ²⁾ (%)	MVA	Simul	R.E ²⁾ (%)	MVA	Simul	R.E ²⁾ (%)	MVA	Simul	R.E ²⁾ (%)
$A_1(\mathbf{K})$	0.1018	0.1019	0.09	0.1137	0.1107	2.63	0.1253	0.1177	6.42	0.1283	0.1197	7.19
$\overline{K_1}(\mathbf{K})$	0.4615	0.4671	1.22	0.3559	0.3702	3.85	0.2521	0.2928	13.91	0.2257	0.2765	18.37
$\overline{T_1}(\mathbf{K})$	9.8205	9.8011	0.20	8.7982	9.0212	2.47	7.9804	8.4849	5.94	7.7964	8.3484	6.61
$\rho_1(\mathbf{K})$	0.3845	0.3890	1.15	0.3130	0.3175	1.41	0.2426	0.2462	1.45	0.2247	0.2306	2.54

Note) ¹⁾ PM : Performance Measure, ²⁾ R.E : Relative Error.

의 수를 일반화하는 것과 각 사용자가 다른 종류의 복합적인 서비스를 제공 받기를 원하는 경우를 고려할 수 있다. 즉, 두 명 이상의 사용자와 두 종류 이상의 서비스가 존재하며, 복합적인 서비스를 구성하는 서비스의 수가 각 사용자마다 다른 경우이다. 이러한 모형에 대한 성능척도는 본 논문을 확장함으로써 얻을 수 있다.

참고문헌

- Baldauf, M., Dustdar S., and Rosenberg F. (2007), A Survey on Context-Aware Systems, *International Journal of Ad Hoc and Ubiquitous Computing*, **2**(4), 263-277.
- Bolch, G., Greiner, S., Meer, H., and Trivedi, K. (1998), *Queueing Networks and Markov Chains*, John Wiley and Sons, Inc., New York.
- Dey, A. and Abowd, G. (2000), Towards a better Understanding of Context and Context-Awareness, *CHI'2000 Workshop on the What, Who, Where, When, and How of Context-Awareness*.
- Dey A. (2001), Understanding and Using Context, *Personal and Ubiquitous Computing*, **5**, 4-7.
- Hur, S. Lee, H., Shin, D., and Lee, W. (2007), A Time-Constrained Information Processing Model in Ubiquitous Environments, *ETRI Journal*, **29**(4), 489-496.
- Moran, T. and Dourish, P. (2001), Context-Aware Computing, *Special Issue of Human-Computer Interaction*, **16**, 1-8.
- Satyanarayanan, M. (2001), Pervasive Computing : Vision and Challenges, *IEEE Personal Communications*, **8**(4), 10-17.
- Schilit, B. and Theimer, M. (1994), Disseminating Active Map Information to Mobile Hosts, *IEEE Network*, **8**(5), 22-32.
- Schilit, B. Adams, N. and Want, R. (1994), Context-Aware Computing Applications, *IEEE Workshop on Mobile Computing Systems and Applications*, 85-90.
- Schmidt, A., Aidoo, K., and Takaluoma, A. (1999), Advanced Interaction in Context, *Lecture Notes in Computer Science*, **1707**, 89-101.
- Schmidt, A. (2006), Ontology-based user context management : The challenges of imperfection and time-dependence, *Lecture Notes in Computer Science*, **4275**, 995-1011.
- Strang, T. and Linnhoff-Popien, C. (2004), A context modeling survey, In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004-The Sixth International Conference on Ubiquitous Computing, 34-41.