

사용자 인지 서비스 가용도의 시뮬레이션 모델

함영만¹ · 이강원^{2*}

Simulation Model for User-Perceived Service Availability

Young-mam Ham · Kang-won Lee

ABSTRACT

Traditional system-oriented measures are no longer adequate to describe the availability perceived by the user. In this paper we investigate the service availability model and the user behavior graph. And we propose the simulation model, which determines the service availability considering user model, system model and service model at the same time. Through the example, we show how to construct the UBG of user model, system model and service model. And we investigate the effect of the parameters of system model and UBG on the service availability.

Key words : Service Availability, User Model, Service Model, System Model

요약

전통적인 시스템 차원에서의 신뢰도 분석 모형에서 탈피하여 사용자가 인지하는 서비스 가용도에 대한 필요성이 대두되고 있다. 본 논문에서는 우선 서비스 가용도 모형에 대해 살펴보고 이의 근간을 이루는 UBG에 대하여 논하였다. 그리고 사용자 모델, 시스템 모델 및 서비스 모델을 복합적으로 고려하여 서비스 가용도를 산출하는 시뮬레이션 모델을 제안하였다. 아울러 예제를 통하여 사용자 모델의 UBG와 시스템 모델을 작성하고 이를 토대로 서비스 모델을 정의하는 일련의 과정을 보여 주었다. 그리고 시스템 모델과 UBG의 여러 파라미터들이 서비스 가용도에 미치는 영향에 대해 살펴보았다.

주요어 : 서비스 가용도, UBG, 사용자 모델, 서비스 모델, 시스템 모델

1. 서론

최근 들어 전통적인 시스템 차원에서의 신뢰도 분석 모형에서 탈피하여 사용자가 인지하는 서비스 가용도 (User Perceived Service Availability)에 대한 필요성이 대두되고 있다. 사용자 관점에서의 시스템 고장은

첫째, 사용자가 시스템을 사용 하거나 운영할 때 고장이 발생 하거나,

둘째, 시스템이 고장 상태에 있을 때 사용자가 시스템

사용을 요구할 때만 인지될 수 있다.

특별히 사용자가 불규칙 적으로 시스템을 이용하고 사용자에게 일정 수준 이상의 서비스 품질(Quality of Service)을 제공해야 하는 통신 시스템이나 교통 시스템의 경우 서비스 가용도가 보다 적절한 척도일 수 있다. 사용자는 주어진 임무 기간 동안 여러 종류의 서로 다른 작업(Task)을 수행하는데 각 작업마다 요구하는 시스템 기능이 서로 다를 수 있다. 아울러 임무 기간 내내 시스템이 작동 상태에 있을 필요가 없고 사용자가 시스템을 요구하는 시간에만 시스템이 작동 상태에 있으면 된다. 이러한 측면들이 고려된 서비스 가용도의 분석 방법은 전통적인 신뢰도 분석 방법과는 다르다.

어떤 시스템이 두 개의 서브시스템 1과 2로 구성되어 있고, 서브시스템 1은 부품 R_1 과 R_2 로 이중화되어 있고, 서브시스템 2는 부품 R_3 으로 되어 있다고 하자. 전통적인 신뢰도 분석은 시스템 차원에서 이루어져 왔는데 이를 위해서 먼저 신뢰도 블록도(Reliability Block Diagram)를

*이 연구는 서울과학기술대학교 교내 학술 연구비 지원으로 수행되었다.

접수일(2011년 11월 30일), 심사일(1차 : 2012년 5월 18일), 게재 확정일(2012년 6월 22일)

¹⁾ 서울과학기술대학교 IT정책전문대학원

²⁾ 서울과학기술대학교 글로벌융합산업공학과

주 저 자 : 함영만

교신저자 : 이강원

E-mail; kwlee@snut.ac.kr

작성한 후 신뢰도 분석을 수행한다. 이제 사용자 차원에서 이 시스템을 이용하는 사용자 행위(User Behavior)를 살펴보자. 먼저 사용자는 이 시스템을 이용하여 두 개의 서로 다른 작업 J_1 과 J_2 를 수행하는데, 작업 J_1 의 수행은 서브시스템 1을, 작업 J_2 의 수행은 서브시스템 2를 이용한다고 가정하자. 만약 작업 J_1 수행 시, 시스템 1의 부품 1과 2가 모두 작동 불능 상태가 되면 임무 실패가 되고, 작업 J_2 수행 시에는 부품 3이 고장 상태이면 임무 실패가 된다. 이제 작업 J_1 과 J_2 의 발생 시간과 빈도, 시스템의 기능을 요구하지 않는 휴지 상태 그리고 임무 실패나 임무 종료 등을 복합적으로 고려하여 사용자 행위도(UBG: User Behavior Graph)를 작성할 수 있다. 따라서 사용자의 관점에서 서비스 가용도를 유도하기 위해서는 시스템 차원에서의 신뢰도 모형과 UBG를 복합적으로 고려하여야 한다.

서비스 가용도를 유도하기 위해 고려해야 하는 user - system interaction의 일반적 특성은 다음과 같다.

- 사용자가 인지하는 서비스 가용도를 유도하기 위해서는 사용자의 시스템 이용 형태를 나타내주는 UBG가 필요하다. UBG는 본 논문의 3장에서 다루었다.
- 시스템과 user interaction의 전 기간 동안 모든 시스템 자원이 작동 상태에 있을 필요는 없다. 다양한 종류의 서비스를 수용하는 대규모 분산 시스템의 경우 user task를 처리하기 위하여 시스템 자원의 일부분만 필요로 한다. 따라서 전통적인 시스템의 신뢰도나 가용도 보다는 서비스 가용도가 더 적합한 측도일 수 있다.
- 시스템 자원은 사용자가 이 자원을 요구하는 시간에 on(동작) 상태에 있으면 된다. 시스템 자원이 요구되어 지지 않을 때, 시스템의 고장(off) 상태는 사용자가 인지하는 서비스 가용도에 전혀 영향을 주지 않는다. 다음 그림 1에서 t_1 과 t_3 이 task 도착 시점을 나타

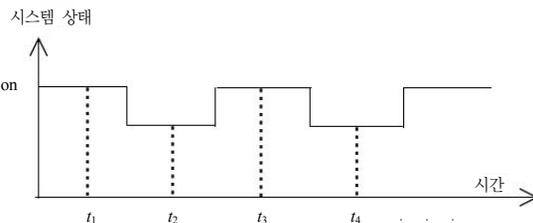


그림 1. task 도착 시점에 따른 시스템가용도

낸다면 사용자 관점에서 시스템은 항상 가용하다. 반면에 t_2 와 t_4 가 task 도착 시점이라면 시스템은 항상 가용하지 않다. 즉 같은 시스템 자원 하에서 task 도착 시점에 따라 시스템 가용도 값은 전혀 다른 값을 나타낸다. 이는 서비스 가용도 분석은 일반적으로 행해지는 시스템 중심의 분석이 아니라 사용자 중심의 분석이어야 한다는 것을 나타낸다.

- 시스템 자원은 user - system interaction 동안 여러 번 요구될 수 있다. 이 경우 전통적인 시스템 가용도 (System Availability)는 서비스 가용도 분석에 적합하지 않다. 대신 일정 임무 기간 동안 여러 차례의 task 도착을 고려하는 joint availability나 random request availability^[11]가 서비스 가용도 분석에 보다 적합한 측도일 수 있다.

서비스 가용도 측도의 도입 필요성을 강조한 여러 선행 연구들이 있었다^[2-4]. 특히 참고 문헌 [4]는 통신 분야에서 서비스 가용도 개념 도입의 필요성을 주장했다. 참고 문헌 [5]는 web 서비스 가용도를 제공자 관점과 client 관점에서 살펴보았고 참고 문헌 [6]에서는 QoS(Quality of Service) 관점에서 VoIP 서비스 가용도 개념을 도입하였다. 참고 문헌 [7]은 IP backbone 망의 서비스 가용도 개념을 제안하였다.

한편 최근 들어서는 여러 서비스 가용도의 모델링에 관한 연구가 진행 되고 있다^[10]. 참고 문헌 [11], [12]에서는 web 서비스의 사용자 인지 가용도 측도로 “Status Identification Based Availability”를 제안하였는데, 사용자가 인지하는 서비스 상태를 네 가지로 정의하였다. 참고 문헌 [13]은 소프트웨어의 사용자 인지 서비스 가용도를 위한 Markovian 모델을, 참고문헌 [14], [15]에서는 다양한 통신망 장치에 적용되는 우선순위 공유보호 시스템(Priority Shared Protection System)을 위한 사용자 인지 가용도 모델을 제안하였다. 위의 연구들은 특정 시스템을 기반으로 사용자가 인지하는 고장 형태를 정의하고, 이를 토대로 사용자 인지 서비스 가용도 모델을 제안하고 있기 때문에 일반적으로 사용할 수 있는 모델과는 거리가 멀다.

본 연구에서는 일반적으로 사용 가능한 서비스 가용도 모델 구축을 목적으로 사용자 모델, 시스템 모델 그리고 서비스 모델을 복합적으로 고려하여 서비스 가용도를 산출하는 시뮬레이션 모델을 제안하였다. 아울러 예제를 통해서 사용자 모델의 UBG와 시스템 모델을 작성하고 이를 토대로 서비스 모델을 정의하는 일련의 과정을 보여주었다. 그리고 시스템 모델과 UBG의 여러 파라미터들이

서비스 가용도에 미치는 영향에 대해 살펴보았다.

본 논문의 목적은 시스템 차원에서의 신뢰도 모형과 UBG를 복합적으로 고려한 서비스 가용도 산출에 있다. 이를 위해 2장에서는 사용자가 인지하는 서비스 가용도 모형을 살펴보고 서비스 가용도를 정의하였다. 3장에서는 사용자의 시스템 이용 행태를 나타내는 UBG 모형을 다루었다. 4장에서는 시스템 차원에서의 신뢰도 모형과 UBG를 복합적으로 고려하여 서비스 가용도를 산출하는 시뮬레이션 방법을 제안하였고 5장에서 본연구의 결론을 내렸다.

2. User-Perceived Service Availability Model

서비스 가용도라 불리는 새로운 신뢰도 측도가 사용자와 시스템의 상호 작용을 고려하기 위하여 제안되었다. 기존 신뢰도 측도가 시스템 수준에서 정의되어지는 반면에 서비스 가용도는 사용자 관점에서부터 정의된다. 사용자 관점에서부터 서비스 가용도를 개괄적으로 살펴보면 다음 그림 2와 같다. 이는 사용자 모델, 서비스 모델, 그리고 시스템 모델의 세 개 수준으로 구성되어 있다

먼저 사용자 모델은 임무 기간 중 서로 다른 임의 개수의 작업 발생과, 발생 시간과 같은 사용자의 행위를 포함한다. 다음 그림 2에서 t_1, t_2, \dots, t_m 은 task의 도착 시점을 나타낸다. 일반적으로 사용자 모델은 UBG로 표현될 수 있는데 task의 발생 유형과 시간, 빈도, 시스템의 기능을 요구하지 않는 휴지 상태, 그리고 임무 실패나 임무 종료

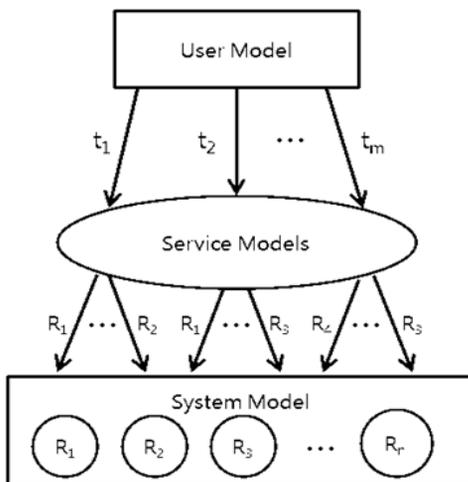


그림 2. 사용자 관점에서 서비스 가용도

등을 복합적으로 고려하여 UBG를 작성한다. 시스템 모델은 전통적 신뢰도 모델에 해당되는 부분으로 하드웨어나 소프트웨어로 구성된 시스템 자원(그림 2의 R_1, R_2, \dots, R_r)의 고장, 수리 그리고 유지 보수 등을 포함한다. 서비스 모델은 사용자 모델과 시스템 모델의 상호 작용으로 표현된다. 즉, 서비스 모델은 시스템 모델과 task의 요구 사항(임의로 도착하는 task가 요구하는 시스템 자원, 예로 처음 도착한 task는 R_1 과 R_2 를 요구하고 두 번째 도착한 task는 R_1 과 R_3 요구)들로부터 유도된다. 시스템에서 task 수행은 이 서비스 모델에 의해 표현된다.

전통적인 시스템 신뢰도 측도에 비해 서비스 가용도는 시스템 자원의 신뢰도뿐만 아니라 시스템 이용 시간과 빈도와 같은 사용자 이용 형태, 그리고 사용자가 시스템에게 요구하는 일의 특성, 즉 시스템 자원 등을 복합적으로 고려해야 한다. 사용자는 임의의 서로 다른 시간에 서로 다른 특성을 갖는 다수 개의 task를 발생시킨다. 이때 특정 임무 기간 동안 도착한 모든 task를 성공적으로 처리할 확률을 나타낼 수 있는 새로운 서비스 가용도 측도가 필요하다. 모든 task가 성공적으로 처리되기 위해서는 먼저 하드웨어나 소프트웨어로 구성된 시스템 자원이 task 도착 시점에 고장 없이 작동해야 한다. 이와 아울러 도착한 task가 원하는 서비스 특성 등을 고려하여 시스템이 사용자가 원하는 서비스를 제공할 수 있어야 한다. 이러한 서비스 가용도 특성을 고려하여 본 연구에서는 서비스 가용도를 다음과 같이 정의하였다.

시스템 이용 시 사용자는 다양한 형태의 서비스를 받고자 서로 다른 시간에 다양한 형태의 임의 개수의 작업을 요구한다. 이때 사용자가 인지하는 서비스 가용도는 임무 기간 동안 발생한 모든 작업들이 성공적으로 처리될 확률로 정의된다.

3. User Behavior Graph

User behavior graph는 사용자의 시스템 이용 행태를 나타내는데, 이용 형태의 특성에 따라 적당한 UBG를 정의하여 사용하면 된다. 먼저 사용자는 임무 기간에 임의 개수의 task를 발생시킨다는 가정 하에, 도착한 task가 모두 동일한 형태인 경우와 서로 다른 형태인 경우가 있기 때문에 요구하는 시스템 자원이 서로 상이한 두 가지 경우로 나누어 UBG의 작성 예를 나타냈다.

3.1 Task 도착 과정을 나타내는 UBG

이 UBG는 일정 임무 기간 동안 발생하는 task 도착 과

정을 모형화한다. 서비스 가용도 값은 task 도착률 뿐만 아니라 도착률 형태에 따라 서로 다른 값을 갖는다. 예로 임무 기간 초기에 시스템이 on 상태로 시작하는 그림 3과 같은 시스템 상태를 가정하자.

이때 그림 3(a)와 같이 시스템이 on 상태에 있는 임무 기간 초반에 task 도착률이 높으면 비교적 높은 서비스 가용도 값을 갖는다. 반대로 그림 3(b)와 같이 임무 기간 후반에 task 도착이 몰리면 앞의 경우에 비해 상대적으로 낮은 서비스 가용도 값을 갖는다. 서비스 가용도는 Task 도착률 형태에 따라 서로 다른 값을 가진다는 것을 알 수 있다. 시간에 따른 task 도착률의 변화를 고려하기 위하여 task 도착을 다음과 같은 non-homogeneous Poisson 과정으로 가정할 수 있다.

$$\Pr[N(T) = k] = \frac{e^{-M(T)} \cdot M(T)^k}{k!}$$

위 식에서 $M(T)$ 는 non-homogeneous Poisson 과정의 mean value function을 나타낸다.

Task의 도착 과정만을 나타내는 UBG는 도착하는 task가 모두 동일한 작업일 경우에 한해서 이용 가능하다. 다음 그림 4는 UBG를 나타낸다. 여기서 $P_i (i = 1, 2, \dots, k)$ 는 i 번째로 도착한 task 수행을 나타내며 k 는 non-homogeneous Poisson 과정에서 결정된다. T_i 는 task의 도착 시간 간격을 나타내는데 사용자 ‘think time’으로 생각할 수

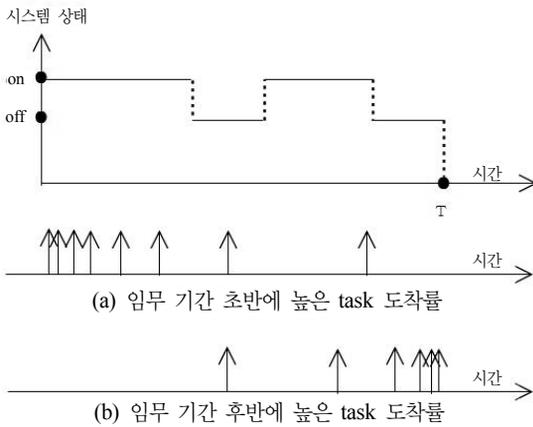


그림 3. 시스템 상태와 task 도착률

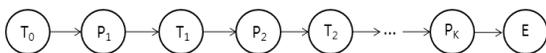


그림 4. 임의의 task 도착 과정을 나타내는 UBG

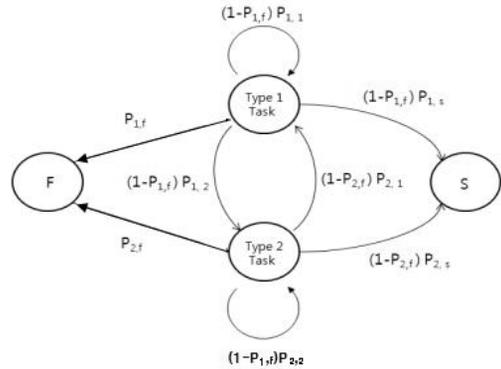


그림 5. 다양한 형태의 task를 고려한 UBG

있다. E는 임무 종료 상태를 나타낸다.

3.2 서로 다른 시스템 자원을 요구하는 Task를 나타내는 UBG

임무 기간 동안 사용자는 k 개의 서로 다른 종류의 task를 수행한다고 가정하자. Type $k (k = 1, 2, \dots, k)$ 의 task 수행을 위해서는 시스템 자원 $R_{k,1}, R_{k,2}, \dots, R_{k,m}$ 이 요구된다고 하자. Type k 의 task 수행 시 요구되는 시스템 자원들이 task가 원하는 서비스를 제공할 수 없는 고장 상태에 있다면 임무 실패로 되어 상태 F로 간다. Type k 의 task 수행 후, 임무 실패로의 전이 확률은 $P_{k,f}$ 로 주어진다. 확률 $P_{k,f}$ 는 task 수행을 위해 필요한 시스템 자원인 $R_{k,1}, R_{k,2}, \dots, R_{k,m}$ 로 구성된 신뢰도 블록도로부터 구할 수 있다.

만약 task를 성공적으로 수행하면, 같거나 다른 종류의 task 수행을 다시 요구하거나 임무가 성공적으로 종결되어 상태 S로 간다. Type k 의 task 수행 후 Type $j (j = 1, 2, \dots, k)$ 의 task 수행으로 전이 확률은 $(1 - P_{k,f}) * P_{k,j}$ 로, 임무가 종결될 확률은 $(1 - P_{k,f}) * P_{k,s}$ 로 주어진다. 그림 5는 $k = 2$ 일 때 위에서 언급한 사용자의 시스템 이용 형태를 나타내는 UBG이다. Task 수행의 각 노드에서 체류 시간은 task 수행 소요 시간으로 볼 수 있는데 이를 고려하면 UBG는 semi-Markov chain으로 모형화 된다.

위의 두 가지 형태의 UBG 이외에도 다양한 형태의 UBG가 존재할 수 있다. 앞에서 언급 했듯이 사용자가 시스템을 이용하는 이용 형태의 특성에 따라 적당한 UBG를 정의하여 사용하면 된다.

4. Simulation Model

본 장에서는 서비스 가용도를 사용자 관점에서 정의한

그림 2와 3장에서 제시한 UGB를 바탕으로 서비스 가용도 모형을 구축하였다. 먼저 수리적 모델에 대하여 살펴보고 시뮬레이션 모델에 관해 논하였다.

일반적으로 시스템 모델 위주의 전통적인 신뢰도 분석은 수리적 모델 구축이 비교적 용이하다. 그러나 사용자 모델과 시스템 모델, 그리고 사용자 - 시스템 상호 작용으로 나타나는 서비스 모델을 복합적으로 고려해야 하는 서비스 가용도의 수리적 모델을 구축하는 것은 매우 어렵다. 따라서 매우 제한된 범위 내에서만 수리적 모델 구축이 가능하다. 본 연구의 선행 연구격인 [1]에서는 random request availability를 제시하였는데, 이는 주어진 user session 기간 동안에 임의로 발생하는 task를 성공적으로 수행할 확률로 정의된다. 사용자 모델로는 일정 임무 기간 중 task 발생이 non-homogeneous Poisson 과정을 따른다고 가정하였다. 시스템 모델로는 작동과 고장 두 가지 상태를 갖는 Markov 모델을 가정하여 수리적 모형을 구축하였는데 random request availability, $A(T)$ 는 다음과 같이 표현된다.

$$A(T) = \sum_{k=1}^{\infty} \left[\int \cdots \int_{\Omega} A(t_1, t_2, \dots, t_k) \cdot \prod_{i=1}^k m(t_i) dt_1 \cdots dt_k \right] \cdot \exp[-M(T)] + \exp[-M(T)]$$

여기서

$$A(t_1, t_2, \dots, t_k) = \prod_{i=1}^k \left[\frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} \cdot \exp[-(\lambda + \mu) \cdot (t_i - t_{i-1})] \right]$$

$$\Omega = 0 \leq t_1 < t_2 < \dots < t_k \leq T$$

T = 임무 기간

λ = 시스템 자원 고장률

μ = 시스템 자원 수리율

$M(t)$ = non-homogeneous Poisson 과정의 mean value function

$m(t) = dM(t)/d(t)$: Task 도착률

하지만 도착하는 task가 모두 동일한 작업을 수행한다고 가정하여, 서로 다른 형태의 task 들이 서로 다른 시스템 자원을 요구하는 그림 2의 서비스 모델을 나타내지 못한다. 또한 시스템 모델도 작동과 고장을 반복해 나가는 하

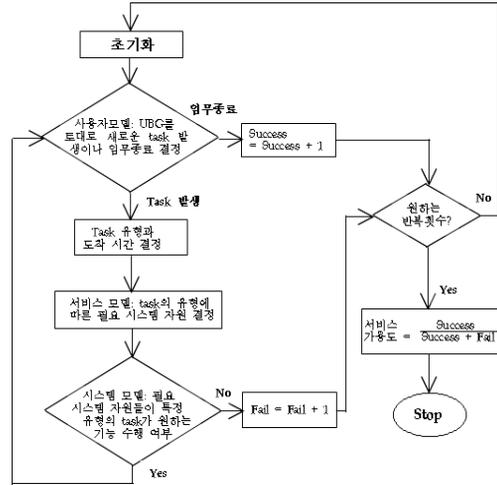


그림 6. 시뮬레이션 흐름도

나의 부품만을 고려하여 시스템 자원을 현실적으로 나타내지 못하고 있다. 이러한 점을 모두 고려한 수리적 모형 구축은 현실적으로 불가능해 보인다. 따라서 본 연구에서는 사용자가 인지하는 서비스 가용도 산출 모델로 시뮬레이션 방법을 제시하였다.

개략적인 시뮬레이션 흐름을 살펴보면 다음과 같다. 그림 6은 전체적인 시뮬레이션 흐름도를 나타낸다.

- 먼저 사용자 모델의 UGB를 토대로 task를 발생시키거나 임무 종료 여부를 결정한다. 발생한 task의 유형도 UGB를 바탕으로 결정된다.
- 발생한 task의 형태에 따른 task가 요구하는 시스템 자원들을 서비스 모델로부터 결정한다.
- 이제 시스템 모델을 토대로 특정 유형의 task 도착 시점에 task 유형에 따른 시스템 자원의 가용 여부를 결정한다. 만약 시스템 자원들이 task가 요구하는 기능을 수행할 수 없으면 임무 실패로 귀결된다. 시스템 자원들이 task가 요구하는 기능을 수행할 수 있으면 도착한 task는 성공적으로 처리되고 첫 번째 단계로 돌아간다.

5. Numerical Example

5.1 사용자, 서비스 그리고 시스템 모델

Web 서비스나 DB 서비스를 구현하기 위해 두 개의 노드 A와 B로 이루어진 다음 그림 7과 같은 컴퓨터 cluster를 생각해 보자.

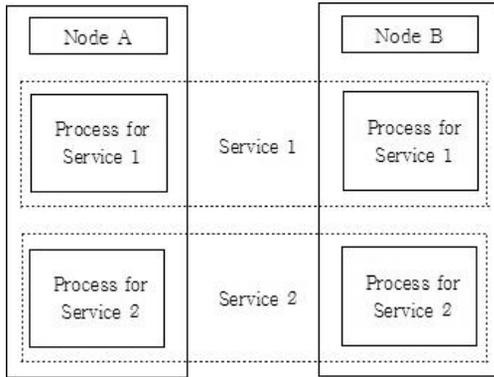


그림 7. 컴퓨터 Cluster 구조

Software fault tolerance를 제공하기 위해서 프로세스 중복을 채택하였다. 즉, 각 서비스를 위한 소프트웨어 프로세스는 신뢰도를 높이기 위해 두 개의 노드에 중복되어 있다. 따라서 서비스 1은 두 개의 프로세스 중 최소 하나만 작동해도 되고, 서비스 2도 마찬가지다. 소프트웨어 프로세스와 노드 하드웨어는 고장이 발생할 수 있다; 서비스 1 프로세스의 고장률은 λ_1 , 서비스 2 프로세스의 고장률은 λ_2 로 주어지고 각 노드의 하드웨어 고장률은 λ_h 로 주어진다. 노드에 하드웨어 고장이 발생하면 노드에서 돌아가는 두 개의 소프트웨어 프로세스는 사용할 수 없다. 이 cluster는 한명의 보수 인력이 있다고 가정한다. 서비스 1 프로세스와 2 프로세스 그리고 하드웨어 고장을 위한 수리율은 각각 u_1 , u_2 그리고 u_h 로 주어진다. 다수 개의 고장이 동시에 발생한 경우 수리공은 프로세스 두 개가 모두 고장 난 서비스를 우선 수리한다.

5.1.1 사용자 모델

위 예제의 사용자 행태도인 UBG는 다음 그림 8과 같다. 사용자는 user session 동안 임의의 개수의 서비스 1 혹은 2를 요구할 수 있다. 그림 8에서 상태 iP ($i = 1, 2$)는 서비스 i 를 요구해 서비스 i 프로세스가 작동하기를 원하는 요구 상태를 나타낸다. 요구가 성공적으로 수행된 후에, 모델은 사용자 ‘think state’인 iT 로 들어가는데 이 상태에서는 어떤 서비스도 요구하지 않는다. 사용자가 상태 iT 를 떠나면 사용자는 확률 P_{ji} 로 서비스 j ($j = 1, 2$)를 요구해 상태 jP 에 들어 갈 수 있고, 확률 P_{iE} 로 또 다른 서비스 요구 없이 session을 끝낼 수도 있다. 상태 iP 에 있을 때 서비스 i 를 위한 요구가 성공적으로 수행될 수 없으면 모델은 user session 실패를 나타내는 상태 F 로 들어간다.

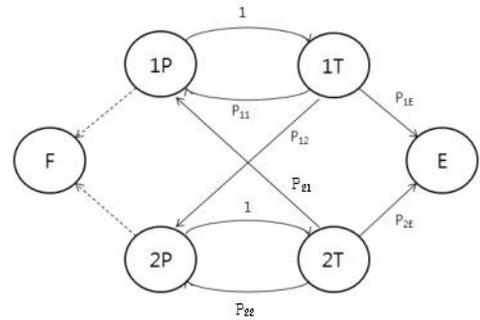


그림 8. 사용자 모델의 UBG

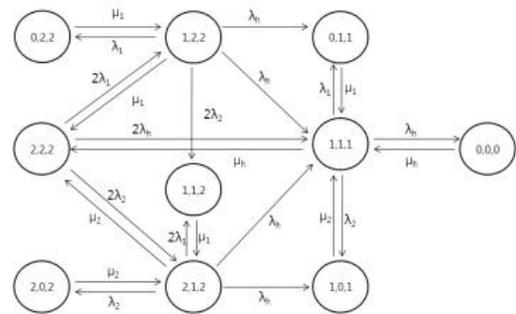


그림 9. 시스템 모델

5.1.2 시스템 모델

위의 computer cluster를 위한 시스템 모델은 그림 9에 나타나 있다. 시스템 모델의 상태 수를 줄이기 위하여 동시에 발생할 수 있는 고장의 최대 수는 두 개로 가정하였다. 각 상태는 (x,y,z) 로 나타낼 수 있는데 x 와 y 는 각각 작동 상태에 있는 서비스 1과 2 프로세스의 개수를 나타내고 z 는 작동 상태에 있는 노드의 개수를 나타낸다. 예로 $(2,2,2)$ 는 노드 A와 B에 있는 두 개의 서비스 1 프로세스와 두 개의 서비스 2 프로세스가 모두 작동 상태에 있음을 나타내고 노드 A와 B에 하드웨어 고장이 없음을 나타낸다. 한편 $(1,2,2)$ 는 노드 A혹은 B에 있는 서비스 1 프로세스 중 하나가 작동 불능 상태이고 서비스 2 프로세스는 모두 작동 상태에 있음을 나타낸다. 한편 노드 A와 B 모두에 하드웨어 고장이 없다. 즉, 두 노드 A와 B 모두 작동 상태에 있지만 그 중 한 노드에서 서비스 1 프로세스는 작동하지 않는다.

5.1.3 서비스 모델

사용자 모델과 시스템 모델을 토대로 서비스 모델을 정의할 수 있다. 사용자 모델 UBG의 상태 $1P$ 에서는 사용

자가 원하는 서비스를 성공적으로 수행하기 위해서 사용자가 서비스를 원하는 시각에 시스템 모델의 상태 $\{(x,y,z) \mid x \geq 1\}$ 에 있어야 한다. 사용자 상태 2P에서는 상태 $\{(x,y,z) \mid y \geq 1\}$ 에 있어야 한다. UBG의 상태 1T와 2T에서는 사용자가 서비스를 요구하지 않기 때문에 시스템 모델이 어떤 상태에 있어도 상관없다.

5.2 Results

5.2.1 입력 자료

5.1에서 제시한 예제의 시뮬레이션을 위한 파라메타를 Table 1과 같이 정하였다.

5.2.2 결과

시뮬레이션을 통한 서비스 가용도 값은 다음 과정을 통해 구할 수 있다. 그림 8의 UBG에서 보듯이 매 user session, 즉 한 번의 시뮬레이션 시행은 성공 혹은 실패로 귀결된다. 일정 횟수의 시뮬레이션 반복 수행 후 서비스 가용도는 “user session 성공 횟수 / (user session 성공 횟수 + user session 실패 횟수)”로 정해진다.

한편 사용자 모델의 UBG를 고려하지 않고 시스템 모델만을 고려할 경우는 안정 상태 시스템 가용도(Steady-State System Availability)를 구할 수 있다. 이는 시스템 모델에서 특정 시점에 시스템이 상태 $\{(x,y,z) \mid x \geq 1, y \geq 1\}$ 에 있을 확률로 정의된다. 이는 시뮬레이션 수행기간 동안 “상태 $\{(x,y,z) \mid x \geq 1, y \geq 1\}$ 에 머무는 시간 / 총 시뮬레이션 수행 시간”으로 구할 수 있다.

본 연구의 시뮬레이션은 ARENA를 이용하여 수행하였다.

- User session 기간 동안 task 발생 횟수

위 파라메타를 이용한 시뮬레이션 수행 결과 user session 기간 동안 task 발생 횟수는 3.202로 나타났다. 이 값은 UBG의 전이 확률 값에 따라 변하는 것으로 예측된다. 다음의 세 가지 전이 확률 값 하에서 task 발생 횟수를 조사하여 그림 10에 나타냈다.

- a. $P_{11} = 0.4, P_{12} = 0.3, P_{1E} = 0.3, P_{21} = 0.4, P_{22} = 0.3, P_{2E} = 0.3$ (본래 전이 확률 값)
- b. $P_{11} = 0.05, P_{12} = 0.05, P_{1E} = 0.9, P_{21} = 0.05, P_{22} = 0.05, P_{2E} = 0.9$
- c. $P_{11} = 0.45, P_{12} = 0.45, P_{1E} = 0.1, P_{21} = 0.45, P_{22} = 0.45, P_{2E} = 0.1$

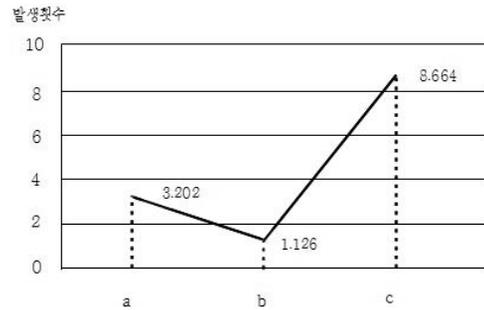


그림 10. 전이 확률에 따른 task 발생 횟수

Task 발생 횟수는 c의 경우가 8.664로 가장 높고 그리고 경우 a가 3.202, 경우 b가 1.126으로 나타났다. c의 경우는 task 처리 후 think time을 거친 다음에 user session을 종료할 확률이 0.1로 낮은 반면에 새로운 task를 발생시킬 확률은 0.9(서비스 1 프로세스 요구가 0.45, 서비스 2 프로세스 요구가 0.45)로 높아 session 기간 동안 task 발생 횟수가 많다. 반면에 b의 경우는 task 처리 후 think time을 거친 다음에 user session을 종료할 확률이 0.9로 높은 반면에 새로운 task를 발생시킬 확률은 0.1(서비스 1 프로세스 요구가 0.05, 서비스 2 프로세스 요구가 0.05)로 낮아, session 기간 동안 task 발생 횟수가 적다. 경우 a는 b와 c의 중간 값을 갖는다.

- 서비스 가용도와 안정 상태 시스템 가용도

Table 1의 시뮬레이션 파라메타를 사용하여 서비스 가용도와 시스템 가용도를 구하였다. 서비스 가용도는 0.961로 시스템 가용도 0.973 보다 낮은 값을 갖는 것으로 나타났다. 이는 시스템 가용도는 임의의 한 시점에 시스템이 작동 상태에 있을 확률인 반면에 서비스 가용도는 사용자 모델의 UBG를 토대로 발생하는 모든 task의 발생 시점에 task 유형에 따라 원하는 시스템 자원이 작동 상태에 있을 확률로 정의되기 때문이다. 실제로 본 예제에서 session 기간 동안 발생하는 task의 개수는 3.202개로 나타났다. 한편 위의 전이 확률 값 $(P_{11} = 0.05, P_{12} = 0.05, P_{1E} = 0.9, P_{21} = 0.05, P_{22} = 0.05, P_{2E} = 0.9)$ 의 경우 서비스 가용도는 0.993으로 시스템 가용도 0.973 보다 높은 값을 갖는다. 이는 일단 task 발생 개수가 1보다 크어도 불구하고 서비스 가용도가 더 높은 값을 갖는 이유는 시스템 가용도는 시스템 모델에서 task가 임의 시점에 시스템이 상태 $\{(x,y,z) \mid x \geq 1, y \geq 1\}$ 에 있을 것을 요구하지만 서비

Table 1. 시뮬레이션 파라메타

Parameter	Value
- 서비스 1 프로세스의 고장률, λ_1	0.25/hour
- 서비스 2 프로세스의 고장률, λ_2	0.25/hour
- 노드 하드웨어 고장률, λ_h	0.25/hour
- 서비스 1 프로세스의 수리율, μ_1	2.5/hour
- 서비스 2 프로세스의 수리율, μ_2	2.5/hour
- 노드 하드웨어 수리율, μ_h	2.5/hour
- 상태 1T의 사용자 think 시간, T_1	0.1 hour
- 상태 2T의 사용자 think 시간, T_2	0.1 hour
- 1T에서 1P로의 전이 확률, P_{11}	0.4
- 1T에서 2P로의 전이 확률, P_{12}	0.3
- 1T에서 E로의 전이 확률, P_{1E}	0.3
- 2T에서 1P로의 전이 확률, P_{21}	0.4
- 2T에서 2P로의 전이 확률, P_{22}	0.3
- 2T에서 E로의 전이 확률, P_{2E}	0.3

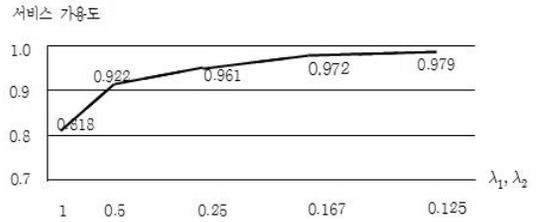
스 가용도는 시스템 상태에 대한 요구 조건이 훨씬 약하다. 즉, UBG의 상태 1P에서는 $\{(x,y,z) \mid x \geq 1\}$ 이면 되고, 상태 2P에서는 $\{(x,y,z) \mid y \geq 1\}$ 이면 된다. 더욱이 사용자 think time인 1T나 2T에서는 사용자가 서비스를 요구하지 않기 때문에 시스템 모델이 어떤 상태에 있어도 무관하다.

- 서비스 1과 2의 프로세스 고장률 λ_1 과 λ_2 그리고 하드웨어 고장 λ_h 에 따른 서비스 가용도

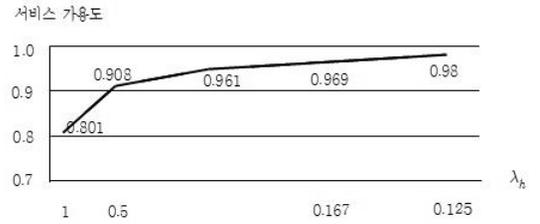
먼저 서비스 1과 2의 프로세스 고장률 λ_1 과 λ_2 를 변화시켜 나가면서 서비스 가용도 값을 구하였다. 그림 11(a)에서 볼 수 있듯이 고장률이 증가하면 서비스 가용도 값은 감소하고 반대로 고장률이 감소하면 서비스 가용도 값은 증가함을 볼 수 있다. 그림 11(b)는 노드 하드웨어 고장률 변화에 따른 서비스 가용도 값을 보여 주는데 서비스 프로세스의 경우와 비슷한 결과를 보여준다.

- 서비스 1과 2의 프로세스 수리율 μ_1 과 μ_2 그리고 노드 하드웨어 수리율 μ_h 에 따른 서비스 가용도

먼저 서비스 1과 2의 프로세스 수리율 μ_1 과 μ_2 를 변화시켜 나가면서 서비스 가용도 값을 구하였다. 그림 12(a)에서 볼 수 있듯이 수리율이 증가하면 서비스 가용도 값은 증가하고 반대로 수리율이 감소하면 서비스 가용도 값

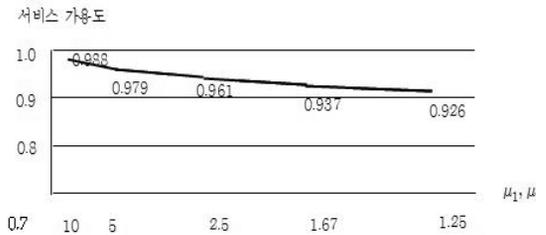


(a) λ_1 과 λ_2 에 따른 서비스 가용도

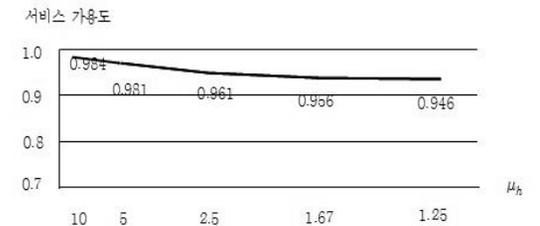


(b) λ_h 에 따른 서비스 가용도

그림 11. λ_1 과 λ_2 그리고 λ_h 에 따른 서비스 가용도



(a) μ_1 과 μ_2 에 따른 서비스 가용도



(b) μ_h 에 따른 서비스 가용도

그림 12. μ_1 과 μ_2 그리고 μ_h 에 따른 서비스 가용도

은 감소함을 볼 수 있다. 그림 12(b)는 노드 하드웨어 수리율 변화에 따른 서비스 가용도 값을 보여 주는데, 서비스 프로세스의 경우와 비슷한 결과를 나타낸다.

- 사용자 think time T_1 과 T_2 에 따른 서비스 가용도

사용자 think time T_1 과 T_2 를 변화시켜 나가면서 서비스 가용도를 구하였다. 그림 13에서 볼 수 있듯이 T_1 과 T_2 값 변화는 서비스 가용도 값에 영향을 미치지 않는 것

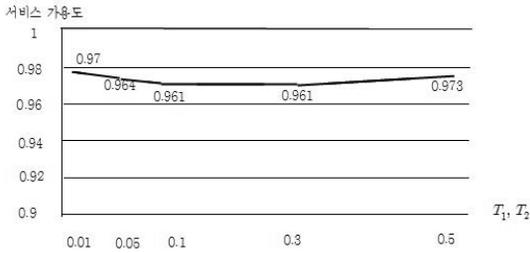


그림 13. T₁과 T₂에 따른 서비스 가용도

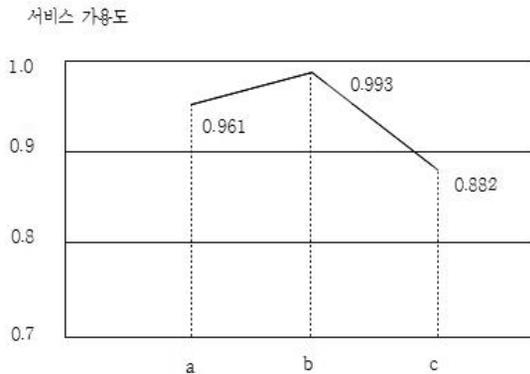


그림 14. 전이 확률에 따른 서비스 가용도

로 나타났다.

- UBG의 전이 확률 P_{ij}에 따른 서비스 가용도

UBG의 전이 확률을 앞에서 처럼 a, b 그리고 c 세 가지를 변화시키면서 서비스 가용도 값을 산출하였다. 그림 14에서 보듯이 b의 경우가 0.993으로 서비스 가용도 값이 가장 높고 a의 경우는 0.961, c의 경우가 0.882로 서비스 가용도 값이 가장 낮은 것으로 나타났다. b의 경우는 think time에서 바로 user session을 끝낼 확률이 0.9로 높아서 한 session 내에 시스템이 처리해야 할 task의 숫자가 작다. 따라서 서비스 가용도 값은 높은 값을 갖게 된다. 반대로 c의 경우는 think time에서 바로 user session을 끝낼 확률이 0.1로 낮아서 한 session 내에 시스템이 처리해야 할 task의 숫자가 많게 된다. 따라서 서비스 가용도 값은 비교적 낮은 값을 갖게 된다. 실제로 시뮬레이션 수행 결과 시스템이 처리해야 할 task의 개수는 앞에서 이미 언급한 것처럼 a, b 그리고 c의 경우 각각 3.202, 1.126 그리고 8.664로 나타났다.

6. 결 론

전통적인 시스템 차원에서의 신뢰도 분석 모형에서 탈피하여 사용자가 인지하는 서비스 가용도에 대한 필요성이 대두되고 있다. 본 논문에서는 우선 서비스 가용도 모형에 대해 살펴보고 이의 근간을 이루는 UBG에 대하여 논하였다. 그리고 사용자 모델, 시스템 모델 및 서비스 모델을 복합적으로 고려하여 서비스 가용도를 산출하는 시뮬레이션 모델을 제안하였다. 아울러 예제를 통해서 사용자 모델의 UBG와 시스템 모델을 작성하고 이를 토대로 서비스 모델을 정의하는 일련의 과정을 보여주었다.

본 연구에서는 제안한 시뮬레이션 모델을 통해 다음 사항들을 조사하였다.

- 먼저 사용자 인지 서비스 가용도 값과 전통적인 가용도 값을 task 도착률과 사용자 think time, 그리고 사용자 상태 전이 확률 값에 따라 비교 분석하였다. 이를 통해 이 두 측도 값의 차이가 발생하는 이유를 설명하고 사용자 인지 서비스 가용도의 필요성을 확인하였다. 즉, UBG의 여러 파라메타 값들이 서비스 가용도에 미치는 영향을 조사하였다.
- 또한 사용자 인지 서비스 가용도를 시스템 모델의 주요 파라메타인 시스템 고장률과 수리율에 따른 변화를 분석함으로써 시스템 모델의 여러 파라메타 값들이 서비스 가용도에 미치는 영향을 조사하였다.

본 연구에서 개발한 시뮬레이션 모델은 다음의 측면에서 계속 보완될 여지를 남겨두고 있다.

첫째 본 논문에서는 계산상의 어려움과 모델의 복잡성 때문에 서비스 가용도 산출을 위해서 시뮬레이션 모델을 제안하였다. 그런데 본 연구에서 사용된 다양한 파라메타들과 서비스 가용도의 인과 관계 파악 이라든지 서비스 가용도에 대한 보다 명확한 개념 정립을 위해서는 수리적 모델이 필요하다. 이는 몇 가지 추가적인 가정이나 모델의 단순화를 통해 이룰 수 있는데, 이에 대한 연구는 추후 과제로 남긴다.

둘째 본 논문에서 제안한 시뮬레이션 모델은 크게 사용자 행위 모델, 시스템 모델, 그리고 이 두 모델의 상호작용으로 정의되는 서비스 모델로 구성된다. 이 중 시스템 모델은 과거 수십 년 동안 연구가 진행되어 연구 결과가 많이 누적되어 있다. 반면에 사용자 행위 모델은 아직 연구 초창기로 일반적으로 사용할 수 있는 모델이 개발되

어 있지 않다. 본 논문에서는 semi-Markov chain을 이용한 UBG를 제안하였는데 노드간의 전이 확률과 각 노드에서의 체류 시간의 추정이 필요한데 이는 간단한 작업이 아니다. 따라서 보다 손쉽게 취득할 수 있는 데이터에 기반을 두고 일반적인 형태를 갖는 UBG에 대한 연구가 계속될 필요가 있다.

참 고 문 헌

1. Kang W. Lee, "Stochastic Models for Random Request Availability", IEEE Trans. on Reliability, Vol.49, No.1, pp.80-84, March, 2000.
2. M.Tortorella, "Service Reliability Theory and Engineering, I: Foundations", Quality Technology and Quantitative Management, Vol.2, No. 1, 2005, pp.1-16.
3. W. Xie, H. Sun, Y. Cao, and K. S. Trivedi, "Modeling of User Perceived Webserver Availability, IEEE International Conference on Communication, May 2003.
4. M. Tortorella, "Service Reliability Theory and Engineering, II: Models and Examples", Quality Technology and Quantitative Management, Vol. 2, No. 1, 2005, pp. 17-37.
5. M. Kaaniche, K. Kanoun, and M. Martinello, "A User-Perceived Evaluation of a Web Based Travel Agency", International Conference on Dependable Systems and Networks, June 2003, pp. 709-718.
6. C. R. Johnson, Yakov Kogan, Y. Levy, F. Saheban, and P. Tarapore, "VoIP: A Service Provider's Perspective", IEEE Magazine, July 2004, pp. 48-54.
7. R. Keralapura and S. Bhattacharyya, "Service Availability: A New Approach to Characterize IP Backbone Topologies", 12th IEEE International Workshop on Quality of Service, June 2004, pp. 232-241.
8. E. E. Lewis, Introduction to Reliability Engineering: John Wiley&Sons, 1987.
9. E. Parzan, Stochastic Processes: Holden Day.
10. W. Dazhl, Service Reliability: Model, Algorithms and Application, Duke University, 2007.
11. L. Shao, L. Zhang, J.Zhao, B.Xie, and H.Mei, "Towards a User-Perceived Service Availability Metric", 2008 IEEE Internatioinal Conference on Service Computing, 2008, pp.549-550.
12. L. Shao, L. Zhang, J.Zhao, B.Xie, and H.Mei, "User-Perceived Service Availability: A Metric and an Estimation Approach", 2009 IEEE International Conference on Web Services, 2009, pp.647-654.
13. K. Tokuno and S. Yamada, "Markovian Model for User-Perceived Software Availability Measurement with Operation-Oriented Restoration", 7th International Conference on Service System and Service Management, 2010.
14. H. Ozaki, A. Kara, and Z. Cheng, "User Perceived Reliability of M-for-N (M:N) Shared Protection Systems", IEICE Trans. on Information and Systems, Vol. E92-D, No. 3, 2009, pp. 443-450.
15. H. Ozaki, A. Kara, and Z. Cheng, "User-Perceived Availability of Priority Shared Protection Systems", International Journal of Quality and Reliability Management, Vol.28, No.1, 2011, pp. 96-108.



함 영 만 (ham@yiit.co.kr)

1994 서울산업대학교 전자공학과 학사
 1996 한양대학교 공학대학원 전자통신공학과 석사
 2008~현재 서울과학기술대학교 IT정책전문대학원 박사과정(산업정보시스템전공)
 영인 IT 기술사 사무소 대표(정보통신기술사, 전자응용기술사, 전자기기기능장)
 대한민국 무선통신 명장

관심분야 : 정보통신, 무선통신, 방송통신



이 강 원 (kwlee@snut.ac.kr)

1980 서울대학교 공과대학 산업공학과 학사
 1982 서울대학교 대학원 공학석사
 1985 Kansas State Univ. U.S.A 공학박사(산업공학)
 1985 한국전자통신연구원 TDX 개발단
 2012~현재 서울과학기술대학교 글로벌융합 산업공학과 교수

관심분야 : 정보통신, 품질 및 신뢰성, O.R.