

웹기반 정보시스템의 재구성을 위한 항해구조 및 사용자행동 모델링

박학수[†] · 황성하^{**} · 이강수^{***}

요 약

최근의 정보시스템의 전형적인 형태인 웹기반 정보시스템(WBIS)은 계속적으로 재구성되어야만 사용자의 욕구를 만족시킬 수 있으며 수익을 창출할 수 있다. 이를 위해, 시스템의 항해구조를 모델링하고 분석해야 하며 로그파일을 통해 사용자의 행동을 모델링하여 시스템의 재구성시에 활용해야한다. 이에 따라, 본 논문에서는 쇼핑몰을 포함한 WBIS의 재구성(또는 튜닝)을 위한 항해구조의 모델링 및 사용자행동 모델링 방법을 제시한다. 항해구조의 모델링을 위해 구조모델, 상태전이 모델 및 페트리넷 모델 및 분석방법을 제시하였다. 또한, 사용자행동 모델링을 위한 로그분석을 위해, 사용자행동 모델링 알고리즘을 제시하고 이를 구현하였다. 끝으로, 재구성 휴리스틱을 제시하였으며 제시한 방법들은 사례 WBIS에 적용하였다.

Navigational Structure and User Behavior Modeling for Resturcturing of Web-based Information Systems

Park Hark Soo[†], Hwang Sung Ha^{**} and Lee Gang Soo^{***}

ABSTRACT

A Web-Based Information System(WBIS), a typical structure of recently information systems, should be dynamically restructured in order to satisfy user's need and make a profit. Thus, we should analyze and modelize the navigational structure of WBIS and utilize it by modeling the navigational structure of behavior of user through log-file as system restructuring. In this paper, we propose the modeling method for navigational structure and user behavior to restructure WBIS including shopping mall. Also, we suggest the structural model, state transition model, Petri net model and analysis method and analyze and implement modeling algorithm for user behavior to analyze log-file of it. Then, we propose some restructuring heuristic and apply the methods to the example of WBIS.

Key words: 웹기반정보시스템, 웹공학, 소프트웨어공학, 리엔지니어링, 웹 항해모델, 웹 로그분석

1. 서 론

개인용 홈페이지, B2C형태의 쇼핑몰 및 B2B형태의 ebXML기반 e-비즈니스시스템에 이르기까지, 대부분의 정보시스템은 웹기반정보시스템(WBIS: Web-

based Information System)이다. 그러나, 대부분의 WBIS의 경우, 사전조사가 부족하고 체계적인 기술(즉, 엔지니어링) 없이 개발 및 운영함에 따라, 접속의 폭주에 대한 처리 문제, WBIS내의 효율적인 항해 문제, 동적 변경에 대한 처리문제, 확장성 지원문제, 이주문제 및 다양한 개발프로세스의 처리문제와 같은 새로운 문제들이 발생하고 있다. 이와 같은 문제를 해결하지 못한 결과로서, WBIS의 개발기간과 비용의 증가, 웹 개발 프로세스의 신뢰성, 확장성, 호환

본 연구는 한국과학재단 목적기초연구(R05-2001-000-01492-0)지원으로 수행되었음.

[†] 정회원, 한국과학기술정보연구원 선임연구원

^{**} 준회원, 한남대학교 컴퓨터공학과 대학원

^{***} 종신회원, 한남대학교 대학원 컴퓨터공학과 교수

성의 저하와 유지보수 문제가 심각해지고 있다[1,2].

이러한 문제들을 “웹위기(web crisis)”라 하며 1960년대 말의 소프트웨어위기 문제에 비유된다. 소프트웨어위기 문제는 소프트웨어공학을 통해 해결 하듯이 웹위기는 “웹공학”[1,2]을 적용함으로써 해결이 가능하다. 웹공학 분야중에서도 특히, WBIS내의 항해구조와 사용자의 항해패턴을 모델링하고 분석하여 이를 재구성(또는 튜닝, 리엔지니어링)하는 기술은 사용자의 만족과 수익증대에 중요한 영향을 미친다[3].

WBIS의 재구성 문제는 실제 상점 또는 사무실에서의 ‘매장 배치문제’에 비유된다. 즉, WBIS가 최적의 항해구조를 갖도록 재구성함으로써, 사용자가 시스템 내에서 쉽고 빠르게 정보를 검색하거나 기능을 수행할 수 있게 하는 것이다. 이를 위해, 기존의 항해구조를 식별하고 로그파일을 분석하여 사용자 행동 패턴에 적용함으로써, 최적의 효율을 갖도록 사용자 중심의 WBIS로 재구성해야한다[3].

그림 1은 본 논문에서 제시하는 절차를 보인다. 본 논문에서는 WBIS의 항해구조 모델링을 위해, 개선된 상태전이도에 의한 항해구조 모델링 방법 및 패턴리벡에 의한 사용자대화 모델링 및 분석 방법을

제시한다. 또한, 사용자 로그분석을 통한 사용자행동 모델링 알고리즘을 제시하고, WBIS의 재구성 휴리스틱을 제시한다. 끝으로, 사용자행동 모델링 알고리즘의 구현 및 사례연구 결과를 제시한다. 본 논문은 WBIS의 성능평가를 통한 정량적인 튜닝문제는 다루지 않으며 특히, 사용자행동 분석 문제는 고객관계 관리(CRM: Customer Relation Management)[4]를 위한 클러스터링 알고리즘이므로 본 논문에서는 다루지 않는다.

본 논문의 2장에서는 WBIS의 항해구조 모델링 및 사용자행동 모델링에 관련한 연구동향과 문제점을 알아보고, 3장에서는 항해구조 모델링 및 분석방법을 제시하며, 4장에서는 로그분석을 통한 사용자행동 모델링 방법 및 재구성 휴리스틱을 제시한다. 5장에서는 사용자행동 모델링 알고리즘을 구현하여 제시한 방법으로 적용한 사례를 보이며 6장에서는 본 연구와 기존의 연구를 비교평가 한다. 끝으로, 7장에서 결론과 향후연구과제를 제시한다.

2. 관련 연구

WBIS는 90년대 중반까지의 HTML과 CGI에 의한 홈페이지 형태(1세대), CGI와 NSAPI프로그램에 의해 레이아웃과 멀티미디어 디자인을 강화한 형태(2세대), 자바와 ActiveX 및 폼을 강화한 형태(3세대) 및 워크플로우나 ebXML을 지원하는 e-비즈니스 시스템 형태(4세대)로 진화되었다[5]. 특히, 기존의 정보시스템에 비해 2세대 이후의 WBIS는 변경 및 확장성 문제, 부하의 폭증성 문제, 커스터 마이징 문제 및 유지보수 문제 등이 발생하고 있다. 이들 중 ‘유지보수 문제’는 다른 문제들을 해결하지 못했을 때 발생하며 ‘웹위기’의 본질이라 할 수 있다[1]. 이들 문제는 기존의 소프트웨어공학뿐만 아니라 소비자 행동론, 인지론 및 마케팅 등의 학문이 연계된 기술인 ‘웹공학’ 기술을 통해 해결할 수 있다[2].

2.1 항해구조 모델링

본 논문의 주제는 웹공학 중에서 WBIS의 개발방법론을 다루며 특히, 항해구조 모델링 및 분석기술에 해당한다. 항해구조 모델링 및 분석기술은 기존의 정보시스템에서는 볼 수 없는 WBIS만의 문제이며 다음과 같은 연구결과들이 발표되어있다.

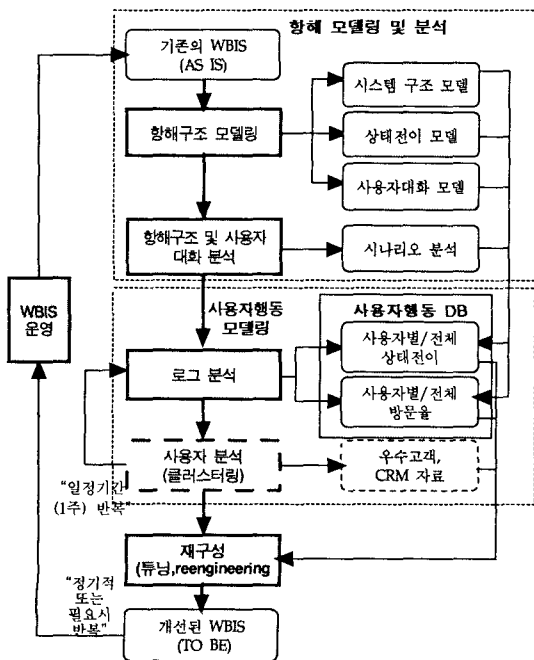


그림 1. WBIS의 재구성 절차의 개요

Web Modeling Language[6]에서는 수직적 항해와 합성 프리미티브 정의를 강조하고 있으며 항해모델은 페이지간의 링크관계로 응용설계자가 명세한다. Torii[7]에서는 객체간의 항해 모드를 사용자정의 항해모드와 미리 정의된 항해모드로 구분하고 있다. W2000[8]에서는 설계자가 항해구조를 모델링시 "follow a link" 링크 연산을 기본연산으로 처리한다.

Relationship Management Model[9]에서는 항해설계단계에서 Relationship Management Data Model의 항해 엘리먼트와 Relationship Management Diagram을 사용한다. Arenus Data Model[10]은 Relationship Management Model을 단순화한 것으로 Navigation Conceptual Model을 이용하여 항해구조를 모델링한다. Object-oriented Hypermedia Design Method[11]은 UML을 통해 항해 클래스 스키마와 항해 컨텍스트 스키마를 설계하며 웹 응용을 객체모델상의 항해 뷰로 간주하며 항해를 위한 기본구성(컨텍스트, 인덱스)을 제공한다.

Object Oriented Hypermedia[12]은 항해접근 다이어그램을 사용하여 항해구조를 모델링한다. 항해접근 다이어그램은 각 사용자가 바라는 항해행위, 객체 population 선택, 항해 객체 순서, 접근의 cardinality를 정의하며 "항해 클래스", "항해 타겟", "항해 링크", "컬렉션" 옵션들을 통해 정의한다. HSDL/SchemaText[13]은 컨텍스트, 구조, 레이아웃 및 항해간의 독립성을 유지하며, 항해구조를 구체화, 부분 및 가시화 항해구조로 구분하며 구조정보를 통해 접근경로와 레이아웃을 자동으로 생성한다.

User Centered Methodology[14]에서 항해 모델링은 첫째, 잠재적 사용자의 톨과 프로파일을 고려하여 항해구조를 형성, 둘째, Way of Navigation은 골달성을 위한 최상의 항해경로이며 사용자 프로파일 개념과 같으며, 셋째, Navigational Semantic Unit은 동일한 골을 위한 Way of Navigation의 집합이다. 또한 사용자프로파일과 AND/OR그래프와 유사한 high-level Goal Reduction Graph을 통해 연산가능 골을 연산 골로 축소하여 프로파일기반 연산 그래프를 획득하고, 연산 골로부터 Way of Navigation을 유도한다. Navigation Node개념은 정보 또는 아이디어의 유닛이며 정보 또는 프로그램을 포함한다.

이러한 연구들은 신규 WBIS의 개발시의 항해구조의 모델링을 위해 제시된 것이므로, WBIS의 실제

사용자로부터의 항해패턴을 고려하지 않고 있으며 기존의 WBIS에 대한 항해구조의 모델링과 시나리오 분석에 대한 지침이 부족하고 사용자대화 에 대한 모델링 지침이 결여되어있다.

끝으로, 본 논문의 접근방법과 유사한 Menasce-Almeida의 연구[3]에서는 기능모델내에서 항해구조와 기능을 '소비자행동 모델 그래프'(CBMG)와 '소비자 방문 모델'(CVM)을 통해 모델링하여 각각 소비자행동을 예측하고 부하모델의 구성정보를 모델링하고 있으며 '클라이언트 서버 대화 다이어그램'(CSID)를 통해 대화를 모델링하고 있다. CBMG는 단순히 상태전이그래프이므로 각 상태내의 세부기능이나 수행시간 등을 모델링하기에 부족하며, CSID는 WBIS내의 서버들간의 대화를 상위수준으로 모델링하므로 WBIS의 구조를 나타내지 못하며 사용자의 병행적이고 세부적인 대화 시나리오를 나타내는데 한계가 있다. 또한, CSID는 정형적이고 수행 가능한 모델이 아니다.

2.2 로그분석 기술

HTTP 프로토콜은 비접속성 방식이므로 세션정보 관리의 어려움이 있으며, 검색엔진과 에이전트로 부터 간접 접속이 많으므로 실제 사용자에 대한 로그 정보를 얻기가 어렵다. 또한, 웹 캐싱으로 인하여 세부적인 사용자로그 정보를 얻기 어렵다[15-17]. 이를 해결하기 위해 페이지내 태깅, TCP/IP 패킷 스니핑[18], 분석프로그램 삽입 및 서버로그분석 방법들[19]과 분석결과를 가시화하는 연구[20]가 활발하다.

한편, 용량이 크며 계속 증가하는 로그파일을 Data Warehouse를 통해 효율적으로 관리하는 방법이 제시되었지만[21] 구체적인 로그파일 DB의 저장구조나 저장 정보량의 축소방법이 제시되어있지 않다. 또한, 본 논문의 주제와 유사한 연구[22]에서는 로그분석 결과를 통해 사용자 패턴을 모니터링하여 웹사이트를 개선하고자 하고 있지만 사용자 패턴의 획득 및 활용방법이 제시되어 있지 않고 웹사이트의 개선 및 재구성에 대한 지침이 부족하다.

2.3 재구성과 튜닝 기술

WBIS의 튜닝은 시스템의 워크로드를 통한 성능평가나 what-if분석을 통해 성능상의 병목을 식별하

여 적절한 성능과 용량을 갖는 하드웨어 또는 소프트웨어 솔루션으로 교체하는 과정이다[3,23]. ReWeb [24]는 프레임이 포함된 페이지간의 구조를 방향성 그래프로 모델링하며 최단경로와 강결합 컴포넌트들을 분석하고 참조패턴을 활용하여 웹의 구조를 경험에 따라 재구성할 수 있도록 지원하는 정적인 재구성도구이다. 이 방법은 항해구조를 상세히 모델링하지 못하며, 사용자행동을 고려한 동적인 재구성을 지원하지 못한다.

3. 항해구조 모델링 및 분석

본 장에서는 WBIS의 항해구조 모델링 방법과 정적특성의 분석방법을 제시한다.

3.1 항해구조 모델링

항해구조 모델링은 WBIS의 항해구조 구조, 상태전이 및 사용자 대화 특성을 정형화 및 도형화한 것이며, 항해구조는 실제 상점 또는 사무실에서 단위 매장(또는 단위 프로세스)들의 배치구조에 해당한다. 본 논문에서는 시스템의 상위수준의 정적 모델링을 위해 WBIS의 구조모델과 상태전이모델을 제시하고, 하위수준의 모델링을 위해 패트리넷을 확장한 BPN(Business Process Net) 모델[25]을 이용하는 방법을 제시한다.

(1) 구조모델

WBIS의 구조모델은 서버와 서버간의 통신을 제어하는 통신망으로 구성된 인트라넷 구조를 보여주는 상위수준의 모델이다. 구조모델에서 '서버'는 둥근 사각형으로 나타내며, 그 내부에는 세부 기능을 기술한다. 서버간의 구조는 화살표와 네트워크(직선으로 표시)를 통하여 상호 연결 구조를 나타낸다. 성능평가를 위해 구조모델에는 각 서버와 네트워크의 하드웨어적 성능(속도, 용량 및 대역폭)과 비용을 포함할 수 있다.

그림 2는 전형적인 WBIS의 구조모델을 보인다. 각 서버와 통신망의 정확한 성능 및 비용은 이미 알려져 있으므로, 고객 측면의 성능(예; 응답시간)과 서버 시스템 측면의 성능(예; 처리량)을 쉽게 구할 수 있다[23].

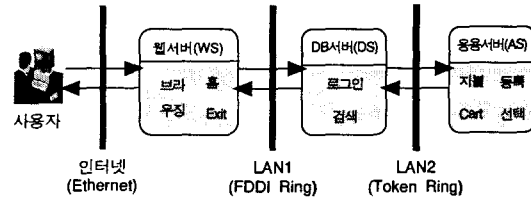


그림 2. WBIS의 구조모델의 예

(2) 상태전이 모델

상태전이 모델은 사용자가 WBIS 내의 기능을 향해진 흔적(즉, 한 상태에서 다른 상태로의 전이)을 정형화한 것이며 그래프형과 매트릭스형 모델이 있다. 본 논문에서는 Menasce-Almeida[3]의 CSID를 확장하고 UML의 상태차트에 "tagged value"를 추가하여 정의하였다.

[정의 1] STD(State Transition Diagram) = <노드, 아크>

- 노드
 - WBIS의 상태를 나타낸다.
 - 특별한 한 개의 노드인 "exit노드"는 모든 노드에 링크되므로 명세를 생각할 수 있다.
 - 2개 이상의 노드를 하나의 "슈퍼노드"로 명세할 수 있다.
 - 상태이름('sname'), 행위('behavior'), 한계시간('deadTime') 및 수행서버('executedBy')는 "tagged value"이며 선택적으로 사용할 수 있다.
 - 'behavior'태그는 표 1과 같이 정의한다.

표 1. STD에서 'behavior'태그의 정의

'behavior' 태그	행동
'entry/action'	상태 시작시 action을 수행
'do/activity'	상태 중간에 activity을 수행
'exit/action'	상태 종료시 action을 수행
'event/action'	상태의 전이를 야기하지 않는 상태 발생시 action을 수행

• 아크: 단방향 또는 양방향 화살표이며 상태 전이의 방향을 나타낸다. UML의 상태차트와는 달리 사용자가 노드를 클릭 할 때가 '이벤트'에 해당하므로

전이 모델에서는 나타낼 수 없는 사용자행동 및 시스템 기능의 병행성 및 비동기성 등을 도형적이고 정형적으로 모델링 및 분석할 수 있다. 부록에는 BPN의 정의를 보이며 세부적인 특성 및 분석방법은 참고문헌[25]에 수록되어 있다.

그림 4는 전형적인 WBIS의 사용자대화를 BPN으로 모델링한 예를 보인다. 각 플레이스는 WBIS내의 '기능'을 모델링 한 것이며, 트랜지션은 fork 및 join 관계를 나타낸다. 일반적인 패트리넷과는 달리 플레이스간에 트랜지션은 생략함으로써, 모델의 가독성이 향상되고 작성이 용이하다. p5에서 p6~p9중 하나가 선택되어 마킹되며(토권의 존재 및 해당 기능의 수행을 의미함) p10은 p6~p9중 하나가 접화 완료되면 마킹된다.

그림 4에서 p6~p9는 각각 '엔트리', '홈', '브라우저' 및 'exit' 기능을 나타내며, p15~p16은 각각 '로그인' 및 '검색기능'을 나타낸다. p21~p24는 각각 '지불', '등록', '장바구니' 및 '선택' 기능을 나타낸다. 또한, p3과 p11은 '인터넷 접속', p12와 p18은 'LAN을 통한 통신기능'을 의미한다. 예를 들어, p4는 "웹서버가 가용함"으로 해석 할 수 있다. BPN의 정의에 따라, 각 플레이스에 기능명, 입력자료타입, 출력자료타입, 수행시간 및 비용값을 부여할 수 있으며 BPN이 모델링한 WBIS의 정적 및 동적 특성들을 분석할 수 있다[25]. 플레이스(예: p6~p9)에는 분기율을 고려할 수도 있다.

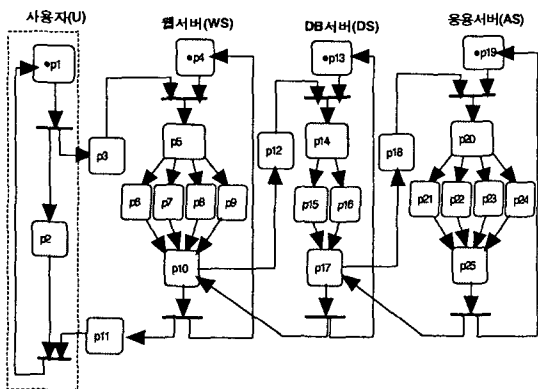


그림 4. BPN에 의한 사용자대화 모델링의 예

3.2 항해구조 및 사용자대화의 분석방법

BPN의 수행규칙[25]에 따라 BPN을 수행하면 도

달성 그래프를 얻을 수 있다. 도달성 그래프는 노드와 아크로 구성되며 노드에는 마킹을 표시하고 WBIS의 모든 항해중의 사용자대화 시나리오와 생존성 정보가 포함되어 있다. 그림 5에서 시작노드(1,4,13,21)로부터 원 위치에 이르는 하나의 경로는 사용자대화 시나리오이며, 사이클이 존재하므로 시나리오의 개수는 무한수이다. 또한, WBIS를 모델링한 그림 5는 모든 노드로 도달 가능하므로, "데드락"이 존재하지 않음을 알 수 있다.

도달성 그래프의 노드는 WBIS의 한 상태를 나타낸다. 예컨대, 노드(2,12,13,19)는 "사용자는 수행완료를 기다리며, 웹서버가 DB서버로 메시지를 보내는 중이고, DB서버와 응용서버는 가용함"으로 해석 된다.

각 플레이스에는 해당 기능의 수행시간이나 비용값을 부여할 수 있으므로 도달성 그래프의 경로(즉, 시나리오)에 대한 전체 수행시간이나 비용 등을 구할 수 있다. δ_k 를 도달성 그래프의 하나의 경로라 하고 γ_i 를 경로상의 한 플레이스의 수행시간이라면, 시나리오의 수행시간 $time(\delta_k)$ 는 다음과 같이 계산할 수 있다.

$$time(\delta_k) = \sum_{i \in \delta_k} \gamma_i$$

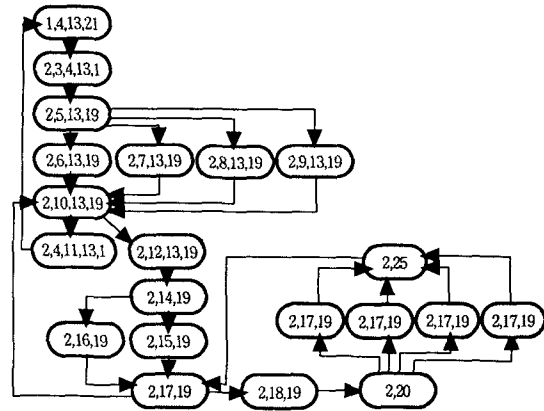


그림 5. BPN에 대한 도달성그래프

4. 사용자행동 모델링 및 WBIS의 재구성

본 단계에서는 WBIS에 대한 사용자 로그를 획득 및 분석하여 사용자별 전체 상태전이율과 기능별 방문율을 구하고, 필요시 클러스터링 결과와 함께 WBIS의 재구성을 위한 자료로 활용한다. 여기서, 사

용자별 상태전이율과 방문율은 사용자행동 모델 또는 '사용자행동 패턴'이라 할 수 있다.

4.1 사용자행동 모델링

사용자행동 모델이란 특정 사용자가 WBIS에 접속하여 행한 행동양식을 특성화 한 것이며 사용자패턴과 같은 의미로 사용한다. 예컨대, 쇼핑몰에서 특정 고객의 구매패턴 및 행동양식은 사용자행동 모델이라 할 수 있다. 그림 6은 사용자행동 모델링 과정을 보인다.

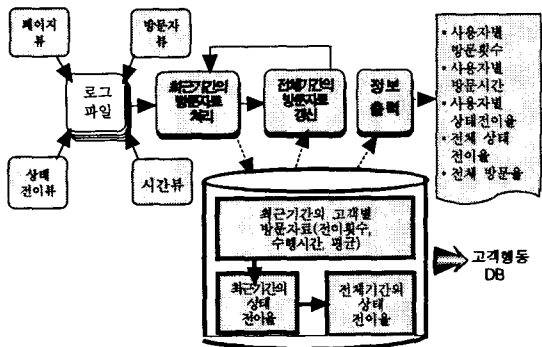


그림 6. 사용자행동의 모델링 과정

(1) 로그파일의 뷰

로그파일에는 많은 정보들이 함축되어 있으며 뷰에 따라 필요한 정보를 추출 또는 유추해야한다. 첫째, '페이지뷰'는 특정한 페이지(또는, 기능)를 누가 사용했는가에 대한 정보를 나타낸다. 둘째, '방문자뷰'는 특정한 사용자가 언제, 어떤 페이지 또는 기능을 사용했는가에 대한 정보를 나타낸다. 셋째, '상태전이뷰'는 한 페이지에서 다른 페이지로 이동한 정보를 나타낸다. 끝으로, '시간뷰'는 사용자가 WBIS에 접속하여 접속을 해제 할 때까지의 시간에 대한 정보를 나타낸다. 본 연구에서는 아파치 로그로부터 각각의 뷰에 해당하는 정보를 추출하여 사용자행동 DB를 구축했다.

(2) 사용자행동 DB구축 방법

사용자행동 DB란 WBIS의 웹서버(예: 아파치 서버)에 접속한 모든 방문기록(즉, 사용자행동 모델링 결과)을 사용자별 세션으로 구분하여 체계적으로 저장한 것이며, Data Warehouse나 CRM을 위한 DB로 사용될 수 있는 "long-term" DB이다. 로그분석시 세

션의 식별 및 로그파일의 관리 문제[15-17,21]는 다음과 같이 해결하였다.

- 세션의 식별문제: 고객별 임계시간을 설정하여 임계시간(예: 30분)내의 로그를 세션으로 한다.
- 항해 기능의 식별문제: 표준 로그파일에는 사용자가 항해한 기능정보가 없으므로, WBIS내의 페이지내 태깅기법을 이용하여 항해된 페이지 정보가 포함된 로그파일을 구성하게 하였다(아파치 서버로그내의 function 태그를 이용).
- 로그파일의 관리문제: 로그파일의 크기를 일정하게 유지하기 위해, 본 연구에서는 그림 7과 같이 일정기간(예: 1주일)내의 로그파일로부터 세션을 식별하고 방문정보만을 추출하여 최근기간의 상태전이율 테이블을 구하고, 이를 전체 상태전이율 테이블에 누적하였다. 따라서, 고객행동 DB의 크기는 고객수에만 비례하고 방문수나 기간과는 상관없이 일정한 크기를 유지하게 된다.

다음은 사용자행동 DB의 구축을 위한 "사용자행동 모델링 알고리즘"이다.

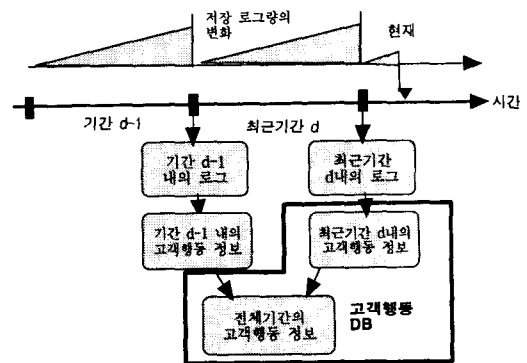


그림 7. 본 연구에서 사용한 로그파일의 관리 방법(화살표는 "고객행동" 정보의 이동방향을 나타냄)

[사용자행동 모델링 알고리즘]

본 알고리즘은 특정 사용자 및 모든 사용자에 대해, 최근기간(예: 1주일)의 상태전이율을 계산하고 이를 이용하여 전체기간의 상태전이율을 구하여 사용자행동 DB에 저장 및 갱신한다.

- 입력: 아파치서버의 공통로그구조(CLF)는 (Host, Ident, Authuser, Date, Request, Status, Bytes)

과 같으며, 이들중 Host, Date 및 Request 필드만 사용하여 소스로그 SL(Source Log)을 구성한다.

$SL = (userIP, demand, execution, function)$

여기서, *userIP*는 Host로부터 유도하며, *demand* (요청시간)은 Date로부터 유도한다. 또한, *execution* (수행 시간)은 *demand*로부터 유도하며(즉, $demand_i - demand_{i-1}$), *function*(요청노드)는 Request로부터 유도한다.

- 출력: 각 고객별 상태전이율 테이블(BRT)
- 자료구조

TT(Term Table) : 최근기간 동안의 전체사용자의 상태전이율 ((0:n × 0:n) 행렬구조임)

$TT(i, 0)$: 노드 *i*의 방문횟수

$TT(0, j)$: 노드 *j*의 평균 수행시간

$TT(i, j)$: 노드 *i*에서 노드 *j*로 분기율

BRT(Branch Ratio Table): 전체기간 동안의 전체사용자의 상태전이율 ((0:n × 0:n) 행렬구조임)

$BRT(i, 0)$: 노드 *i*의 누적 방문횟수

$BRT(0, j)$: 노드 *j*의 평균 수행시간

$BPM(i, j)$: 노드 *i*에서 노드 *j*로 분기율

RVRT(Recent Visit Ratio Table) : 최근기간 동안의 전체사용자의 방문률

$RVRT(i)$: 노드 *i*의 최근 방문률

TVRT(Total Visit Ratio Table) : 전체기간 동안의 전체사용자의 방문률

$TVRT(i)$: 노드 *i*의 전체 방문률

$TRN(n, n, m) = [tt_{i,j,u}]$: 최근기간 동안의 특정 사용자의 상태전이 횟수 (즉, 사용자 *u*가 노드 *i*에서 노드 *j*로 전이한 횟수이며 $1:n \times 1:n \times 1:m$ 행렬구조임)

$EXE(n, n, m) = [exe_{i,j,u}]$: 최근기간 동안의 특정 사용자의 수행시간 합계 (즉, 사용자 *u*가 노드 *i*에서 노드 *j*로 전이한 경우의 수행시간 합계)

$AVG(n, n, m) = [avg_{i,j,u}]$: 최근기간 동안의 특정 사용자의 수행시간 평균 (즉, 사용자 *u*가 노드 *i*에서 노드 *j*로 전이한 경우의 평균 수행시간이며 $[trn_{i,j,u}] \div [exe_{i,j,u}]$ 와 같음)

여기서, *n*은 추가노드인 “exit 노드”를 포함한 전체 노드수 이기도 하며 “exit 노드”의 번호로도 사용

하며 *m*은 사용자수임

- 알고리즘

BRT(*,*,*) 초기화.

For each term do //예, term = 1주일 //

사용자별 행동모델링(*TRN, EXE, TRN*) //사용자별 세션식별, 전이횟수, 수행시간 및 평균 // *TT*(*,*,*) 초기화.

For us = 1 to n do, For i=1 to n do, For j=1 to n do // TT 획득 // $TT(i, j) = TT(i, j) + TRN(i, j, us)$ $TT(i, 0) = TT(i, 0) + EXE(i, j, us)$ $TT(0, i) = TT(0, i) + TRN(i, j, us)$ end_do end_do end_do

tota = 0

For i = 1 to n do tota = tota + TT(0, i) end_do

For i = 1 to n do $CVRT(i) = TT(i) \div tot$ end_do // 최근기간 동안의 방문율 계산 //

totb = 0

For i = 1 to n do // TT 를 BRT 에 누적 //

$BRT(i, 0) = BRT(i, 0) + TT(i, 0)$

$BRT(0, i) = \{BRT(0, i) \times BRT(i, 0) + TT(i, 0) \times TT(0, i)\} \div \{BRT(i, 0) + TT(0, i)\}$

For j = 1 to n do

$BRT(i, j) = \{BRT(i, 0) \times BRT(i, j) + TT(i, j)\} \div \{BRT(j, 0) + TT(i, 0)\}$ end_do

totb = totb + BRT(0, i)

end_do

For i=1 to n do $TVRT(i) = BRT(i) \div totb$ end_do

// 전체기간 동안의 방문율 계산 //

*BRT*에는 전체기간 동안의 상태전이율이 저장됨

end_do

사용자별 행동모델링(*TRN, EXE, TRN*)

*SL*를 *userIP* 순으로 정렬한후 *demand* 순으로 재정렬하여 정렬된 로그 *OL*(Ordered log)를 생성.

For all j in *OL*, do

IF($execution_j \leq threshold$) then *OL* = *OL*

– j end_do //예: 임계시간 = 1800sec//

EL = *OL*.

For each user us do // 사용자별 전이횟수, 시간 및 평균 계산 //

EXE(*,*,*), *TRN*(*,*,*), *AVG*(*,*,*) 초기화.

For each functions (즉, *f1, f2, ..., fn*) in user


```

us do //function은 동일 사용자의 요청노드 집합임
//
TRN( $f_k, f_{k-1}, us$ ) = TRN( $f_k, f_{k-1}, us$ ) + 1.
EXE( $f_k, f_{k-1}, us$ ) = EXE( $f_k, f_{k-1}, us$ ) + executionk
AVG( $f_k, f_{k-1}, us$ ) = EXE( $f_k, f_{k-1}, us$ ) ÷ TRN( $f_k, f_{k-1}, us$ )
end_do
end_do

```

본 알고리즘은 일정한 기간(예: 1주일)마다 수행되어 사용자행동 DB내의 테이블들을 갱신하게된다. 즉, 최근의 1주일간의 로그파일로부터 TRN, EXE 및 AVG를 생성하며 이를 이용해 TT와 RVRT를 구하고, 최종적으로 BRT와 TART를 갱신해 간다.

본 알고리즘은 다음과 같은 시간과 공간 복잡도를 갖는다. 시간 복잡도의 경우, 사용자별 행동모델링은 $O(2*t*c*log_c) + O(t*m*f)$ 이며, 메인 알고리즘은 $O(t \times n^3) + O(3 \times n) + O(t \times n^2)$ 이다. 여기서, n 은 노드수, t 는 구간수(빈도), c 는 접속수(로그파일의 레코드수), m 은 사용자수, f 는 유효로그수이며 솔팅속도는 $2 \times c \times \log_c$ 이다(Quick Sort 알고리즘 적용시). 따라서, 전체시간은 $O(t \times 2 \times c \times \log_c) + O(t \times m \times f) + O(t \times n^3) + O(3 \times n) + O(t \times n^2)$ 이며, $t \ll n \ll m \ll f \ll c$ 이므로(예: $n = 20$ 기능, $m = 1000$ 명, $f = 10000$ 유효접속수, $c = 50$ 만 전체접속수, $t : 1$ 주일), $O(t \times 2 \times c \times \log_c) \approx O(c \times \log_c)$ 가된다. 접속건수 c 의 $c \times \log_c$ 에 비례하며 이는 원시로그의 정렬시간에 해당한다. 따라서, 효과적인 원시로그의 정렬알고리즘이 본 알고리즘의 속도를 좌우한다.

공간(메모리) 복잡도의 경우, 상태전이율을 저장하기 위한 2개의 $(n+1) \times (n+1)$ 테이블과 사용자별 전이횟수 등을 저장하기 위한 3개의 $n \times n \times m$ 테이블 및 방문을 저장하기 위한 2개의 $n \times 1$ 테이블이 요구된다. 여기서, n 은 WBIS내의 노드(상태 또는 기능)수이며 m 은 사용자수이다. 따라서, $O(2n^2m + 2n)$ ($= O(n^2) + O(n^2m) + O(2n)$)의 공간이 소요되며 구간의 크기와는 무관하다.

한편, 본 알고리즘에서는 매트릭스(테이블) 자료구조를 이용했지만 매트릭스들은 WBIS의 노드들의 구조상 희박행렬 형태이므로, 인접 링크드 리스트 자료구조를 이용할 수도 있다. 이 경우, 매트릭스의 희박성 여부에 따라 메모리를 줄일 수 있다.

4.2 고객분석

방문 DB에는 최근의 1주일간의 로그파일로부터 최근기간 동안의 특정 사용자의 상태전이횟수(TRN), 수행시간(EXE), 평균수행시간(AVG)이 저장되어있다. 또한, 최근기간 동안의 전체사용자의 상태전이율(TT), 방문율(RVRT), 상태전이율(BRT)과 방문율(TART)이 저장되어있다.

TRN, EXE 및 AVG은 사용자별 모델(또는 패턴)이며, TT, BRT, RVRT 및 TVRT는 전체사용자에 대한 모델이다. 특히, 사용자별 모델은 '클러스터링'을 통해 사용자 패턴별로 분류하여 WBIS의 개인화 및 마케팅(개인화)을 위한 자료로 활용할 수 있다. 한편, 전체사용자모델은 WBIS의 튜닝(재구성)을 위한 자료로 활용할 수 있다.

4.3 WBIS의 재구성 휴리스틱

정적분석 및 동적분석 결과를 통해 WBIS의 구조와 병목부분을 식별하여 WBIS를 재구성한다. 재구성 문제는 백화점에서 고객별 마케팅을 실시하고 매장을 재배치하여 고객의 동선거리와 구매율을 높이는 문제에 비유된다. 다음은 WBIS의 재구성을 위한 휴리스틱이다.

- 병렬처리: 사용빈도가 높은 노드(즉, 기능)는 병렬처리 하거나 멀티쓰래딩(즉, 데몬수를 늘임)함(예: '검색기능'을 병렬처리함)
- 병합: 사용빈도가 낮은 기능은 하나의 기능으로 통합함(예: '고객인증'과 '로그인기능'을 통합)
- 격리: 고도의 보안성을 요구하는 기능은 방화벽을 설치함('결제기능' 앞에 방화벽을 설치함)
- 재배치: 기능간의 전이가 많거나 유사한 기능을 가지며 서로 다른 서버에 처리되는 기능을 한곳으로 재배치한다(예: '검색기능'과 '주문기능'을 하나의 서버에서 처리).
- 분할: 복합적인 기능을 세부기능으로 분할
- 북마킹: 간접적으로 이동해야하는 빈도가 높은 항해경로에 대해 직접 링크함

그림 8은 재구성 규칙에 대한 BPN 모델링을 보이며 BPN(즉, WBIS)의 튜닝시에 활용할 수 있다. 위와 같은 규칙을 적용하여 시스템을 재구성해 가며 그 결과는 다시 정적분석과 동적분석을 통해 새로운 문

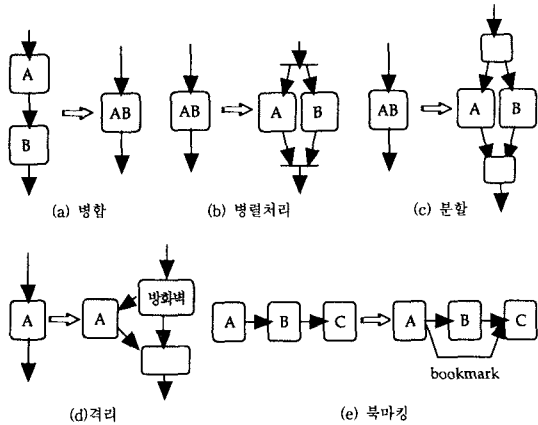


그림 8. WBIS의 재구성 휴리스틱에 대한 BPN 모델링

제점을 발견하여 계속하여 개선해간다.

5. 구현 및 적용 예

본 장에서는 WBIS의 한 종류인 가상적인 쇼핑물 [3]을 대상으로 적용하였다. 또한, '사용자행동 모델링 알고리즘'을 구현하여 가상적인 쇼핑물의 사용자 모델링 업무에 실험적으로 적용하였다.

5.1 항해구조 모델링 및 분석 결과

사례 WBIS에 대한 항해구조 모델링 결과로써, 그림 2는 가상적인 쇼핑물에 대한 구조모델이며, STD와 STM는 각각 그림 3, 표 2와 같다. 또한, 그림 4는 BPN에 의한 사용자대화 모델링 결과를 보이며 그림 5는 BPN에 대한 RG를 보인다. 끝으로, 3.2절에서는 가상적인 WBIS에 대한 항해구조 및 사용자대화의 분석결과들을 일부 보였다.

5.2 사용자행동 모델링 결과

4장에서 제시한 '사용자행동 모델링 알고리즘'을 구현하였고 로그레코드를 생성하여 사례 WBIS의 사용자행동 모델링에 적용하였다. 알고리즘은 C로 구현하였고 GUI는 비주얼 C++ 6.0을 사용하였다. 일반적으로 WBIS의 로그파일은 해당 기관의 대외비에 해당하므로, 본 연구에서는 "아파치" 웹서버의 로그 파일을 바탕으로 하여 그림 9와 같은 13,580건의 로그 레코드를 임의로 생성하여 실험하였다.

```

203.247.40.147 - - [23/Mar/2002:11:34:31 +0900] "GET
/main/main_image/research_down.jpg HTTP/1.1" 304 -
211.105.186.232 - - [23/Mar/2002:12:03:26 +0900] "GET
/main/topframe.htm HTTP/1.1" 200 592
211.105.186.232 - - [23/Mar/2002:12:04:24 +0900] "GET
/main/menuframe.htm HTTP/1.1" 304 -
210.110.71.4 - - [23/Mar/2002:12:34:50 +0900] "GET
/~jhko/images/snow.gif HTTP/1.1" 200 592
210.102.203.128 - - [23/Mar/2002:12:53:31 +0900] "GET
/main/main.htm HTTP/1.1" 200 592
203.242.109.29 - - [23/Mar/2002:13:04:24 +0900] "GET
/~jinbu2000 HTTP/1.1" 301 321
203.242.109.29 - - [23/Mar/2002:13:04:25 +0900] "GET
/~jinbu2000/ HTTP/1.1" 304 -
:
    
```

그림 9. 본 연구에서 사용한 아파치 로그파일의 일부

생성된 로그파일을 사용자행동 모델링 프로그램에 적용한 결과는 그림 10~그림 12와 같다. 또한, 그림 13은 그림 12를 도식화한 것이다. 본 논문에서는 사용자별 방문시간(EXE)과 사용자별 평균수행시간(AVG) 결과를 생략하였다.

In\Out	b	s	a	b	p	t	l	r	x
총합	0	142	0	0	0	0	3	3	1
검색(성)	21	139	0	0	0	2	0	0	1
항해구조(성)	0	141	33	143	3	3	3	3	3
인강우장(성)	21	0	0	139	0	2	0	0	2
지침(성)	0	0	0	0	0	0	0	0	2
선택(성)	0	95	22	98	0	0	0	0	2
로그인(성)	4	0	0	26	0	0	0	0	1
등록(성)	0	0	0	0	0	0	3	0	1
End(성)	0	0	0	0	0	0	0	0	0

In\Out	Trn	h	s	a	b	p	t	l	r	x
평균수행시간	191	94	314	164	421	68	112	237	81	152
총합	149	0	0.95	0	0	0	0	0.82	0.02	0.01
검색(성)	163	0.13	0.85	0	0	0	0.01	0	0	0.01
항해구조(성)	332	0	0.42	0.10	0.43	0.01	0.01	0.01	0.01	0.01
인강우장(성)	164	0.13	0	0	0.85	0	0.01	0	0	0.01
지침(성)	2	0	0	0	0	0	0	0	0	1.00
선택(성)	217	0	0.44	0.10	0.45	0	0	0	0	0.01
로그인(성)	31	0.14	0	0	0.85	0	0	0	0	0.01
등록(성)	10	0.60	0	0	0	0	0	0.30	0	0.10
End(성)	13	0	0	0	0	0	0	0	0	0

그림 10. 최근기간 동안의 특정사용자의 상태전이횟수(Trn)와 전체사용자의 상태전이율(TT) 결과

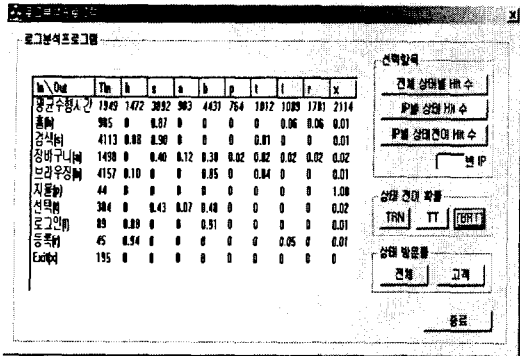


그림 11. 전체기간 동안의 전체사용자의 상태전이율(BRT) 결과

사용자행동 DB는 WBIS의 문제점 개선 및 재설계를 위한 기초자료로 활용될 수 있으며 마케팅 부서에서는 소비자별 마케팅 전략으로 활용될 수 있다. 예컨대, 그림 11은 그림 12처럼 나타낼 수 있으며 검색 기능(s)과 브라우저기능(b)은 최근에는 잘 사용되지 않고 있음을 보인다. 그림 10에서 장바구니(a)에서 지불(p)로 전이할 확률은 전체고객 및 전체기간 동안 1%에 머물고 있으며, 그림 12의 TVRT에서 전체 방문중 지불기능은 0.4%이다. 즉, 구매율(전체 방문당

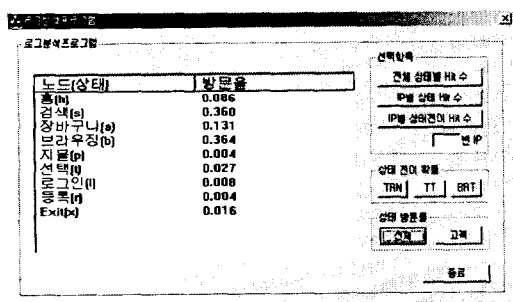
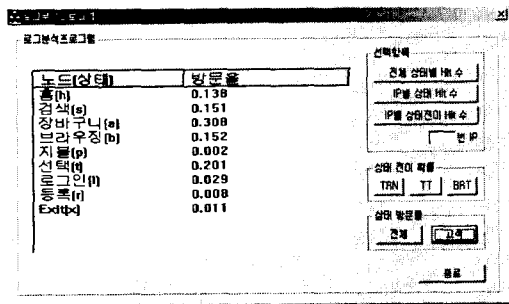


그림 12. 최근기간 동안의 방문율(RVRT)과 전체기간 동안의 방문율(TVRT)

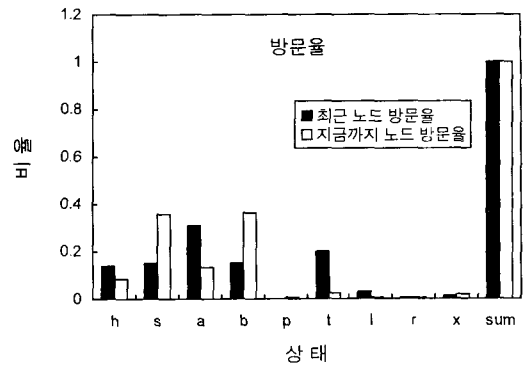


그림 13. 노드 방문율의 변화

구매비율)이 0.4%이므로 WBIS는 개선의 여지가 있는 것으로 판단되어 마케팅을 강화하고 WBIS의 구조를 튜닝 할 필요가 있다.

5.3 재구성예

그림 14는 재구성 휴리스틱 중 재배치 규칙을 적용하여 인증서버(PS)를 추가 설치하여 '고객인증' 및 '지불인증' 기능을 재배치하였고 분할 규칙을 적용하여 '검색' 기능을 '제품검색'과 '고객검색'으로 분리한 결과를 보인다.

재구성된 WBIS에 대한 상태전이도, BPN 및 도달성그래프는 본 논문에서 제시한 방법을 통해 생성할 수 있으며, 재구성된 WBIS를 구현하여 운영중 발생한 로그파일로부터 사용자행동 알고리즘 및 프로그램을 통해 사용자행동 DB를 구축하여 재구성의 효과를 평가할 수 있다.

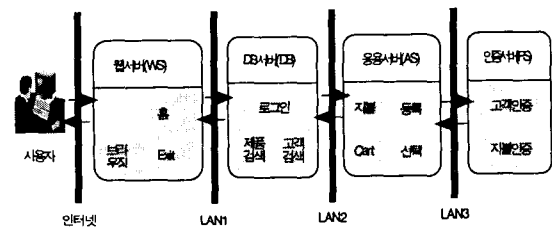


그림 14. 재구성된 WBIS의 구조모델

6. 평가

본 연구의 분야인 웹공학의 경우 아직 소프트웨어 공학에 비해 학문적, 기술적 체계가 미흡하며 프로토타입 수준의 다양한 방법론 및 도구가 제시되어 있을

뿐이다. 따라서, 본 방법과 기존의 방법을 정량적으로 비교 평가하기는 어려우며 본 장에서는 정성적인 비교 평가를 실시하고자 한다.

첫째, WBIS의 항해구조 명세모델의 관점에서 기존의 연구[6~14]들은 WBIS의 요구사항분석 및 설계단계에서 항해구조의 모델링을 위해 제시된 것이므로, WBIS의 실제 사용자로부터의 항해패턴을 고려하지 않고 있으며, 기존의 WBIS에 대한 항해구조의 모델링과 사용자 시나리오의 모델링 및 분석 기능이 부족하다. 특히, Menasce-Almeida[3]의 CBMG는 단순히 상태전이 그래프이므로 각 상태내의 세부 기능이나 수행시간 등을 모델링하기 어려우며 CSID는 WBIS내의 서버들간의 대화를 상위수준으로 모델링하므로 WBIS의 구조를 나타내지 못하며 사용자의 병행적이고 세부적인 대화 시나리오를 나타내는데 한계가 있다. 또한, CSID는 정형적이고 수행가능한 모델이 아니다.

본 논문에서 항해구조 모델로써 제시한 상태전이도는 CBMG와 UML의 상태전이도에 태그를 추가하므로써 상태전이율 뿐아니라 상태내의 기능 및 기능수행시간 등을 모델링할 수 있다. 특히, CSID의 단점을 보완한 BPN[25]은 WBIS의 고유특성인 병행성 및 비동기성 등을 정형적이고 도형적으로 모델링하고 도달성 그래프를 이용해 사용자대화 시나리오를 분석할 수 있다. BPN은 패트리넷의 변형이며 “수행가능형” 모델이다. 패트리넷[26, 27]은 1964년 이후로 각 분야에서 성공적으로 활용되어온 정형모델이다. 따라서, BPN은 WBIS의 시뮬레이션 모델로도 활용될 수 있을 것이다.

둘째, 사용자 모델링을 위한 로그분석 방법의 관점에서, 기존의 연구들[21,22]에서는 사용자행동패턴의 획득 및 로그파일 DB의 저장구조나 저장 정보량의 축소방법이 설명되어있지 않으며 WBIS의 재구성에 대한 지침이 부족하다. 본 논문에서 제시한 사용자행동 모델링 알고리즘은 ‘웹서버 로그분석 방법’과 ‘태깅 방법’을 결합하여 로그를 분석하고 상태전이 횟수, 상태전이율 및 방문율을 구하여 사용자행동 DB를 구축하는데 활용된다. 알고리즘에서는 로그파일의 크기를 줄이는 방법을 적용하여 일정한 크기의 DB가 유지될 수 있도록 하고있다.

셋째, WBIS의 재구성(또는 튜닝, 리엔지니어링)의 관점에서, 기존의 연구에서는 what-if분석에 의

한 성능의 튜닝을 위한 방법을 주로 제시하고있으며 [3,24] WBIS의 항해구조의 재구성 방법에 대한 연구가 부족하다. 본 논문에서 제시한 모델 및 재구성 휴리스틱들은 궁극적으로 WBIS의 재구성을 위한 것이다.

끝으로, 본 논문에서 새롭게 연구된 내용은 다음이 요약된다. 1) WBIS의 항해구조 모델인 상태전이도(STD), 상태전이매트릭스(STM)을 제시하였다. 2) 사용자대화 모델인 BPN을 제시하였다. 3) 사용자행동 모델링 알고리즘을 제시하였다. 4) WBIS의 재구성(또는 튜닝, 리엔지니어링)을 위한 휴리스틱을 제시하였다.

7. 결 론

본 논문에서 다룬 WBIS의 재구성을 위한 항해구조 및 사용자행동 모델링 및 분석문제는 현재에 당면한 웹위기를 다소 극복할 수 있는 방법이다. 특히, 항해구조 모델로서 제시한 상태전이모델 및 BPN 모델은 WBIS의 본질적인 특성, 즉, 병행성 및 비동기성을 모델링하고 분석할 수 있으며 수행가능한 정형적 모델이므로, 기존의 항해모델[3,5-14]의 대안으로 사용될 수 있다. 또한, 사용자행동 모델링 알고리즘은 로그분석을 효율적으로 수행할 수 있으며, 사용자행동 DB를 구축하는데 적용할 수 있다. 본 논문에서 제시한 재구성 휴리스틱은 모델링 및 분석결과와 함께 WBIS를 재구성하는데 활용될 수 있을 것이다.

본 논문에서는 다루지 않은 사용자행동 DB의 활용문제 즉, 클러스터링 알고리즘 및 이를 이용한 고객관계 관리문제[4]는 본 연구의 후속연구로 이루어져야한다. 또한, 본 연구에서는 WBIS상에서의 사용자의 항해스키마만을 다루었으나, DB스키마 및 자원스키마를 동시에 고려하는 통합스키마의 개발과 사용자 모델링(즉, 로그분석)의 정확도를 높이는 기술 개발이 요구된다.

참 고 문 헌

- [1] A. Ginige and S. Murugesan, “Web Engineering: An Introduction,” *IEEE Multimedia*, Vol.8, No.1, pp.14-18, Jan-Mar. 2001.
- [2] A. Ginige and S. Murugesan, (ed.), “Special

- Issue in Web Engineering," Part 1 and Part 2, *IEEE Multimedia*, Vol.8, No.1, and No.2, Jan-Mar and April-June. 2001.
- [3] D. Menasce and V. Almeida, *Scaling for E-business: Technologies, Modes, Performance and Capacity Planning*, Prentice-Hall, 2000.
- [4] B. Mobasher, et al., "Automatic Personalization Based on Web usage Mining," *Communication of the ACM*, Vol.43, No.8, 2000.
- [5] T. Powell, "Web Site Engineering: Beyond Web Page Design," *Powell Internet Consulting*, Oct. 1997 (<http://www.pint.com/present/webeng/sld001.htm>).
- [6] S. Ceri, et al. "Web Modeling Language (WebML): A Modeling Language for Designing Web Sites," *9'th International WWW conf.*, 2000.
- [7] S. Ceri, P. Fraternali, S. Paraboschi, "Data-driven One-to-one Web Site Generation for Data-intensive Applications," *Proc. of the 25th VLDB*, 1999.
- [8] L. Baresi, et al. "From Web Sites to Web Applications: New Issues for Conceptual Modeling," *Lecture Note in Computer Science*, Vol. 1921, 2000, pp. 89-100.
- [9] T. Isakowitz, et al., "RMM: A Methodology for Structured Hypermedia Design," *Communications of ACM*, Aug. 1995.
- [10] G. Mecca, et al., "Aranues in the Era of XML," *IEEE Data Engineering Bulletin*, Sep. 1999.
- [11] D. Schwabe, et al., "Engineering Web Applications for Reuse," *IEEE Multimedia*, Jan.-Feb. 2001, pp.20-30.
- [12] J. Gomez, C. Cachero, O. Pastor, "Conceptual Modeling of Device-independent Web Applications," *IEEE Multimedia*, April-June, 2001, pp. 26-39.
- [13] M Kessler, A Schema based Approach to HTML Authoring, *4'th Int WWW Conf.*, Dec. 1995.
- [14] C. Gnaho, F. Larcher, "A User Centred Methodology for Complex and Customizable Web Application Engineering," *1'st ICSE Workshop on Web Engineering*, May 1999..
- [15] G. Fleihman, "Web Log Analysis: Who's Doing What, When?," *Web Developer Magazine*, Vol. 2, No. 2, May/June 1996(<http://www.webdeveloper.com>)
- [16] B. Krishnamurthy and J. Rexford. "Software Issues in Characterizing Web Server Logs," *Proc WWW-1998*, 1998.
- [17] "Assessing Web Site Usability form Server Log Files," *White Paper, Tec-Ed, Inc.*, Dec, 1999 (<http://www.teced.com>).
- [18] A. Feldmann, "BLT: BiLayer Tracing of HTTP and TCP/IP," *Proc. WWW 2000 conf.*, 2000.
- [19] 김형택, 민옥길, *효과적인 인터넷 마케팅을 위한 웹로그분석*, 도서출판 비비컴, 2001.
- [20] J. Pitkow and K. Bharat, "WebViz: A Tool for Access Log Analysys," *Proc. WWW-1994*, 1994.
- [21] "Delivering Enterprise Scalibility," *White Paper, Webtrends Corp.*, Dec. 1999 (<http://www.pi.danet.de>).
- [22] M. Drott, "Using Web Server Logs to Improve Site Design," *ACM Proceedings on the 16'th Annual International Conference on Computer Documentation*, pp43-50, Sep, 1998.
- [23] P. Killela, *Web Performance Tuning*, Oreiley, 1998.
- [24] F. Ricca, P. Tonella, "Understanding and Restructuring Web Sites with ReWeb," *IEEE MultiMedia*, Vol.8, No.2, pp40-51, Apr-Jun. 2001.
- [25] 장수진, 최상수, 이강수, "웹기반 비즈니스 프로세스로의 리엔지니어링과 명세를 위한 BPN 모형", *한국정보처리학회 논문지(심사중)*, 2002.
- [26] J. Peterson, *Petri Net Theory and Modeling of System*, Prentice-Hall, 1981.
- [27] K. Jensen and G. Rozenberg(eds.), *Highlevel Petri nets - Theory and Application*, Springer-Verlag, 1991.

부록. BPN의 정의

본 논문은 BPN(Business Process Net) 모델을 단순히 사용하는 것이므로, BPN에 대한 상세한 점화규칙 및 모형화 방법들은 참고문헌 [25]를 참고해야한다. 여기서는 BPN의 구조적 정의만을 기술한다. BPN은 패트리넷의 구조상의 복잡성을 단순화하고 Activity diagram을 확장한 레이블된 방향성 그래프이며, 그 정의와 각 구성요소는 다음과 같이 정의한다.

$$BPN = (P, T, A, TK, Mo)$$

- $P = \{p_{so}, p_l, \dots, p_n, p_{si}\}$ 은 비즈니스 '활동'을 정의하는 플레이스의 집합이며, 끝이 둥근 사각형으로 표기한다. p_{so} 와 p_{si} 는 각각 소스와 싱크 플레이스이며, 비즈니스 활동의 시작과 종료를 모형화한다. p_{so} 와 p_{si} 는 제외한 플레이스(P')에는 활동명, 활동시간, 활동비용, 입력 및 출력자료타입이 매핑된다. 즉,

$$P' \rightarrow (Activity_Id \times In_Data_type_List \times Out_Data_type_List \times Activity_Time \times Activity_Cost)$$

여기서, Activity_Name는 스크립트 타입이며 In_Data_type_List과 Out_Data_type_List는 자료의 타입(예: XML-Schema 또는 DTD) 리스트(0개 이상)이다. Activity_Time과 Activity_Cost 3개의 실수벡터타입이다. 즉, $\langle a, b, m \rangle, a, b, m \in \mathbf{R}$ 를 나타낸다.

- $T = \{t_{fork}, t_{join}\}$ 는 비즈니스 활동간의 포크 및 조인관계를 정의하는 트랜지션의 집합이다.

- t_{fork} (포크 트랜지션)은 후속 비즈니스 활동의 동시시작(concurrent initialization)을 모형화하며 막대로 표기한다.
- t_{join} (조인 트랜지션)은 사전 비즈니스 활동의 비동기적 결합(asynchronous join)을 의미하며 막대로 표기한다.

- $A \subseteq (P \times T) \cup (T \times P) \cup (P \times P)$ 플레이스와 트랜지션 플레이스와 플레이스를 연결하는 아크(arc)의 집합이며 비즈니스 활동의 순서 의미하

며 화살표 또는 직선의 끝을 작은 원으로 표기한다. 직선의 끝을 작은원으로 표시한 아크를 '금지아크'(inhibitor arc)라하며 부정(NOT)의 의미를 나타낸다. 특히, 플레이스 간을 연결하는 레이블된 아크는 '트랜지션아크'(transitional arc)라 명명하며 트랜지션의 개념이 함축되어 있다. 한 플레이스로 입력 및 출력되는 2개 이상의 트랜지션아크는 각각 OR 및 선택관계를 나타낸다.

- TK는 토큰이며 플레이스의 활성화를 표시한다.
- Mo: 초기 마킹

'BPN모듈' BPNm은 BPN에서 Mo, p_{so} 및 p_{si} 를 제외한 부분집합이며 BP내에서 하나의 행위자(agent)에 의해 처리되는 부분을 모형화한다.

$$BPNm \subseteq [BPN - (Mo, p_{so}, p_{si})]$$

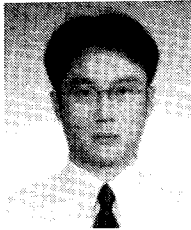
BPN은 활동간의 AND, OR, NOT논리를 표시할 수 있으며 구조상 Activity diagram과 유사하지만 BP의 활동간의 순서관계 뿐 아니라, 활동의 수행시간 및 비용, 활동에서 입력 및 출력되는 자료구조 및 자료흐름을 모형화 할 수 있다. 표준 패트리넷과는 달리 포크 및 조인 트랜지션을 제외하고는 플레이스간에 트랜지션을 표시하지 않고 '트랜지션 아크'로 표시한다. 즉, '플레이스 중심'의 패트리넷이다. 이로써, 트랜지션과 아크의 수가 반정도로 줄어들므로, 가독성이 향상된다.

박 학 수



1989년 한남대학교 전자계산공학과(학사)
 1991년 한남대학교 대학원 컴퓨터공학과(석사)
 1997년~현재 한남대학교 대학원 컴퓨터공학과 박사과정
 1991년~현재 한국과학기술정보

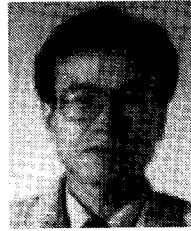
연구원 선임연구원
 관심분야: 소프트웨어공학(신뢰도), 분산컴퓨팅, 멀티미디어응용, 차세대 네트워크 응용기술(QoS, MPLS), 보안공학



황 성 하

2001년 한남대학교 컴퓨터공학과
졸업(학사)
2001년~현재 한남대학교 대학원
컴퓨터공학과 석사과정

관심분야: 소프트웨어 공학, 웹 엔지니어링, 정보보호 및
위협분석, 정보시스템 보안



이 강 수

1981년 홍익대학교 컴퓨터공학과
졸업(학사)
1983년 서울대학교 대학원 전산
학과 졸업(이학석사)
1989년 서울대학교 대학원 전산
학과 졸업(이학박사)
1985~1987년 국립대전산업대학

교 전자계산학과 전임강사
1992~1993년 미국일리노이대학교 객원교수
1995년 한국전자통신연구원 초빙연구원
1998~1999년 한남대학교 멀티미디어학부장
1987~현재 한남대학교 컴퓨터공학과 정교수
관심분야: 소프트웨어공학, 병행시스템 모델링 및 분석,
정보보호시스템 평가, 멀티미디어교육 커리
클