

연상된 사용자 경험정보 축척 및 분석을 위한 AUX 모델

AUX Model for restoring and analyzing Associative User Experience informations

류춘열*, 양해솔**

호서대학교 벤처대학원 IT응용기술학과*, 호서대학교 벤처대학원**

Chun-Yeol Ryu(cy_ryu@shoseo.ac.kr)*, Hae-Sool Yang(hsyang@hoseo.edu)**

요약

현재 IT 분야는 응용을 수행하는 처리 장치가 소형화, 고성능화되고, 센서 기술의 비약적 발달로 인해 다양한 스마트 기능 실현이 가능해지고 있다. 특히, 축적된 사용자 정보를 기반으로 다양한 분석 및 예측을 시도하는 UX(User Experience) 분야는 UI(User Interface)를 중심으로 최근 연구가 활발히 진행되고 있다. 그러나 실세계에서 행동하는 사용자 경험을 효과적으로 분류하고 저장하는 표준화된 형식의 공통 프레임워크에 대한 연구는 아직 미비하다. 본 논문에서는 실세계에서 센싱되어 해석되는 다양한 사용자의 행동정보를 체계적으로 저장 및 참조하기 위한 정보 표현 모델로 AUX(Associative User Experience) 모델과 처리구조를 제안하였다. 이러한 표현모델과 처리체계 분리는 인문적인 사용자 정보의 특성과 기술적인 처리체계의 독립을 통해 AUX 모델 적용 및 처리체계에 전문성과 생산성 및 유연성을 보장한다. AUX 모델은 관심 있는 사용자의 다양한 동작 정보를 확장된 E-TCPN 모델을 이용하여 표현하였다. 그리고 AUX 모델을 응용에서 참조할 수 있도록 XML을 이용한 자료구조로 표현하였다. XML로 표현된 AUX 모델은 응용에서 처리할 수 있는 AUX 정보처리 아키텍처를 설계하고, 이를 적용하여 성능 분석을 수행하기 위해 VOD 서버의 스트리밍 트래픽을 할당하는 MPP 알고리즘을 개선하여, AUX가 처리되는 과정을 나타내었다. 그리고 성능을 분석하기 위해 VOD의 MPP 트래픽 할당 기법에 적용하고, 동작을 시뮬레이션 하였다. AUX 모델을 적용한 MPP 트래픽 할당 알고리즘은 적용하지 않은 알고리즘에 비해 재생 편차가 10.41% 향상되었음을 알 수 있었다. 성능분석 결과는 사용자의 접근 매체 유형과 접근 콘텐츠 정보를 AUX로 변환하여, 동적 트래픽 할당을 위한 MPI, CPI 계산 결과가 스케줄링에 반영되어 재생 성능이 향상되었음을 알 수 있었다.

■ 중심어 : | 사용자 경험 | AUX 모델 | E-TCPN 모델 | MPP 트래픽 할당 기법 |

Abstract

In the IT industry, processing units of IT applications are getting smaller and high efficient. Furthermore, the realization of various smart functions is highly feasible now due to advances in sensing technology. The service infrastructures on high efficient and compact mobile devices are applied to various areas. These also could be possessed by users and is built into the devices. Currently, studies on the UX(User Experience) field to attempt an analysis and prediction of user's information are continuing with reference to the UI(User Interface). However, research on the common framework of classification and storing the user-information, and standardization of form has not been attempted yet. In this study, we proposed the AUX(Associative user Experience) model and process structure to store various empirical data by users. The AUX model expressed a diversity of user's empirical data using extended E-TCPN model. And also, we expressed the data structure using XML with reference to the application of AUX model. This expressed model and separation of process structure guarantee its specialty, productivity and flexibility through the humanistic characteristics of users and the independence of technical process structure. The AUX model maps out the AUX information process architecture and expressed the process with the improved MPP algorithm, to analyze of its performance. The simulation of movements applying to MPP traffic allocation of VOD is used to analyze of its performance. The playback deviation of MPP Graphic Allocation Algorithm where the AUX model was applied was improved by 10.41% more than the one where it was not applied. As a result of that, playback performance has improved due to the conversion of AUX with accessing media, content of users and dynamic traffic allocation such as MPI and CPI.

■ keyword : | UX(User Experience) | AUX Model | E-TCPN Model | MPP Traffic Allocation Algorithm |

I. 서론

현재 IT 분야는 응용을 수행하는 처리 장치가 소형화, 고성능화되고, 센서 기술의 비약적 발달로 인해 다양한 스마트 기능 실현이 가능해지고 있다. 국내 선풍적인 인기를 누리고 있는 스마트폰 역시 이러한 기술기반을 통해 실현된 구체적인 결과물로 인식되고 있다. 과거 정부 주도로 많은 경제적인 부담을 안고 U-지능공간을 실현하려고 노력하였으나 큰 성과가 없었다. 이에 반해 스마트폰은 인간의 편의성을 극대화하여 사용자의 요구에 따라 급속하게 스마트폰 장치와 4G 인프라가 사용자와 서비스 공급자를 통해 전 지역에 확대 보급되고 있다. 이것은 정부가 목표로 하는 U-지능공간과 유사한 서비스 인프라가 기업과 사용자 주도하여 확립된 것이다.

이러한 고성능 소형 모바일 서비스 인프라는 사용자가 소유하거나, 보이지 않게 내장될 수 있다. 이는 과거 고가화 된 소형 장치의 비현실적인 제작비로 인해 실현되기 힘든 분야의 한계를 뛰어 넘어 인간 중심적인 처리를 수행하기 위한 다양한 기술에 대한 연구 개발로 이어지고 있다. SVC(Scalable Video Coding) 알고리즘 기반의 3차원 가속도 센서를 이용하여 사용자의 동작을 추정하는 기술, Zigbee를 이용하여 실내 사용자의 위치추적, 압전센서를 통해 신경정신과에서 사용자의 스트레스 정도를 산정하기 위해 심박 변화율 HRV(Heart Rate Variable)를 이용한 해석 기술 등 다양한 인간 행동, 상태 정보를 측정하는 기술이 완성되고 있다.

따라서, 각 분야별 사용자의 다양한 동작정보를 센서를 통해 물리적으로 측정되는 방식이나, 응용에서 처리하는 표현 방식에 따라 논리적으로 부분적 정보를 저장하고 분석하는 기법은 광범위하게 연구되고 있다. 그러나 실세계에서 행동하는 사용자 경험 정보를 효과적으로 분류하고 저장하여 필요한 응용분야에 표준화된 형식으로 유연하게 전달해주는 공통 프레임워크에 대한 연구는 아직 시도되고 있지 않거나, 연구가 활발하지 않다.

본 논문에서는 실세계에서 센싱되어 해석되는 다양

한 사용자의 행동정보를 체계적으로 저장 및 참조하기 위한 정보표현모델로 AUX(Associative User Experience) 모델과 처리구조를 제안하였다. 제안된 표현모델과 처리체계 분리는 인문적인 사용자 정보의 특성과 기술적인 처리체계의 독립을 통해 AUX 모델 적용 및 처리체계에 전문성과 생산성 및 유연성을 보장한다.

AUX 모델 및 처리구조를 통해 사용자 행동 이력을 분석하여 다양한 분석과 예측을 시도하는 UX기반의 UI 분야와 사용자 성향에 따른 시스템 처리의 반응적 연동분야에 적용시킬 수 있다. AUX 모델은 관심 있는 사용자의 다양한 동작 정보를 확장된 E-TCPN 모델[4]를 이용하여 표현한다. 그리고 AUX 모델을 응용에서 참조할 수 있도록 XML을 이용한 자료구조로 표현한다. XML로 표현된 AUX 모델은 응용에서 처리할 수 있는 AUX 정보처리 아키텍처를 설계하고, 이를 적용하여 성능 분석을 수행하기 위해 VOD 서버의 스트리밍 트래픽을 할당하는 MPP(Media Presentation for Preference) 알고리즘[7]을 개선하여 AUX가 처리되는 과정을 나타내었다.

본 논문의 구성은 2장에서 AUX 모델을 위한 E-TCPN 모델을 확장하고, AUX 모델의 XML 표현과 처리를 위한 AUX 정보 처리 아키텍처를 설계하였다. 그리고 AUX 모델의 적용 및 성능 분석을 위해 VOD 서버를 위한 MPP 트래픽 할당 스케줄링 알고리즘에 적용하고, 3장에서 결론을 요약하였다.

II. 본론

1. AUX 모델의 개요



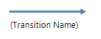
AUX 모델은 사용자의 경험적 정보를 바탕으로 한 통계적 연관관계를 추론할 수 있도록 표현된 모델로 Petri-Net 모델[1-3] 기반의 E-TCPN[4]를 확장한 모델이다. AUX 모델은 사용자의 동작 정보와 사물의 동작 정보를 Place와 Transition으로 표현하였다. AUX 모델에 Place와 Transition 상태 공간 해석을 통해 사물과 사용자의 경험상태, 경험상태간의 전이(Transition), 경험상태간 전이에 따른 시간, 각 경험상태와 전이의

가중치로 표현될 수 있다. AUX 모델을 구성하는 개체에 대한 심볼과 표현을 [표 1]에서 나타내고 있다.

AUX 모델 구성요소인 Place는 사람의 연령, 감정상태, 건강상태, 위치 등 Property, Attribute 등을 포함한 State를 표현한다. Transition은 Place 상태에서 전이가 되는 Action, 전이의 원인이 되는 외적 환경 요인 등 Place의 상태 전이가 발생하는 원인을 표현한다. Time Expression은 Place와 Transition의 시간 표현, 임계시간 등을 표현할 때 사용된다. Weight Expression은 Place와 Transition의 의미를 가지는 가중치를 표현할 때 사용된다.

위와 같은 AUX 모델의 심볼들을 이용하여 동작 추론을 위한 사물과 인간의 다양한 최소 단위의 동작 상태를 Place와 Transition의 상태 공간으로 표현한다. 그리고 전체 동작을 표현하는 AUX 모델 공간에서 사용자의 경험 정보의 횡수, 유사 동작 행위를 검색할 수 있다. 이러한 사용자의 동작 추적을 통해 다양한 행위 및 결과에 대한 경험치가 산출된다. 산출된 경험정보는 사용자를 대상으로 하는 자동화된 최적 제어 동작에 대한 결정, 다수 사용자의 경험정보를 기반으로 한 서비스 추론을 할 수 있다.

표 1. AUX 모델의 심볼 정의

Name	Symbol	Description
Place		관심 있는 개체에 대해 측정시간에 Capture된 경험상태
Union Place		Place와 Transition을 포함하는 Place의 집합
Transition		(Transition Name)으로 인한 이전 경험상태와 현재 경험상태간 전이
Time Expression	$(Sym./Trs.)^n$	Place 혹은 Transition의 시간 표현
Weight Expression	$(Sym./Trs.)^{2n}$	Place 혹은 Transition의 가중치 표현

2. 실시간 프로세스 공간 표현을 위한 E-TCPN

사용자의 경험치를 표현, 추적하여 경험상태의 전이를 연상하기 위해서는 시간적 표현이 가능해야 한다.

그래서 AUX 모델은 실시간 표현을 위한 E-TCPN 모델을 이용하여 사용자 동작 공간을 표현한다.

CCSR 기반의 확장된 Petri-Net인 E-TCPN(Extended-Timed Constraints Petri-Net)은 기존 처리동작을 나타내는 프로세스 공간 표현 모델에 비해 시간 중심의 계산과 상태를 표현하는 Place의 명세 구조가 간단하다. 그러므로 전체 작업의 동작 과정을 표현하고 Place의 공간 탐색이 용의한 장점이 있다[4].

E-TCPN은 Place와 Transition에 최대·최소 시간제한과 시간간격(Duration)을 가진 6-Tuple로 정의된다 [4].

$$N=(P, T, F, A, D, M)$$

- N : {P, T, F, A, D, M}인 Tuple들의 집합
- P : {P1, P2, ..., Pn}인 Place의 집합
- T : {T1, T2, ..., Tn}인 Transition의 집합
- F : {P, T} 또는 {T, P} 순서쌍에서 가능한 아크(arc)의 집합
- A : a{n, ex(d), d}의 집합으로 n은 동기화 되어야 하는 대상 Place의 수, ex(d)는 동기화 처리 시간, d는 동기화가 완전히 처리되고 점화(Fire) 되어야 하는 시간 간격
- D : 시간 간격(Duration) d의 집합
- M : m-vector를 가진 마킹(marking), $\{M(p_1), \dots, M(p_j), \dots, M(p_m)\}$ 의 집합. 여기서, $(M(p_j))$ 는 Place 내에 Token의 수이고, M_0 는 초기 마킹

그림 1. E-TCPN의 Tuple 정의

경험상태 공간의 명세하기 위해 E-TCPN은 각 경험상태 공간 P와 Q에 대한 항목(Term)을 [표 2]와 같이 BNF 문법으로 나타낼 수 있다.

표 2. BNF로 표현된 E-TCPN의 명세 구문

$P: nil \mid r(d) \mid \partial(d) \mid a(n, ex(d), d) \mid P+Q \mid P;Q \mid P \parallel Q \mid fw(P, d) \mid e(P, d)$

또한, E-TCPN을 이용하여 3개의 프로세스로 이루어진 프로세스 공간을 [그림 2]와 같이 명세할 수 있다[8].

```
P1 ::= a(0, ex(0), 0); p2; a(0, ex(0), 0); p3
```

그림 2. E-TCPN 명세 구문을 이용한 프로세스 명세 표현

그리고, CCSR 기반의 E-TCPN 명세구문의 경험상태 공간에 대한 P와 Q는 아래의 그림과 같이 표현된다.

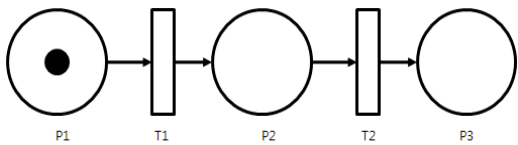


그림 3. 경험상태 공간 P의 E-TCPN 표현

그러므로 E-TCPN 모델을 이용하여 AUX 모델의 시간 공간과 처리 동작을 명세하거나 도시할 수 있다.

3. AUX 모델 처리를 위한 XML 명세

AUX 모델은 경험상태 공간을 시각적으로 표현하고 응용 처리를 수행하기 위해 XML 형태로 변환된다. [표 3]은 E-TCPN 기반으로 명세된 AUX 모델의 XML 명세 태그에 대한 정의를 나타내고 있다.

표 3. AUX 모델의 XML 태그 정의

Tag	Description
AUX	AUX 모델 표현 문서의 범위
N	P, T, F, A, D, M 태그 Grouping
P	Place 표현
T	Transition 표현
F	P와 T의 순서쌍 표현
A	경험상태 공간의 시간 동기화 표현
D	시간간격 표현
M	Place 내에 마킹 표현
NIL	경험공간 종료 표현
R	소요 시간 표현
B	지연 시간 표현
ADD	경험공간의 선택 표현
SEQ	경험공간의 순차 조합 표현
PAL	경험공간의 병렬 조합 표현
SW	임계시간내 시작표시를 위한 시간 구성자 표현
FW	임계시간내 종료표시를 위한 시간 구성자 표현
EXT	임계시간내 정시표시를 위한 시간 구성자 표현

위와 같이 AUX 모델은 응용 처리를 위해 XML 문서로 변환된다.

4. AUX 정보 처리 아키텍처

AUX 모델은 다양한 분야에서 표현되는 사용자의 단위 동작을 이용하여 규칙화된 사용자 경험정보(User Experience)를 생성한다. 생성된 AUX 모델 기반의 사용자 경험정보를 참조하여 응용 처리하기 위해서는 규칙화된 사용자 경험정보를 Capture해서 응용 처리를 위한 비즈니스 로직까지 도달하는 순서를 공통적으로 표현할 수 있다.

AUX 처리를 위한 아키텍처 구성은 기능적 역할에 따라 5개의 Layer로 구분된다. Node Layer는 정보 수집을 위한 독립개체를 처리하는 계층, Interface Layer는 Node Layer에서 발생하는 단위 데이터를 AUX 처리를 위한 변환 인터페이스를 제공하는 계층, Application Layer는 처리를 위한 비즈니스 로직을 구현하는 계층, Middleware Layer은 AUX 정보를 처리하기 위한 상주형 미들웨어를 구현하는 계층, Repository Layer는 AUX 정보를 분류 및 저장 공간과 운용성을 제공하는 계층이다. 이와 같은 5개의 계층을 통해 AUX 정보를 처리하기 위한 구체적인 구성 개체는 [그림 4]와 같이 나타낼 수 있다.

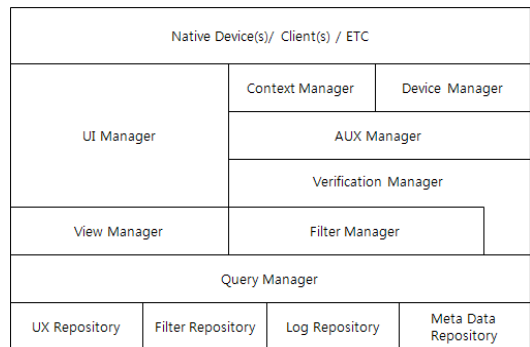


그림 4. AUX 정보 처리 구성도

AUX 처리를 위한 5계층은 [그림 5]와 같이 AUX 정보 처리 구성 모듈과 대응된다.

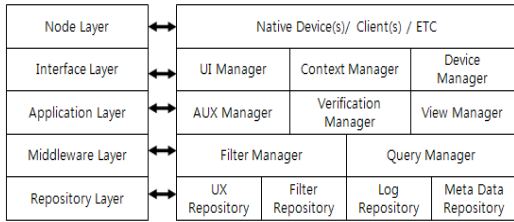


그림 5. AUX처리 구성 스택과 처리 모듈간 매핑

AUX 정보 처리 동작은 자연계에서 정보 수집을 위해 센서 등의 독립된 장치인 Native Device, 정보 수집 로컬 단말기인 Client 등의 정보 수집 노드들로 구성된 u-지능공간을 통해 AUX 정보를 수집한다. 수집된 정보는 의미 종속적인 단위 정보로 Context Manager가 가공 수집하며, 장치 종속적인 단위 정보는 Device Manager가 가공 수집한다. Context Manager와 Device Manager로부터 수집된 사용자 경험 단위 정보는 AUX Manager를 통해 정형화된 AUX 모델 정보로 변환 표현된다. AUX Manager를 통해 정형화된 사용자 경험 정보는 Verification Manager를 통해 수집된 논리적 정보와 물리적 정보에 대한 검증을 수행한다. 검증된 AUX 정보는 응용 처리를 위해 관심 있는 AUX 정보만을 Rule 기반의 추론기관을 내장한 Filter Manager가 관심 있는 AUX 정보만 추출한다. 추출된 관심 있는 AUX 정보는 Query Manager를 통해 UX Repository에 저장된다. Query Manager는 UX Repository에 저장된 사용자 경험 정보를 응용에서 요구 시 추출하여 View Manager에게 전달한다. View Manager는 UX Repository에 저장된 관심 있는 정보의 단위 I/O 정보 블럭이다. UI Manager는 관심 있는 I/O 정보 블럭을 사용자에게 표현한다.

[그림 6]에서 AUX 정보 처리 과정을 전체 구성도로 나타내고 있다.

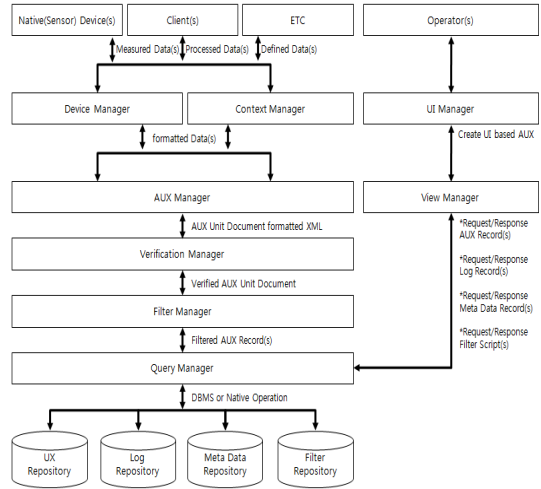


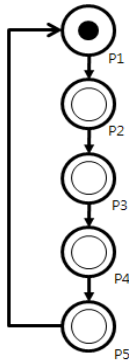
그림 6. AUX 처리를 위한 정보 처리 구성도

5. 구현 및 실험 결과 분석

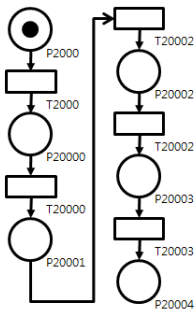
5.1 AUX 모델을 이용한 MPP 스케줄러 구현

MPP기반의 스케줄링 기법은 재생 스트리밍을 요청한 매체의 선호도와 콘텐츠의 선호도에 기반한 실시간 트래픽 할당 처리 알고리즘이다. MPP기반 알고리즘은 [6]에서 제안한 구간재생량(Short-term Popularity Pattern)에 기반한다. [7]에서는 재생할 모든 콘텐츠의 인기도를 구간재생량에 기반하여 측정하고 측정된 인기도를 바탕으로 각 콘텐츠가 프록시의 캐쉬 메모리에 최적으로 할당되도록 구현되었다. 본 논문에서는 MPP기반의 알고리즘에서 트래픽 할당 정도를 나타내는 MEDIACONVERTINGRATE 상수[7]를 매체 선호 인덱스 MPI(Media Preference Index)와 콘텐츠 선호 인덱스 CPI(Contents Preference Index)로 분리하여 적용하였다. MPI 인자는 사용자가 VOD 서버에 콘텐츠의 스트리밍 요청 매체 정보의 접근 횟수를 기반으로 계산된다. 그리고 CPI 인자는 사용자가 VOD 서버에 요청 콘텐츠의 접근 회수를 기반으로 산정된다. [그림 7]의 트래픽 할당 알고리즘은 기존 MPP 트래픽 할당 알고리즘의 MEDIACONVERTINGRATE 상수를 사용자 접근 정보를 기반으로 계산된 MPI와 CPI를 이용하여 보완하였다.

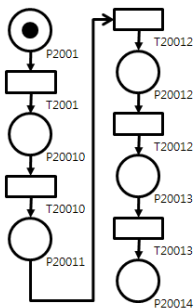
AUX 모델을 적용한 MPP 트래픽 할당 알고리즘의 기존 MCR(Media Converting Rate) 상수계수를 getTopRate 메소드를 통해 AUXDB에 저장된 ServiceClient j의 MPI와 CPI 계수값을 추출하여 평균 값으로 내림차순 정렬한 후, j번째 계수값을 반환하여 사용자의 접근 빈도 정보를 MPP에 반영한다. MPI와 CPI를 추출하기 위해 AUX 모델을 위한 Petri-net 공간 명세는 [그림 8][그림 9]와 같이 표현된다.



(a) MPP 알고리즘의 전체 처리 공간 명세



(b) MPP 알고리즘의 MPI 처리 공간 명세



(c) MPP 알고리즘의 CPI 처리 공간 명세

그림 8. MPI와 CPI 추출을 위한 AUX 모델

[그림 8]에서 명세된 MPI AUX 모델과 CPI AUX 모델의 XML 명세는 [그림 9]와 같이 나타내었다.

```
<AUXINFO>
<AUXMETA Title="CPI">
<AUXTYPE>MEDIA</AUXTYPE>
<AUXCOUNT>10</AUXCOUNT>
</AUXMETA>
<AUXNODEINFO SEQ=0>
<IDENT TYPE="INTERNET">192.168.0.1</IDENT>
<MEDIATYPE>PC</MEDIATYPE>
<DATASOURCE TYPE="MOV">
<NAME>KBSNEWS</NAME>
<CONTENTTYPE>NEWS</NAME>
<TIMECONSTRAINT>SoftRealTime</TIMECONSTRAINT>
<REQTRF UNIT="BPS">(1024*768*4*30*COMPRESSRATE)</REQTRF>
<DATASOURCE>
<SCHINFO TYPE="FW">
<CONDITION>CONTINUE</CONDITION>
<DURATION UNIT="SEC">4</DURATION>
<SCHRESULT>SUCCESS</SCHRESULT>
</SCHINFO>
</AUXNODEINFO>
<AUXNODE NAME="P2001">
<EXT>
<P Name="P2001"/><T Name="T2001"/>
<P Name="P20010"/><T Name="T20010"/>
<P Name="P20011"/><T Name="T20011"/>
<P Name="P20012"/><T Name="T20012"/>
<P Name="P20013"/><T Name="T20013"/>
<P Name="P20014"/>
</EXT>
</AUXNODE>
</AUXINFO>
```

(a) CPI AUX 모델의 XML 표현

```
<AUXINFO>
<AUXMETA Title="MPI">
<AUXTYPE>CONTENT</AUXTYPE>
<AUXCOUNT>4</AUXCOUNT>
</AUXMETA>
<AUXNODEINFO SEQ=0>
<IDENT TYPE="INTERNET">192.168.0.2</IDENT>
<MEDIATYPE>SMARTTV</MEDIATYPE>
</AUXNODEINFO>
<AUXNODEINFO SEQ=1>
<IDENT TYPE="INTERNET">192.168.0.2</IDENT>
<MEDIATYPE>MOBILE</MEDIATYPE>
</AUXNODEINFO>
<AUXNODEINFO SEQ=2>
<IDENT TYPE="INTERNET">192.168.0.2</IDENT>
<MEDIATYPE>PC</MEDIATYPE>
</AUXNODEINFO>
<AUXNODEINFO SEQ=4>
<IDENT TYPE="INTERNET">192.168.0.2</IDENT>
<MEDIATYPE>BEAMPROJECT</MEDIATYPE>
</AUXNODEINFO>
<AUXNODE NAME="P2000">
<EXT>
<P Name="P2000"/><T Name="T2000"/>
<P Name="P20000"/><T Name="T20000"/>
<P Name="P20001"/><T Name="T20001"/>
<P Name="P20002"/><T Name="T20002"/>
<P Name="P20003"/><T Name="T20003"/>
<P Name="P20004"/>
</EXT>
</AUXNODE>
</AUXINFO>
```

(b) MPI AUX 모델의 XML 표현

그림 9. AUX 모델의 XML 명세

AUX가 적용된 MPP 알고리즘은 AUX 정보처리 아키텍처에서 처리 과정은 [그림 10]과 같이 처리되었다.

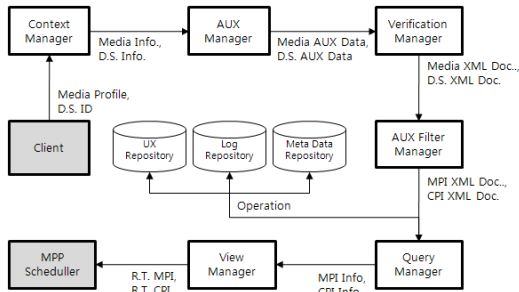


그림 10. MPI와 CPI의 AUX 처리 흐름도

AUX를 이용한 MPP 기반의 트래픽 할당 알고리즘은 단일 모듈로 멀티미디어 스트리밍 서버 내에 구현되었다. 클라이언트는 다양한 재생 환경에 대한 정보를 가진 Profile로 구성된 PC 환경의 시뮬레이터로 구현하였다. 그리고 스트리밍 서버와 클라이언트 PC는 인터넷 환경의 네트워크로 연결되었다. [그림 11]은 실험을 위한 장치 구성을 표로 나타내고 있다.

구성매체	장치형태	CPU	RAM	NIC 전송률
스트리밍 서버	Server PC	Dual-Core 3.0G	4G	1Gbps
PC 재생기	Desktop PC	P4-3.0G	2GByte	100Mbps

그림 11. Test Bed 구성

[그림 12]에서 AUX를 이용한 MPP 기반의 트래픽 할당 알고리즘을 적용한 응용물에 대한 화면을 보여주고 있다.

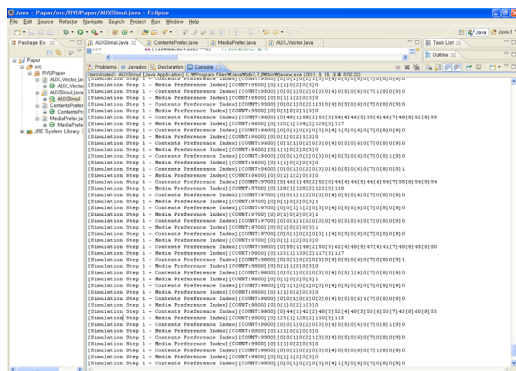


그림 12. 트래픽 할당 응용의 실행화면

5.2 실험 및 성능 분석

실험을 위해 콘텐츠 선호 인덱스와 매체 선호 인덱스를 추출하기 위해 5명의 사용자와 10개의 콘텐츠에 대해 4개의 매체를 이용하여 총 10000번 동안 콘텐츠에 접근하도록 시뮬레이션 했다. [표 4]는 시뮬레이션 조건을 요약하고 있다.

표 4. AUX 획득을 위한 시뮬레이션 조건

시뮬레이션 수	10000
시뮬레이션 조건	Random
User 수	5
Media 수	4(DTV, Mobile, Beam Project, PC)
Contents Category 수	7(News, Movie, Drama, Sports, Music, Entertainment)
Contents의 실시간성	RT(RealTime)/NRT(Not RealTime)

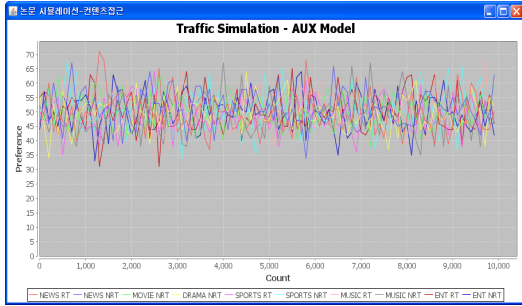
실험 환경을 위한 시뮬레이션 조건 변수는 다음과 같이 요약되었다.

표 5. AUX 획득을 위한 시뮬레이션 조건

서버 트래픽	12.5Mbyte(100Mbps)
데이터 소스 샘플링	1024*780, 4bit, 30FPS
엔코딩 형태	Mpeg 2(TS Stream)
데이터소스/크기(Kbyte)	News_RT.mpg / 27,235, News_NRT.mpg / 26,713, Movie_NRT.mpg / 22,771, Drama_NRT.mpg / 22,049, Sports_RT / 22,033, Sports_NRT / 19,885, Music_RT / 26,735, Music_NRT / 35,354, Entertainment_RT / 252,785, Entertainment_NRT / 200,991

스트리밍 서버는 사용자가 이용하는 매체를 통해 콘텐츠가 요청되면 최대 100Mbps(12.5Mbyte)까지 트래픽을 할당 할 수 있도록 구성하였다.

[표 5]와 같은 조건으로 AUX를 위한 콘텐츠 선호 인덱스와 매체 선호 인덱스 추출을 위한 시뮬레이션 결과가 [그림 13]과 같이 나타났다.



(a) 콘텐츠 선호 인덱스



(b) 매체 선호 인덱스

그림 13. 각 User별 선호도 AUX

[그림 13]의 (a) 시뮬레이션은 10개의 콘텐츠에 대해 5명의 사용자가 임의의 접근한 횟수 집계를 통해 콘텐츠 선호도의 추이를 시뮬레이션 되었다.

[그림 13]의 (b) 시뮬레이션은 10개의 콘텐츠에 대해 5명의 사용자가 임의의 접근한 매체의 횟수 집계를 통해 매체 선호도의 추이를 시뮬레이션 되었다. 특히, 선호 매체 추이 변화에 따라 AUX를 이용한 스케줄링 성능과 AUX를 이용하지 않은 스케줄링 성능의 비교를 위해 PC와 모바일의 매체 선호도가 대치되도록 시뮬레이션 하였다. 시뮬레이션 전반부는 PC의 선호도가 모바일의 선호도보다 높도록 설정되었다. 시뮬레이션 후반부는 모바일의 선호도가 PC의 접근횟수보다 높도록 설정하였다.

[그림 14]는 AUX를 적용한 MPP 스케줄러의 성능을 평가하기 위해 전송 대역폭에 따른 매체 재생 프레임 수의 변화를 나타내고 있다.

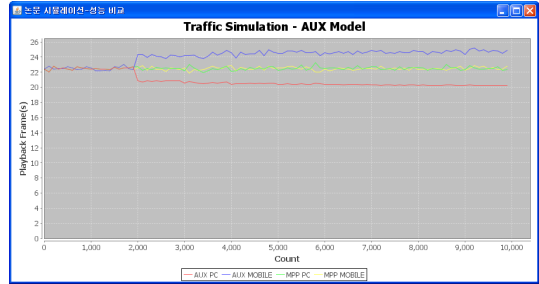


그림 14. PC/Mobile 매체 재생 성능 비교

AUX를 적용한 MPP 스케줄러는 AUX를 적용하지 않은 MPP 스케줄러에 비해 트래픽 전송 시뮬레이션 후반부로 진행할수록 사용자의 재생 프레임 수가 늘어났다. 특히, 성능 평가를 위해 의도적 접근율이 교차되도록 시뮬레이션한 PC와 모바일 부분에서 2000부터 재생 프레임의 편차율이 발생하였다.

이러한 편차율 변화는 시뮬레이션 초반부에 임의의 설정한 매체 및 콘텐츠 선호 상수값과 실제 사용자의 접근 선호도가 유사하여 크게 차이가 나지 않았다. 그러나 시뮬레이션 수행 횟수가 5000번이 넘어가면서 사용자의 PC 매체 이용횟수는 20에서 40회 사이로 줄어들었지만, 모바일 매체의 이용횟수는 220에서 260회 사이로 늘어나 매체 선호도가 증가하였다. 그래서 AUX 정보 유지와 스케줄링 동작 처리의 부하는 상수 선호도에 비해 정비례되므로 AUX를 적용한 MPP 스케줄러가 AUX를 적용하지 않은 MPP 스케줄러에 비해 최대 4 프레임의 재생 편차를 나타내었다.

[그림 14]에서 각 사용자에 대한 프레임 재생 주요 구간 분석을 통해 AUX를 적용한 MPP 스케줄러와 AUX를 적용하지 않은 MPP 스케줄러의 트래픽 전송에 따른 프레임 재생 편차는 10.41%로 분석되었다. 그러므로 AUX 비적용 스케줄러 스트림의 결과 21.56 프레임 재생률에 비해, AUX 적용 스케줄러 스트림은 24.74 프레임의 평균 재생률을 가지므로 3.18개의 더 많은 프레임이 재생되었음을 실험결과를 통해 알 수 있었다.

III. 결론

본 논문에서는 사용자로부터 발생하는 다양한 경험

정보를 통합적으로 저장하기 위해 AUX 모델을 제안하였다. AUX 모델은 관심 있는 사용자의 다양한 발생정보를 확장된 Petri-Net 모델을 이용하여 표현하였다. 그리고 AUX 모델을 기반으로 응용 처리를 위한 미들웨어 아키텍처를 설계하고, 이를 MPP를 이용한 트래픽 스케줄러에 적용하고 성능을 비교 평가하였다.

매체 선호 인덱스를 기반으로 각 사용자에 대한 프레임 재생 주요 구간 분석의 결과, AUX 기반의 MPP 스케줄러는 기존 MPP 스케줄러에 비해 프레임 재생 편차가 10.41% 향상되었음을 알 수 있었다. 그 이유로는 기존 MPP 스케줄러의 경우, 매체 선호도가 상수로 고정되어 있으므로 매체 접근 선호도가 실시간으로 반영되지 않았기 때문이다. 반면, AUX 모델을 적용한 MPP 스케줄러는 접근 매체 선호도에 따라 전송 매체의 재생 특성에 따른 적응적 스트림 변환 및 전송을 통해 각 요청 사용자별 재생 프레임의 수가 늘어났다.

향후 연구에서는 AUX 모델을 실시간 VOD 스트리밍 서버에 적용하여 VOD 서버에 AUX 인자를 결정하고, 실세계의 처리 성능을 분석한다.

참 고 문 헌

[1] G. Bruno, A. Castella, I. Pavesio, and M. P. Pescarmona, "A New Petri Net Based Formalism for Specification, Design and Analysis of Real-Time Systems," Proceeding Real-Time Ssystems Symposium, pp.294-301, 1993.

[2] J. E. Coolahan and N.Roussopoulos, "Timing requirement for time-driven systems using augmented Petri nets," IEEE Trans. Software Eng., Vol.SE-9, pp.603-616, 1983(9).

[3] N. G. Leveson and J. L. Stolzy, "Safety analysis using Petri nets," IEEE Trans. Software Engineering, Vol.SE-13, No.3, pp.386-387, 1987(3).

[4] 최동한, "E-TCPN을 이용한 CCSR 적용기법과 스케줄가능성 분석", 부경대학교 졸업논문, 1999.

[5] 서상진, "실시간 자동화 제어 엔진의 설계 및 구현", 부경대학교 졸업논문, 2001.

[6] 홍현옥, 박성호, 정기동, "시리즈 비디오 데이터의 접근 패턴에 기반한 프록시 캐싱 기법", 멀티미디어학회논문지, 제7권, 8호, pp.1066-1077, 2004.

[7] 진현준, 서상진, 박노경, "스트리밍 데이터를 위한 적응적 실시간 트래픽 할당 기법", 전자공학회논문지, 2006.

[8] 한성호, "실시간 스트리밍 서비스를 위한 E-TCPN 기반의 지능형 스케줄러 설계 및 구현에 관한 연구", 호서대학교 졸업논문, 2005.

저 자 소 개

양 해 술(Hae-Sool Yang)

정희원



- 1975년 2월 : 홍익대학교 전기공학(공학사)
- 1978년 8월 : 성균관대학교 정보처리학과(공학석사)
- 1991년 3월 : 日本 오사카대학교 정보공학과 S/W공학 전공(공학박사)

- 1975년 5월 ~ 1979년 6월 : 육군중앙경리단 전자계산실 시스템분석장교
- 1980년 3월 ~ 1995년 5월 : 강원대학교 전자계산학과 교수
- 1986년 12월 ~ 1987년 12월 : 日本 오사카대학교 객원연구원
- 1995년 6월 ~ 2002년 12월 : 한국소프트웨어품질연구소 소장
- 1999년 11월 ~ 현재 : 호서대학교 벤처전문대학원 교수
- 2010년 3월 ~ 현재 : 호서대학교 글로벌창업대학원 원장

<관심분야> : S/W공학(특히, S/W품질보증과 품질평가, 품질감리 및 컨설팅, SI, S/W프로젝트관리, 품질경영.

류 춘 열(Chun-Yeol Ryu)

정회원



- 1985년 2월 : 경상대학교 사범대학 수학교육과(이학사)
 - 1990년 3월 : 日本 요코하마국립대학 전자정보공학과(공학석사)
 - 2000년 3월 ~ 2003년 2월 : 부경대학교 일반대학원 전자계산학과(박사수료)
 - 2008년 3월 ~ 2011년 8월 : 호서대학교 벤처전문대학원 IT응용기술학과(박사수료)
 - 1990년 4월 ~ 2004년 3월 : 日本 관리공학연구소 연구원
 - 1999년 3월 ~ 2000년 8월 : 영산대학교 컴퓨터공학과 전임교수
 - 2000년 8월 ~ 현재 : 서울호서전문학교 근무
- <관심분야> : S/W공학, IT응용기술