

# GPIS Tracking 메커니즘의 시각구조 Mashup Web

안성은\*, 김정중\*\*

## Mashup Web of Visual Structure using the Mechanism of GPIS Tracking

Sung-Eun An\*, Kim, Jung-Joong\*\*

### 요 약

인터넷 웹 콘텐츠는 정적, 동적 데이터로부터 Mashup 서비스까지 진화되어왔다. 최근에 화두가 되고 있는 Mashup 서비스는 서로 다른 서비스들을 혼합하여 새로운 형태의 정보를 제공할 수 있는 콘텐츠를 생성해낸다, 하지만 서비스 구성에 있어 시스템 자원의 활용보다는 콘텐츠 자체에만 초점을 맞추고 있는 현실이다. 논 본문에서는 Mashup 서비스를 구성하는 정책과 사용자 친화적인 화면 구성방안을 제안한다.

### ABSTRACT

Web Contents have been evolved from statics, dynamics contents to Mashup service. In recent years, Mashup which can create new contents by mixing different services is topic on the web service. But, Mashup is only concentrate on making new contents to provide for client. In this paper, we propose the policy of composing services and client-friendly UI.

### Key Word

Mash-up, GPIS, Tracking Mechanisms, Visual Structure, Web

### 1. 서 론

인터넷 웹(Internet Web)은 인류가 의사소통의 도구로서 컴퓨터와 네트워크 통신으로 정보와 데이터 유통을 혁신시킨 실체로서, 정보화사회(情報化社會)를 이

루게 구축했으므로 없어서는 아니 될 사회적 기반시설(SOC: Social Organization Structure)이다.

그러나 이 웹에서 정보와 데이터 유통 기술은 하드웨어적 기능으로 기술(技術)의 한계성에 다다랐고, 그 유통 성능의 향상 문제 보다는 시각적 인지구조로 바

\* 강원대학교 컴퓨터학과(ahnse@kangwon.ac.kr)

\*\* 강원대학교 컴퓨터학과 교수(vincent@kangwon.ac.kr)

#논문번호 : KIIECT2009-04-09

#접수일자 : 2009.10.23

#최종논문접수일자 : 2009.11.25

른 참의미를 전달할 표현 기술(記述)이 문제로 나타났다.

그러므로, 이 연구는 인터넷 웹에서의 참 뜻을 바로 소통될 수 있도록 가시적인 시각구조에 협력 학습의 형태로 진화되어질 Mashup Web의 디지털 멀티미디어 표현기술(記述)을 연구하고, 그 설계와 구현을 하려는 것이다.

이 연구는 웹기반 서비스(SOA: Service Oriented Architecture) 접근방법으로 구성하려는 방법을 제시하며, 각각의 서비스가 또 다른 서비스들을 호출하고, 이를 사용하려는데 필요한 시스템 자원을 절약하기 위한 방법으로 서비스들의 다형성을 기반하며 단편화 방법과 즉시성(caching) 방법을 이용한 서비스 구성을 하는 모형이다.

그리고, 이 정보와 데이터를 이용하려는 사람들이 명확하고 직관적으로 이해할 수 있도록 표현을 하는 기술(記述)의 방법으로 시각화구조에서 Mashup Web을 구현한다.

그러므로, 이 논문의 서론에 이어서, 제 2장에서는 문헌연구와 조사를 하며, 제 3장에서는 본 논문의 서비스 정책을 위한 CFP framework에 대한 소개를 한다.

그리고, 제 4장에서는 시각구조의 Mashup Web을 기술(記述)하는 모형을 설계하고 구현한다. 마지막으로 제 5장에서 결론을 맺는다.

## II. 관련연구

이 장에서는 동적 웹 콘텐츠를 서비스 하기 위한 시스템 자원을 절약하기 위한 기존연구들을 살펴본다. 또한, CFP framework에 연결될 수 있는 방법으로 기존의 연구들을 분류하였다. 비록 모든 분류법이 직관적으로 맞지 않아도 적절한 추정치에 분류하였다.

### 1) 서버 (Server-Side)

FastCGI 방법은 지속적인 HTTP 요청을 처리하기 위한 방법으로, CGI 프로세서들은 첫 번째 HTTP 요청을 서비스한 직후 중단되도록 설계되어졌다[1]. 유사한 방법으로 콘텐츠 데이터베이스를 위한 서버 프로세스가 지속적인 연결을 하는 연구도 있다[2]. 또한, 서로 다른 미들웨어 시스템을 비교분석한 연구도 있었는데, C 기반의 미들웨어 시스템의 성능이 Java 기반의 것 보다 성능이 우수하였다. 하지만, Java 기반의 미들웨어 시스템은 프로그램용이성, 공개성 등의 장점이 있다[3]. 동적 웹 콘텐츠를 위한 Java 기반의 미들웨어 시스템들을 성능 및 프로그램용이성 관점에서 철저히 비교 분석한 연구도 있다[4].

### 2) 단편화 (Fragmentation)

초기 단편화 연구인 SSI(Server Side Include)는 페이지 요청 될 때마다 페이지의 특정 부분이 고립되거나 재 생성되는 문제점을 가지고 있다. 예를 들면 카운터, 서버 시간, 요청된 파일에 대한 정보 등이다. 뒤를 이어서 동적 웹 페이지

를 특정 조각들로 나누어 템플릿 파일에 의해 재구성되는 페이지형성 방법이 제안되었다[5].

그림1은 템플릿파일을 이용한 동적 웹 페이지 단편화의 전형적인 형태이다.

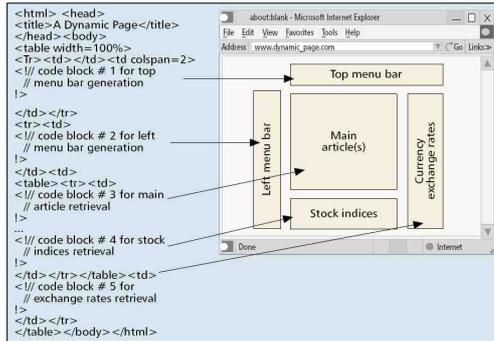


그림 1. 전형적인 단편화 형태

Fig 1. A typical form of fragmentation

DWC(동적웹콘텐츠)의 임의 단편화를 지원하는 방법도 있었으나, 페이지를 세분화하여 서버측에서만 캐싱을 제공한다. 또한 임의 단편화화 조각들의 즉각적인 실행을 지원하지만 캐싱 정책을 사용하지 않는 방법도 있다.

앞선 방법은 포털, 뉴스 사이트와 같이 콘텐츠의 생성이 데이터베이스의 변화에 의한, 즉 사용자와의 상호작용이 비교적 덜한 응용프로그램에 적합하다. 후자의 방법은 콘텐츠 생성이 사용자의 선택에 따라 변하는 전자상거래와 같은 상호작용이 활발한 웹 응용프로그램에 적합하다. 다시 말하면 재활용이 가능한 데이터의 비중에 따라 응용프로그램의 서비스 정책도 변화되어야한다.

### 3) 캐싱 (Caching)

동시에 많은 사용자에게 캐싱된 콘텐츠를 서비스하기 위한 연구들이 있었다. 정적인 HTML 조각들, XML 조각들, 데이터베이스 쿼리결과들의 캐싱을 제한한 연구는 엄격한 제한점과 특정 프로그램에서만 가능하다는 단점이 있으며, 임의 단편화를 지원하지 않기 때문에 캐싱정책을 동적페이지 임의의 조각에 적용할 수 없다. 좀 더 유연하고 일반적인 단편화 캐싱정책에 관한 연구도 있었으며, 이 방법은 단편화 조각을 태그로 감싸는 방법으로 템플릿 파일에 부분적으로 캐싱을 적용한다. 또한 ASP, PHP, CFP와 같은 스크립트 언어들도 서버 측에 단편화 캐싱을 지원한다.

CSI(Client Side Include)는 ESI와 HPP를 혼합한 방식으로 클라이언트 브라우저에 ActiveX를 이용하여 플러그인을 설치해야한다. 클라이언트 측에서 요청된 DWP(동적웹페이지)와 관련된 템플릿 파일과 콘텐츠 조각들을 캐시로 보관하고 ESI방식을 이용하여 페이지를 생성한다. 즉, CSI 방식은 낮은 통신 자원을 가진 클라이언트에게 적합한 방법이 될 수 있다. 하지만 이 방법은 문서의 캐시된 부분이 다른 조각에 사용될 수 없으며 임의의 단편화를 지원하지 않는다. 이러한 점을 극복하기위해 클라이언트 브라우저에서 어떤 템플릿 파일에 의해서나 재사용이 가능한 콘텐츠 조각으로 단편화가 가능하게 하는 방법이 제안되었다.

### 4) 다형성 (Polymorphism)

앞서 언급한 것처럼 단편화 레벨의 캐

싱은 저장된 DWC(동적웹콘텐츠) 조각들을 위한 엄격한 재배열을 하는 페이지 레이아웃이나 템플릿 파일이 필요하다. 이러한 엄격한 레이아웃이나 템플릿을 보유해야하는 점을 극복하기위한 방법으로 단편화된 콘텐츠 조각들을 임의로 재배열하는 방법 다형성이라 한다.

### III. CFP Framework

#### 1) CFP framework

동적 웹 콘텐츠처리를 위한 다층 구조의 미들웨어 시스템의 부하분산에 대한 연구들은 주로 세 가지 공통적인 관점에 초점을 맞추고 있다. 즉, 앞장에서 언급한 캐싱(caching), 콘텐츠 단편화(fragmentation), 콘텐츠 다형성(polymorphism)이며 CFP framework는 세 가지 관점을 기반으로 각각의 연구들의 특성 및 지난연구들의 발전과정을 파악한다.

CFP 약어에서 캐싱(caching)은 나머지 특성보다 시기적으로 앞선 연구이며, 플록시 서버, 웹 서버, 응용프로그램 서버, 클라이언트 브라우저와 같이 네트워크에 있는 다층 구성요소에서 콘텐츠를 재사용함으로써 네트워크 및 시스템 자원을 절약하는 방법을 제안한 것이다.

단편화(fragmentation)는 DWP(동적웹 페이지)를 조각내어 세분화하지만, 의미적으로는 별개의 조각으로 나누지 않는 방법으로, 정교한 캐싱을 실현함과 동시에 DWC(동적웹콘텐츠)를 생성할 수 있게 하는 원칙이다.

마지막으로 가장 최근에 이슈가 되고

있는 다형성(polymorphism)은 서버에서 불필요한 중복 콘텐츠 생성을 피하고 한 가지이상의 방법으로 DWP가 조합될 수 있도록 하는 기술이다. 즉, DWC 조각의 배치는 클라이언트의 선호에의해 동적으로 이루어지게 한다. 이러한 방법으로 다형성은 템플릿 파일을 스스로 동적으로 변화함으로써 새로운 차원의 콘텐츠 발전을 가능하게 한다. 또한 다형성은 동적 페이지를 생성한 스크립트가 임의의 조합하의 동일한 콘텐츠를 생산할 수 있다면, 독립적(stand-alone) 특성을 가질 수 있다.

초기 CGI 스크립트는 이미 생성된 콘텐츠 조각들을 재이용하지 않고 내부적으로 전체 페이지를 다시 생성하였기 때문에 캐싱, 단편화, 다형성 부분에서 제로라고 이야기 할 수 있다. 즉, CFP framework의 가장 최하단인 영점에 위치한다.

#### 2) CFP framework 구성

세 가지 주요원칙을 계량화하기위해서는 양적인 평가보다는 적절한 추정을 이용하였다. 그림 2에서 캐싱(caching) 원칙을 보면 DWC가 웹서버보다 사용자에게 근접되어 저장되면 성능이 우수하다고 평가하였다. 즉, 가장 우수한 캐싱정책은 client-side 캐싱이며, 다음으로는 proxy-side 캐싱, 그리고 server-side 캐싱으로 순서가 정해진다.

단편화(fragmentation) 원칙은 DWP(동적웹페이지)를 임의의 DWC(동적웹콘텐츠)로 조각화 할 수 있는 정도에 따라 성능이 구분되어진다. 즉, 임의의 조각

수에 임의의 타입으로 모두 단편화 된다면 이를 Arbitrary라 칭하여 가장 우수하다고 평가하였으며, 제한된 특정타입(XML 데이터, query 등)에 임의의 수로 단편화 된다면 High로, 타입으로만 분류되고 단편화 할 수 없다면 Low라고 명칭 하였다. 즉, 템플릿 파일에서 분리되지 않는 조각들은 다른 클라이언트에서 재사용 될 수 없다고 정의하였다.

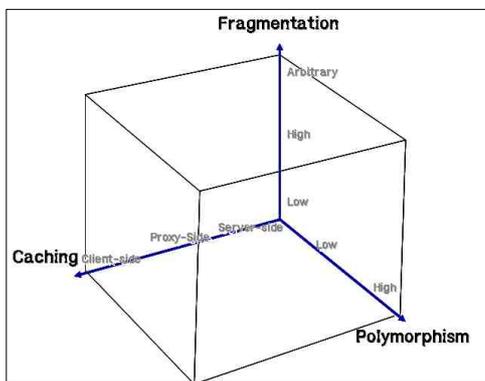


그림 2. CFP Framework  
Fig 2. CFP Framework

다형성(polymorphism) 원칙은 DWP(동적웹페이지)에 있는 DWC(동적웹콘텐츠) 조각들의 임의구성을 지원하는 정도에 따른다. 즉, DWC 조각들의 임의 구성을 위한 유동성 구조를 지원하면 High, 조각들의 임의 구성을 위한 기본적인 지원을 하면 Low라고 정의하였다[6].

#### IV. 시각구조의 Mashup Web 설계

##### 1) Mashup 시각구조의 Web 설계를 위한 CFP framework 적용

본장에서는 앞장에서 언급한 주요 연구 방법들을 CFP framework에 적용하여 세 가지 주요원칙(caching, fragmentation, polymorphism)에 부합하는 정보를 매트릭스에 표시한다. 이러한 방법은 기존의 연구들의 발전과정과 앞으로 접근해야할 연구 대상을 직관적으로 판단할 수 있는 근거를 제시 할 수 있으며, 본 논문에서 제안하는 Mashup 시각구조 Web 설계를 위한 서비스 정책의 방안으로 삼는다.

시대적인 관점에서 기존연구들을 명시하기 위해 배경색이 밝은색에서 어두운색으로 표시하였다. 즉 그림4에서 어두운 배경색이 진할수록 최근에 제안된 연구이다.

그림3을 분석하면 DWC(동적웹콘텐츠)는 빠른 응답시간을 위해 클라이언트에 캐시되는 경향으로 발전해왔으며, 캐시 조각의 재 사용률을 높이기 위해 보다 세밀하게 단편화되어왔다. 또한, 개인에게 맞춤형 서비스를 제공하기 위한 가능한 많은 콘텐츠 조합을 구성할 수 있어야 하는 다형성 문제가 새롭게 등장하고 있다. 사실 이러한 문제는 웹 서비스뿐만 아니라 사용자와 상호작용이 이루어지는 온라인 시스템, 운영체제, 데이터베이스 시스템에도 적용되는 사항이다.

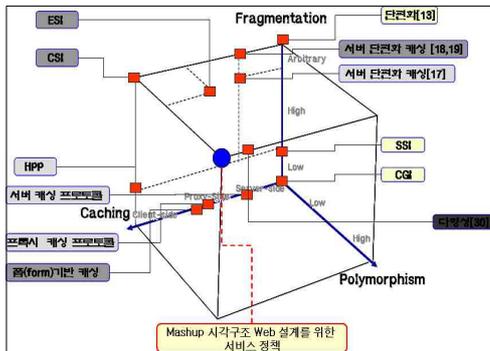


그림 3. 기존연구 흐름도

Fig 3. Original Research Flowchart

단편화 정책은 현대 웹 응용프로그램이 지원하는 콘텐츠의 다양성과 복잡성에 대응하기 위해 주로 점점 세분화되어 왔다. 이러한 다양성은 동적 웹 페이지를 각각의 콘텐츠의 특징별로 세분화하여 캐싱하는 정책을 가져야한다. 또한, 각각의 콘텐츠 조각은 동시에 생성되어야 하는데 서버 자원의 효율적인 운용을 위해서는 콘텐츠 단편화를 추구 해야한다. 마지막으로 다형성은 클라이언트의 선택한 관심분야의 개인화된 동적 페이지 레이아웃을 형성하는데 필요한 특징으로, 앞으로는 개인이 서비스를 사용하는데 있어 일관적인 패턴으로 서비스를 사용하게 하고, 특정 목적정보를 검색하는데 있어서도 기존의 검색 패턴을 기억하여 사용자가 빠르고 편리하게 서비스를 사용할 수 있도록 하는데 필요한 특징이다.

그림3에서 처럼 Mashup 시각구조 Web 설계를 위해서는 사용의 정보인식 편의성을 위하여 미들웨어시스템에 가중되는 부하를 감소시키는 Web 서비스를 설계 한다. 기본 시스템은 Google Map

과 지역정보를 Mashup하였다. 기본 서비스 정책은 다음과 같다.

- (1) 캐싱 (Caching) 정책
  - 사용자의 접속 위치정보를 ip 이용하여
  - 추론하여 초기 설정
  - 지도상에서 이동 패턴을 수집 클라이언트에 저장

- (2) 단편화 (Fragmentation) 정책
  - 지도로딩 모듈
  - 지도 줌인/아웃 모듈
  - 지역단위 이동 모듈
  - 사용자 검색 패턴 모듈
  - 정보단위 노출기능

- (3) 다형성 (Polymorphism) 정책
  - 단편화된 단위 기능을 사용자가 선택 사용할 수 있어야 한다.

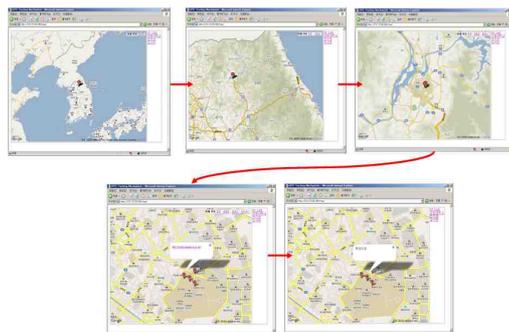


그림 4. Mashup 시각구조 Web 설계

Fig 4. Web Design, Visual Structure Mashup

## V. 결론

본 논문은 정보와 표현 시스템의 발달에 따라 GPIS Tracking 기술과 시각구조의 Mashup Web을 설계하고 구현하였다.

핵심 기술과 기법은 Mashup의 요소로서 지도, 위치정보, 목적정보 예측으로 선점할 수 있게 하였으며, 시각구조는 복합정보를 문자, 기호, 그래픽스 및 영상을 다차원표현으로 하고, GPIS로부터 목적정보를 Tracking하도록 접속위치정보와 최종 목적정보에 도달할 수 있도록 개인에게 특화된 화면정보를 제공하므로 시각구조로 기술(記述)하였다.

## 참고문헌

- [1] Fastcgi white paper from open market, Inc, <http://www.fastcgi.com/devkit/doc/fastcgi-whitepaper/fastcgi.htm>
- [2] Y.-H. Liu *et al.*, "A Distributed Scalable Web Server and Its Program Visualization in Multiple Platforms," *IEEE ICDCS Conf.*, 1996, pp. 665-72.
- [3] E. Cecchet *et al.*, "Performance Comparison of Middleware Architectures for Generating Dynamic Web Content," *Middleware Conf.*, 2003, pp. 282-304.
- [4] S. Papastavrou *et al.*, "An Evaluation of the Java-Based Approaches to Web Database Access," *Int'l. J. Cooperative Info. Sys.*, vol. 10, no. 4, 2001, pp. 401-22.

- [5] J. Challenger *et al.*, "A Publishing System for Efficiently Creating Dynamic Web Content," *IEEE INFOCOM Conf.*, 2000, pp. 844-53.
- [6] A. Datta *et al.*, "Proxy-based Acceleration of Dynamically Generated Content on the World Wide Web: An Approach and Implementation," *SIGMOD Conf.*, 2002.

## 저자약력

안성은(SungEun, Ahn)



2000년 강원대학교  
기계공학과 학사  
2002년 강원대학교  
컴퓨터정보통신공학  
석사  
2009년 강원대학교  
컴퓨터과학과 박사

<관심분야> 시스템프로그래밍,  
Mashup Web, 보안

김정중(Kim, Jung-Joong)



1984년 서울산업대학교  
전자계산학과(학사)  
1986년 고려대학교 전자정보  
전공(석사)  
1994년 고려대학교 전산학과  
과(박사 수료)  
1985~1992 한국방송공사  
(KBS) 전산정보실  
전산분석팀장  
1995~현재 강원대학교  
컴퓨터공학과 교수  
2000~2001 뉴질랜드 오클랜드  
대학교 연구교수

<관심분야> 소프트웨어 공학, 객체지향시스템, 멀티미디어시스템