
XML 문서의 클러스터링 기법을 이용한 스케치맵 시스템

Sketch Map System using Clustering Method of XML Documents

김정숙*, 이아리**, 홍경표**
삼육대학교 컴퓨터학부*, (주)이비전코리아**

Jung-Sook Kim(kimjs@syu.ac.kr)*, Ya-Ri Lee(lyaree@hotmail.com)**,
Kyung-pyo Hong(plhongah@hotmail.com)**

요약

최근 각광을 받고 있는 지도(이하 맵)를 활용한 서비스는 맵에 접근한 후 인터페이스를 통해 다양한 매쉬업 형태의 결과를 제공하는 방식이다. 이러한 서비스는 사용자에게 정확한 정보를 제공할 수는 있지만 맵의 재활용은 어렵다. 본 논문의 스케치맵 시스템은 기존의 대형 맵 시스템과는 달리 목적에 부합하는 특정 지점과 경로를 XML 문서로 표현한다. 또한, 스케치맵 간에 클러스터링 방법을 사용함으로써 맵에서 표현되는 지점을 최적의 내용으로 갱신한다. 그 결과로서, 목적지점에 대한 경로를 간단하게 약도로 표현하기 위해 설계된 맵 서비스 시스템이다. 본 시스템은 스케치 맵의 XML 문서 입력에 대하여 스케치맵 생성기에서 분석·분할·클러스터링의 과정을 통해 유효한 형태의 스케치맵을 생성한다. 스케치맵의 분할 및 병합을 위한 길의처리 방법으로는 LCS(Longest Common Subsequence) 알고리즘을 사용하였다. 또한, 본 스케치맵 시스템에 대한 기대효과를 시뮬레이션으로 제시하여 정보와 지식을 공유하는 보이는 맵들이 모여 거대한 맵을 형성함으로써 새로운 검색 포털로서의 역할을 수행할 수 있음을 보인다.

■ 중심어 : | 스케치맵 | 클러스터링 | 매쉬업 | 분할 |

Abstract

The service that has recently come into the spotlight utilizes the map to first approach the map and then provide various mash-up formed results through the interface. This service can provide precise information to the users but the map is barely reusable. The sketch-map system of this paper, unlike the existing large map system, uses the method of presenting the specific spot and route in XML document and then clustering among sketch-maps. The map service system is designed to show the optimum route to the destination in a simple outline map. It is done by renovating the spot presented by the map into optimum contents. This service system, through the process of analyzing, splitting and clustering of the sketch-map's XML document input, creates a valid form of a sketch-map. It uses the LCS(Longest Common Subsequence) algorithm for splitting and merging sketch-map in the process of query. In addition, the simulation of this system's expected effects is provided. It shows how the maps that share information and knowledge assemble to form a large map and thus presents the system's ability and role as a new research portal.

■ keyword : | Sketch-map | Clustering | Mash-up | Split |

* 본 연구는 2008학년도 삼육대학교 학술연구비에 의해 수행되었습니다.

I. 서론

맵은 인터넷 기반 서비스에서 각광받는 분야 중 하나이다. 맵과 관련한 서비스로는 지도 데이터 및 위성사진 데이터의 제공을 포함하여, 맵에 접근한 후 인터페이스를 통해 다양한 매쉬업 형태의 서비스를 제공하는 2차 애플리케이션 등이 있다. 물론 제공된 소스에 접근할 수 있는 API의 한계 때문에 자신의 애플리케이션을 통하여 맵 데이터를 활용하여 서비스하는 경우도 있다. 맵에서 특정 지점을 표현하려면 GPS나 위치기반시스템 등을 통하여 검색된 지점을 기준으로 그 지점에 대한 맵 데이터를 갱신하여야 한다. 이러한 서비스는 사용자에게 정확한 정보를 제공할 수는 있지만 맵을 재사용하기는 어렵다.

본 논문의 스케치맵 시스템은 기존의 대형 맵 시스템과는 달리 목적에 부합하는 다수의 지점을 표현한 맵의 이동경로를 간단하게 요약(이하 스케치맵)로 표현하는 것으로, 이동성이 있고 유연하면서도 간단한 XML 문서를 사용하는 맵 서비스 시스템이다. 이 시스템은 스케치 맵의 XML 데이터 입력에 대하여 스케치맵 생성기에서 분석·분할·클러스터링의 과정을 통해 유효한 형태의 스케치맵 XML을 생성한다. 생성된 XML 문서는 KML(Keyhole Markup Language)[15] 파일과는 달리 특정 지점과 경로를 표현하는 XML 문서로서, 스케치맵 간에 클러스터링 방법을 사용하여 맵에서 표현되는 지점을 최적의 내용으로 갱신하여, 목적 지점에 대한 경로를 탐색하는 기법을 적용한 결과이다.

스케치맵 시스템에서는 약도나 경로를 표현하는 방법으로 바이너리 맵 데이터와 경로 정보를 표현한 스크립트 구조를 적용하였다. 또한, 맵 데이터의 수정을 위해 별도의 애플리케이션을 사용하여 하나의 입력 지점에 대하여 방대한 데이터를 지속적으로 업데이트할 수 있도록 스케치맵 간의 관계를 휴리스틱하게 분석함으로써 문서의 유효성을 갖도록 처리하였다. 그 결과, 다수의 사용자가 직접 참여하여 원하는 맵 문서를 생성하고 검색할 수 있는 시스템이 되도록 하였다. 그리고, 본 연구에서는 일관된 스케치맵 문서를 유지하기 위하여 가중치에 기반한 클러스터링 기법을 적용하였으며, 스

케치맵 XML 문서의 분할 및 병합을 위한 질의처리 방법으로 LCS(Longest Common Subsequence) 알고리즘[6]을 사용하였다. 마지막으로, 본 연구 시스템에 대한 기대효과의 적용사례를 시물레이션으로 제시하여, 실제 정보와 지식을 공유하는 보이는 맵들이 모여 거대한 맵을 형성하여 새로운 검색포털로서의 역할을 수행할 수 있음을 보인다.

본 논문의 구성은 2장에서 관련연구로 클러스터링 기법과 맵을 표현하는 XML 기법에 대하여 설명한다. 3장에서는 스케치맵 시스템의 입력자료, 스케치맵 생성 과정 및 실험 결과에 대하여 설명한다. 4장에서는 제안한 스케치맵의 다양한 활용에 대한 시물레이션을 제시하고, 마지막으로 5장에서는 결론 및 향후 연구에 대하여 언급한다.

II. 관련 연구

본 논문은 맵 구조를 가진 간소화된 스케치맵의 제안이다. 따라서 제안하는 스케치맵은 XML 문서로서, 스케치맵 문서들 간의 클러스터링 방법을 기반으로 능동적이고 일관된 스케치맵 문서의 저장 기법과 검색 알고리즘을 제안하였다.

본 장에서는 관련 연구로 XML 데이터의 클러스터링 기법, 지도를 표현하는 XML 기법에 대하여 소개한다.

1. XML 데이터의 클러스터링 기법

XML의 사용이 늘어남에 따라 XML 데이터를 효율적으로 저장하고 관리하기 위한 클러스터링 연구가 활발히 진행 중이다. 이러한 클러스터링에 관한 연구는 크게 문서의 구조적인 특성에 기반한 방식과 내용에 기반한 방식으로 구분할 수 있다. 또한 최근에는 경로에 기반한 클러스터링 기법(SL, SP)을 통해 직관적으로 클러스터링하는 기법에 대한 연구도 관심을 끌고 있다.

문서의 구조 기반 연구는 과거 객체지향 데이터베이스 분야에서 연구된 기법들이 대표적이다. [2]에서는 문서의 구조를 토대로 깊이우선(depth-first), 넓이우선(breadth-first) 방식을 제안하였다. 이들 중 깊이우선

방식은 XML 원본에 나타나는 순서대로 구성요소들을 저장하는(document order)방식과 유사하며, 클러스터링에 대한 특별한 고려를 하지 않는 많은 XML 저장소에서 이 방식을 사용하여 XML 문서를 저장한다. 하지만 정형화된 구조를 가지지 않는 XML 문서의 특성을 고려할 때, 일정한 구조에 기반하여 클러스터링을 수행하는 이러한 방식은 한계를 갖는다.

그리고, 문서의 내용 기반 연구에서 XML의 경우, 경로 질의를 보다 효율적으로 지원할 수 있도록 문서의 내용에 기반한 클러스터링 방식에 대한 연구가 주로 수행되었다. XML의 경로는 태그(tag)라는 구성요소의 레이블들로 구성된다. 따라서 내용에 기반한 XML의 클러스터링 기법에서는 기본적으로 이러한 태그를 이용하여 클러스터링을 수행한다. [5]에서는 XML 문서를 저장하는 방식들을 클러스터링 관점에서 비교, 분석하고, XML 문서를 저장할 때 적용할 수 있는 세 가지 클러스터링 방식에 대하여 기술하였다. 첫 번째 방식은 DTD 분석 등을 토대로 실제로 연관성을 가지는 구성요소들을 클러스터링하는 방식으로 [3]에서 사용하는 방식이 여기에 해당한다. 두 번째 방식은 동일한 성격을 갖는 구성요소들을 함께 클러스터링하는 방식으로 [3][4]에서 사용하는 방식이다. 여기서 동일한 성격이란 동일한 태그명을 가지는 구성요소들 혹은, 형제 관계에 있는 구성요소들의 성격을 말한다. 이 경우 저장소로부터 XML 문서 원본을 다시 구성하는 경우를 제외한 대부분의 상황에서 동일한 태그명에 따라 클러스터링하는 경우가 보다 우수한 성능을 보여 주었다. 세 번째 방식은 XML 문서 원본에 나타나는 순서대로 구성요소들을 저장하는 방식으로, 앞서 언급한 구조에 기반한 클러스터링 방식이다. [5]에서는 DTD에 기반한 두 가지 클러스터링 기법을 제안하였다. 첫 번째는 DTD에서 동일한 노드 타입을 가지는 구성요소들을 함께 저장하는 기법으로, 앞에서 언급한 [2]의 두 번째 방식과 유사하다. 두 번째는 스카마 그래프를 토대로 의미에 따라 서브 트리를 분류하고, 이에 기반하여 구성요소들을 분류, 저장하는 기법이다. 하지만 저장의 최소 단위인 의미 블록의 크기에 대한 고려가 없고, 이를 사용한 클러스터링 기법이 상대적으로 큰 우위를 가지지는 못하는

단점을 지닌다.

기존의 내용에 기반한 클러스터링 기법들이 단순히 태그를 이용한 굵은 단위의 클러스터링이었다면, [8]에서는 경로 패턴에 기반한 클러스터 색인(clustered index)인 PathGuide를 제안하였다. 제안한 방법은 동일한 경로 패턴을 가지는 구성요소들을 함께 저장하여 단순 질의뿐만 아니라 조상-자손 관계를 포함한 복잡한 형태의 경로 질의까지 효과적으로 처리할 수 있다. 하지만 PathGuide는 서픽스(suffix) 단위로 클러스터링이 발생하기 때문에, 제한적인 경우에만 효과를 얻을 수 있다. 즉, 연속되는 부모-자식 관계의 긴 서픽스 형태를 가지는 보다 다양한 형태의 질의 처리에는 이득을 얻지 못한다.

경로 기반의 직관적인 클러스터링은 SL(Same Label) 클러스터링과 SP(Same Path) 클러스터링의 두 가지 기법이 사용되고 있다. SL 클러스터링은 동일한 태그명을 가지는 모든 노드들을 동일한 클러스터에 저장하는 기법이다. 하나의 클러스터에 저장되는 데이터 노드들은 문서 순서대로 저장되며, 각 클러스터는 저장된 데이터 노드의 레이블을 식별자로 가진다. 각 레이블별로 하나씩 클러스터가 생성되므로, 클러스터링을 수행한 후 최종적으로 생성된 클러스터의 개수는 문서에 포함된 서로 다른 레이블의 개수와 같다. 구현은 구조의 결합 등에서 사용하는 레이블의 역색인 리스트를 이용하면 간단하게 이루어진다.

SP(Same Path) 클러스터링은 루트로부터의 절대경로가 동일한 모든 노드들을 동일한 클러스터에 저장하는 기법이다. 하나의 클러스터에 저장되는 데이터 노드들은 문서 순서대로 저장되며, 각 클러스터는 저장된 데이터 노드의 절대경로를 식별자로 가진다. 이 기법은 절대 경로가 같은 노드들을 하나의 목표 집합(target set)으로 관리하는 DataGuide와 같은 경로 색인을 이용하면 쉽게 구현이 가능하다. 각 절대 경로마다 하나씩 클러스터가 생성되므로, 클러스터링을 수행한 후 최종적으로 생성된 클러스터의 개수는 문서에 대한 DataGuide의 노드 수, 즉 목표 집합의 총 개수와 같다.

본 연구에서 하나의 클러스터에 저장되는 데이터 노드들은 순서 없이 저장되며, 각 클러스터는 저장된 데

이터 집합을 대표하는 고유 이름을 가진다. 이것은 SP 클러스터링과 다르게 순서가 없으며, 클러스터링된 노드 집합들 사이에서도 분할 가능한 특징이 있다. 클러스터링을 수행한 후 최종적으로 생성된 클러스터의 개수는 최대 문서의 개수만큼 만들어진다. 또한 루트로부터 최하위 노드까지의 최대 깊이는 문서의 개수만큼 깊다. 스케치맵의 구조를 결정하는 방법은 적어도 최대 깊이, 최대 문서에서 최적의 문서를 찾아내는 것이다.

SL과 SP 클러스터링의 차이는 클러스터링 단위에 있다. SP 클러스터링은 동일한 레이블을 가지는 노드들 중에서도 절대 경로에 따라서 서로 다른 클러스터에 저장되기 때문에 SL 클러스터링에 비해 보다 미세 단위의 클러스터링을 수행한다. 이러한 단위의 차이는 질의 처리에도 영향을 준다. SL과 SP 클러스터링은 특정 형태의 질의에서만 효과를 보이는 한계를 가진다. 따라서 이러한 문제를 해결하기 위해서는 질의 형태에 국한하지 않는 융통성있는 클러스터링 기법이 필요하다.

이러한 문제점에 대한 해결책으로서 경로 유사성(Path Similarities) 기반 클러스터링 방법이 제안되었다[10]. 하나의 SL 클러스터를 이루는, 즉 경로가 동일한 레이블로 끝나는 서로 다른 SP 클러스터들 간의 경로 유사도는 SP 클러스터를 정점(vertex)으로 가지고 정점들 간의 경로 유사도를 간선의 가중치(weight)로 가지는 가중 그래프(weighted graph) $G=(V, E)$ 로 표현할 수 있다. 하나의 XML 문서에 대해 경로 유사도 그래프는 문서에 포함된 개별의 레이블마다 각각 하나씩 존재한다. 이 때 경로 유사성 클러스터링은 이러한 경로 유사도 그래프를 토대로 SP 클러스터들을 분류하는 그래프 분할 작업(graph partitioning problem)이 되며, 모든 레이블의 경로 유사도 그래프에 대해 분할 작업을 수행함으로써 클러스터링 과정이 완료된다. 일반적으로 그래프 분할 알고리즘은 NP 완전 문제(NP complete problem)로 적절한 휴리스틱을 사용하여 해결한다. 경로 유사도 그래프의 분할을 위해 [1]과 유사한 탐욕 알고리즘(greedy algorithm)을 수행한다.

2. 맵을 표현하는 XML 기법

맵을 표현하는 XML 기법으로는 2차원 벡터를 이미

지화하는 VML & SVG 방법, VML를 이용한 동적 지역분할 방법, MAP 용량을 줄이는 방법 등이 있다.

맵을 이미지화하는 VML & SVG 방법에서, VML(Vector Markup Language)은 XML의 서브셋으로, 벡터 정보를 텍스트와 HTML 데이터로 통합할 수 있다. 또한 SVG(Scalable Vector Graphics)와 비교된다. SVG는 2차원 벡터 그래픽의 표현을 위한 XML 서브셋[16]이다. SVG는 웹에 적합하며 출력과 다양한 화면으로 조절한다. 웹 애플리케이션을 위해서는 2차원 벡터를 만들고 그래픽을 생성할 때, 자바스크립트 라이브러리와 함께 SVG와 VML을 이용한다. 즉, 웹상에서 생성한 그래픽 객체는 DOM 객체로서, 자바스크립트를 이용하여 이벤트 핸들러에 접근한 후 수정할 수 있다. 자바스크립트 라이브러리를 이용하면 쉽게 크로스 브라우저에 대하여 호환되는 그래픽을 제공하는 장점이 있으며, 웹상에서 벡터 그래픽을 작성할 수 있다. 2차원 벡터를 표현하는 이들 XML의 서브셋은 맵 데이터를 만드는데 사용되기도 하지만, 확장된 마인드 맵 또한 SVG, VML로도 생성할 수 있다[17]. 확장된 맵의 개념은 2차원 벡터정보와 함께 그 이상도 나타낼 수 있다. 물론 맵 데이터를 생성할 때 XML 문서를 사용하는 방법보다 일반적인 바이너리 형태의 이미지로 맵 데이터를 사용하고 있다. Flash는 2차원 벡터를 처리한다는 관점에서는 매우 유사하지만 XML 기반의 문서와 폐쇄된 바이너리 형식이라는 점에서 차이점이 있다. 즉, VML과 SVG의 경우에는 XML 문서형식으로 언제든지 수정의 접근성이 뛰어나지만, Flash는 데이터를 수정하기 위한 별도의 애플리케이션이 필요하다.

VML을 이용한 동적 지역분할 방법은 2차원 벡터 생성 기반 기술이 부족한 것을 개선하기 위한 기술로 개발되었다. 공통의 맵 데이터를 다양한 사용자들이 함께 이용하는 경우, 사용자에 따라 사용 맵 영역을 분할하여 처리하여야 한다. [9]에서는 이를 위한 맵 데이터의 분할과 병합을 능동적으로 대처할 수 있는 방법을 제시하고 있다.

MAP 용량을 줄이는 방법은 일반적인 맵 데이터가 다양한 정보를 포함하기 위해서는 큰 용량의 저장 환경을 요구하게 된다는 것에 대한 해결책이다. 즉, 애플리

케이션이 효율적으로 맵 데이터를 사용하기 위해서는 좀 더 경량화하기 위한 방법들에 대한 연구[14]가 수행되고 있다.

III. 스케치맵 시스템

본 논문의 스케치맵 시스템은 스케치맵에 대한 XML 데이터를 입력받아, 스케치맵 생성기에서 분석·분할·병합의 처리과정을 통하여 유효한 스케치맵 XML의 결과를 출력하며, 시스템 구성도는 [그림 1]과 같다.

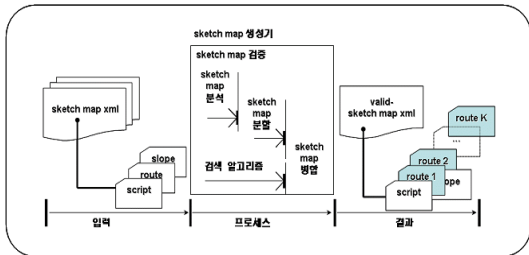


그림 1. 스케치맵 시스템 구성도

스케치맵 생성기의 처리과정은 다음과 같다. 먼저, 입력으로 사용된 스케치맵 경로에 대한 구성요소를 분석하여 중복된 구성요소가 있는지 확인하는 분석 과정을 거친다. 그리고 분석 과정에서 스케치맵 XML의 모든 구성요소를 탐색하면서 적절한 분할 위치를 선정한다. 마지막으로, 분할위치로 선정된 구성요소를 기준으로 새로운 스케치맵 작성을 위한 분할과 병합과정을 반복하면서 유효한 경로를 가진 스케치맵 XML이 생성된다.

1. 입력 자료

스케치맵의 입력 자료는 [그림 2]와 같은 잘 구성된 XML로 제안한다. 그 이유는 RDBMS(Rational Database Management System)에게 좋은 선택을 제공할 수 있는 등 유연함을 가지고 있기 때문이다. 이러한 XML 구조는 애플리케이션 관점에서도 이질이나 이기 종간 자료의 교환 시 의미 전달이 동일하게 이루어진다. 또한 스케치맵 XML은 클러스터링하거나 재구성하

기 쉽고, 복잡한 쿼리에 대한 효율적인 대처가 가능하고 다양한 애플리케이션으로 자료 변형이 가능하다. 하지만 다양한 경로를 표현하는 맵 데이터의 요청에 대하여 최적의 경로임을 보증할 수 있도록 하기 위해서는 최적화된 XML 클러스터링 방법이 요구된다.

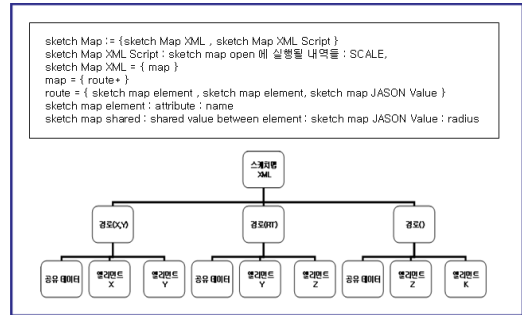


그림 2. 스케치맵 XML 구조

유효한 스케치맵에 대한 XML에는 [그림 3]과 같이, 맵 집합 A와 맵 집합 B의 관계에 대한 경로의 연관성을 radian m으로 표시하였다. 스케치맵 m(A, B)의 경로 A와 B를 표시하기 위한 방법은 id만 부여하고 순서는 갖지 않도록 표현하였다. 즉, A = Br와 같이 B를 기준으로 했을 때, m만큼 회전한 것이다. 이를 그림 4의 유효한 스케치맵 AD에서 표현해 본다면, “A=Brm”와 “B=Drn”의 값을 그대로 유지한다.

<pre><xml id="A, C"> <route slope="y, r1"> <vertex_name="A" object_name="x"> <vertex_name="B" object_name="y"> </route> <route slope="y, r2"> <vertex_name="B" object_name="x"> <vertex_name="C" object_name="y"> </route> </xml></pre>	<pre><xml id="A, D"> <route slope="y, r3"> <vertex_name="A" object_name="x"> <vertex_name="B" object_name="y"> </route> <route slope="y, r4"> <vertex_name="B" object_name="x"> <vertex_name="D" object_name="y"> </route> </xml></pre>
---	---

그림 3. 유효한 스케치맵의 XML

스케치맵 XML AC는 스케치맵 XML AB와 스케치맵 XML BC를 분석하여 생성된 유효한 맵이다. 스케치맵 XML AC가 존재하기 이전의 맵은 AB 혹은 BC였다. 맵 A의 집합에서 B의 존재를 확인하게 되면 맵 A 집합의 AB는 AB:BC로 분할된다. 이 분할된 맵의 경로를 병합하면 AC를 가진 유효한 스케치맵 XML이 된다.

[그림 4]는 유효한 XML의 실제 경로를 추출해 낸 것이다. 유효한 경로는 점선으로 표시되며, 가중치를 부여하여 느슨하게 연결된 경로이다. 유효한 경로를 통하여 생성된 또 다른 유효한 경로는 마지막에 참조된 느슨한 경로의 합계로 표시되도록 하였다.

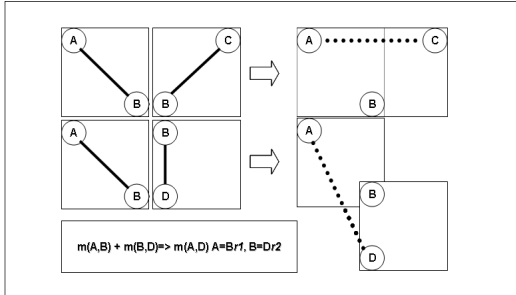


그림 4. 유효한 스케치맵의 XML의 경로도

2. 스케치맵 생성

2.1 클러스터링 기법

하나의 스케치맵에 대한 XML 문서에는 두 개의 지점(구성요소)이 있으며, 또 하나의 새로운 스케치맵 문서가 입력될 경우, 이들을 클러스터링해야 한다. 이때의 클러스터링은 스케치맵의 단순화, 경로 분할, 스케치맵 클러스터링 등의 3단계를 통해 수행한다. 첫 번째 단계인 단순화는 XML에서 노드를 구성하는 맵을 지칭한다. 두 번째 단계인 경로의 분할은 새로운 맵이 입력되었을 때, 기존 XML 노드를 검색하여 맵 집합에 존재하지 않으면 새로운 노드로 추가하는 과정이다. 마지막 단계인 스케치맵 클러스터링에서는 맵 집합 혹은 맵이 기존의 맵에 병합되는 단계이며, 먼저 단순화시킨 후 경로를 분할하여 맵 집합을 클러스터링한다. 만약, 기존의 스케치맵 집합에 경로를 추가하는 경우, 그 경로가 존재하면 기존 경로의 가중치를 고려하여 스케치맵의 경로를 분할하고 기존 맵 경로는 삭제한다. 또한, 경로가 존재하지 않으면 스케치맵 집합에 경로를 추가하는 방법으로 클러스터링 한다.

본 연구에서 수행하는 클러스터링 기법을 설명하기 위해 몇 가지 기본 용어의 개념들을 다음과 같이 정의한다.

[정의 1]

$M(Kn) = \{ \text{경로(Route)에서 } Kn \text{ 지점을 가지는 모든 sketch Map} \}$
 $m : \text{sketch Map}$
 $\text{route} : \text{두 지점을 연결하는 경로, 속성}$
 $m(x, y) : \text{두 지점 } x, y \text{의 강한 연결 경로로 표현한 맵}$
 $mn(x, y) : \text{클러스터링된 } x, y \text{에 가중치 } n \text{의 값이 적용된 맵}$

또한, 경로 AB와 경로 BC는 맵 집합 B에 속하는 경로들이고, [정의 2]와 같은 관계로 표현하기로 한다.

[정의 2]

$m(A, B) \cap m(B, C) \in M(B) \Rightarrow \{ \exists v \mid M(A) \cap M(C) \text{ 's } v \}$
 $m(A, B) + m(B, C) > m_1(A, C)$
 $m_1(A, C) + m(C, D) > m_2(A, D) : m(A, B) + m(B, C) + m(C, D)$

임의의 x 지점을 포함하는 [그림 5]의 스케치맵 클러스터링 과정을 [정의 1], [정의 2]를 통하여 표현하면 다음과 같다.

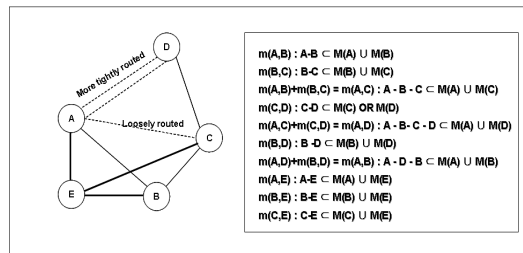


그림 5. M(x) : x 지점을 경로로 포함한 스케치맵

위의 과정을 통하여, 본 연구에서 제안한 클러스터링 알고리즘은 [그림 6]과 같다.

클러스터링 알고리즘

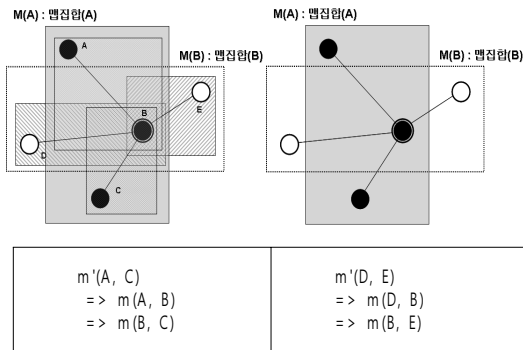
```
while(true) {
    if (search(M(k), m(Vi, Vj))
        //k구성요소를 가진 맵집합에서 m(Vi, Vj) 탐색
    then
        split(m(k, Vi), m(k, Vj)) //두 개의 경로 Vi, Vj로 맵을
        분할
    else add(M(k), m(k, Vi)) //새로운 경로의 추가
}
```

그림 6. 클러스터링 알고리즘

2.2 질의처리 방법

본 스케치맵 시스템에서는 두 지점을 연결하는 경로를 우선적으로 선택하며, 차례로 경로를 통하여 만들어진 새로운 경로(loosely routed)를 맵으로 사용하게 된다. 이렇게 만들어진 새로운 경로는 다시 새로운 맵의 하위 경로로 사용된다. 즉, 두 문자열을 입력받아 공통적인 가장 긴 문자열을 찾는 LCS(longest common subsequence) 알고리즘[6]을 적용하여, 하위 경로를 포함하고 있는 맵 소속임을 찾기 위하여 문제를 더 분할하게 된다. 결국, 실제 맵이 존재하는지, 이 맵의 집합에 경로가 있는지를 판단하기 위한 알고리즘으로 LCS 알고리즘을 사용하였다.

입력된 맵이 병합되는 맵 집합을 선택하는 기준은 [그림 7]로 표현할 수 있다.



위의 과정을 통하여, 새로운 맵 $m'(AD)$ 가 다음과 같이 생성된다.

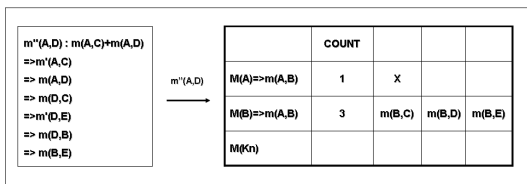


그림 7. 새 맵이 위치할 맵 집합의 결정방법

맵은 그림 8의 알고리즘을 통해, 가중치(COUNT)를 탐색하여 맵 집합에서 가장 많이 나온 곳, 즉 가중치가 가장 큰 값을 선택하여 문서의 노드로 사용하도록 클러

스터링한다.

```


가중치 탐색 알고리즘


maxCount(m(rn, ln)) {
  while(each(M(Kn))) {
    //m(rn, rn-1) . m(ln, rn-1) 의 선택은 휴리스틱함
    if(exist(m(rn, rn-1), M(Kn)))
      then
        maxCount(m(rn, rn-1));
      else return M(Kn);
    if(exist(m(ln, rn-1), M(Kn)))
      then
        maxCount(m(ln, rn-1));
      else return M(Kn);
  }
}
    
```

그림 8. 가중치탐색 알고리즘

[그림 7]에서, LCS 알고리즘을 이용하여 가장 많이 나타나고 있는 맵의 집합에 새로운 경로가 있도록 정의하는 과정을 설명한다. 스케치맵에서는 트리의 가장 하단 노드 경로를 가진 최상단의 맵을 선택한다. 탐색된 맵 집합 $M(X)$ 는 맵과 경로를 가지며, 그 경로의 가중치가 가장 큰 값을 탐색하여 새로운 맵으로 추가함으로써 도달 가능한 경로 맵이 생성된다.

2.3 도달 경로 탐색 과정의 예

[그림 9]와 [그림 10]에서는 도달할 수 있는 경로를 찾아주는 과정을 설명한다. 맵 $m(A, C)$ 는 경로 AC로 유도된 $m(A, B)$ 와 $m(B, C)$ 를 가지고 있고, $m(D, E)$ 는 경로 DE로 유도된 $m(D, B)$ 와 $m(B, E)$ 를 가지는 상태에서 새로운 $m(A, E)$ 를 클러스터링한다.

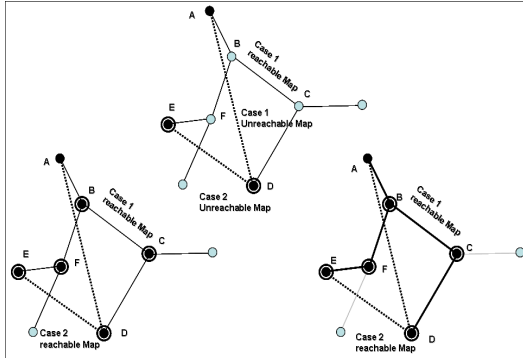


그림 9. 도달 경로 탐색과정

도달 경로 탐색 과정

$m(A, E)$ 의 맵 경로를 찾는 방법 : $m(A, E) : m(E, F) m(F, B) m(B, A)$

=> $m(E, D) m(A, B) m(B, C) m(B, D) | m(A, D) m(E, F) m(F, B) m(B, C) m(B, D) | m(E, D) m(A, D)$

$m(E, D) m(A, D)$ 의 경로 탐색과정

=> $m(E, F) m(F, B) m(B, C) m(C, D) m(A, B) m(B, C) m(C, D)$

- * 맵 경로 집합에서 중복된 경로는 SKIP(Delete)
- (=> $m(E, F) m(F, B) m(B, C) m(C, D) m(A, B) m(B, C) m(C, D)$)
- * 집합을 가중치를 이용하여 SKIP(DELETE)

=> $m(E, F) m(F, B) m(B, C) m(A, B) m(B, C)$

=> $m(E, F) (M(B) | M(C))$

- *** $count(M(B)) > count(M(C)) \rightarrow M(C)$ 소거
- $M(B) (m(B, F), m(B, C), m(B, A), m(B, C)) | M(C) (m(C, D))$
- : 두 집합중 가중치가 가장 높은 B맵 집합을 선택, M(C) 집합 제거

=> $m(E, F) m(B, F) m(B, C) m(B, A) m(B, C)$

- ** $m(B, C)$ 소거
- : $M(A) (m(B, A))$

=> $m(E, F) m(F, B) m(A, B)$

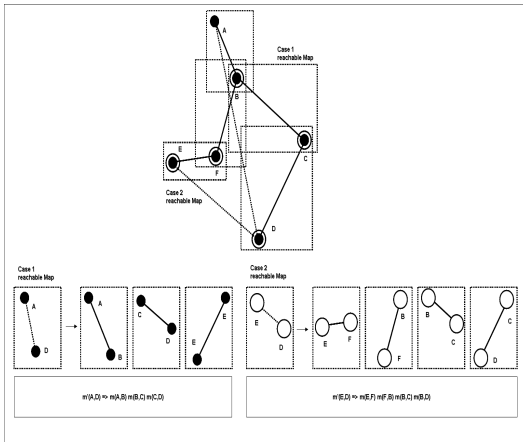


그림 10. 단순화된 맵의 경로 (1단계)

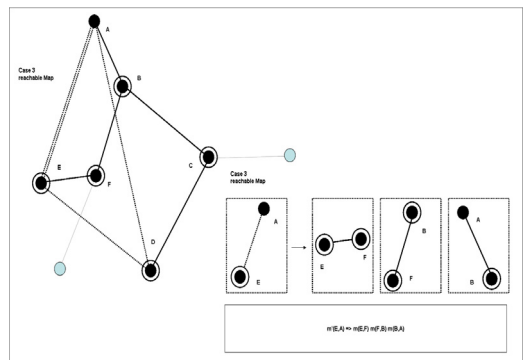
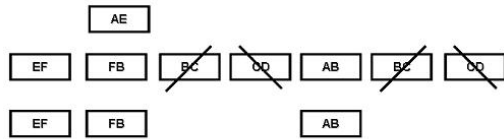


그림 11. 단순화된 맵의 경로 (2단계)

위 과정에 대한 도달 경로 탐색 과정은 다음과 같다.

[그림 12]는 $m(A, E)$ 의 경로를 클러스터링된 스케치 맵 $m(A, E)$ 의 XML로 표현한 것이다. 경로는 있으나 도달할 수 없는 경우에도, 스케치맵에서는 도달할 수 있는 방법으로 검색하여 스케치맵 XML로 표현해 준다.

Case 3 reachable Map $m(AE)$



```

m(AE) = m(EF) m(FB) m(BC) m(CD) m(AB) m(BC) m(CD)
<SML ID="AE">
<route slope="Y,a"><V N="E" OBJN="X"><V N="F" OBJN="Y"></route>
<route slope="Y,b"><V N="F" OBJN="X"><V N="B" OBJN="Y"></route>
<route slope="Y,c"><V N="B" OBJN="X"><V N="C" OBJN="Y"></route>
<route slope="Y,d"><V N="C" OBJN="X"><V N="D" OBJN="Y"></route>
<route slope="Y,e"><V N="A" OBJN="X"><V N="B" OBJN="Y"></route>
<route slope="Y,f"><V N="B" OBJN="X"><V N="C" OBJN="Y"></route>
<route slope="Y,g"><V N="C" OBJN="X"><V N="D" OBJN="Y"></route>
</SML>

m(AE) = m(EF) m(FB) m(AB)
<SML ID="AE">
<route slope="Y,a"><V N="E" OBJN="X"><V N="F" OBJN="Y"></route>
<route slope="Y,b"><V N="F" OBJN="X"><V N="B" OBJN="Y"></route>
<route slope="Y,c"><V N="A" OBJN="X"><V N="B" OBJN="Y"></route>
</SML>
    
```

그림 12. 스케치맵 $m(A, E)$ XML

IV. 시스템 활용 시뮬레이션

본 논문의 스케치맵 시스템은 기존의 맵 데이터 입력이 탐색 경로가 포함된 바이너리 형태를 사용하는 것과 다르게 XML 데이터를 사용하여, 이전 스케치맵들

과 클러스터링된 새로운 스케치맵 XML을 출력한다. 또한 스케치맵 XML 문서 구조는 사용자의 문서를 포함하여 편집도 가능하며, 두 지점간의 관계를 표시하는 값이 유일한 입력 데이터이다. 연속된 지점 사이의 관계는 문서를 클러스터링하는 과정을 통해 하나의 맵 문서로 통합되어 새로운 문서가 생성된다. 따라서 스케치맵 데이터는 한 개 이상의 경로를 가진 맵으로 구성되어 각 단위별로 하나의 맵 단위로 사용할 수 있는 XML 문서이다.

1. 목적지 약도

[그림 13]의 (a)는 A(삼육대)에서 H(광화문)까지의 이동 경로를 나타내는 스케치맵이다. 삼육대 ~ 광화문 간의 중간 경로는 (b)와 같이 세부 맵 집합으로 나타나며, 재귀적 분석을 통하여 검색([그림 14])되어 실제 스케치맵 m'(삼육대, 광화문)의 문서를 만들어낸다.

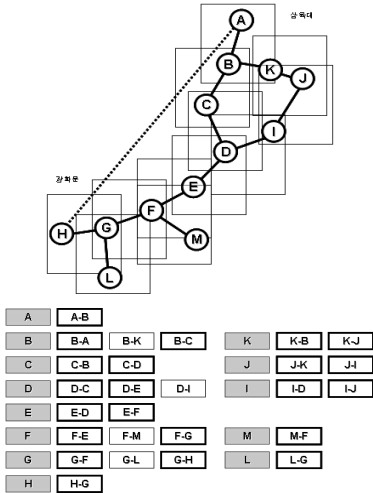


그림 13. (a)삼육대~광화문의 스케치맵, (b)A(삼육대)~H(광화문)의 중간경로 집합

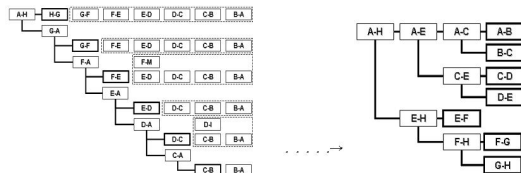


그림 14. 스케치맵 m(삼육대, 광화문)의 검색 과정을 통해 클러스터링된 문서

2. 인맥정보 표현

스케치맵 시스템은 유형의 약도를 표현할 수 있는 것에 더하여, 실제로 존재하지 않는 무형의 약도를 그리는 곳에도 활용이 가능하다. [그림 15]는 개인의 인맥에 대한 정보이며, 본 시스템을 이용하여 인맥정보를 표현할 경우, 최종적으로 생성된 스케치맵은 가중치가 큰 맵으로 표현된 스케치맵을 가진 문서를 얻게 될 것이다.

이아리	이-이진우	이-홍경표	이-김정숙	이-이상주	이-ABC
홍경표	홍-이아리	홍-김정숙	홍-리민준	홍-장기익	홍-이상주
김정숙	김-이아리	김-홍경표	김-민세오	김-서은원	김-ABCD
이상주	이-이아리	이-홍경표	이-이수희	이-이진우	이-김경원
김경원	김-이상주	김-김성연	김-백정수	김-서미란	김-최가박

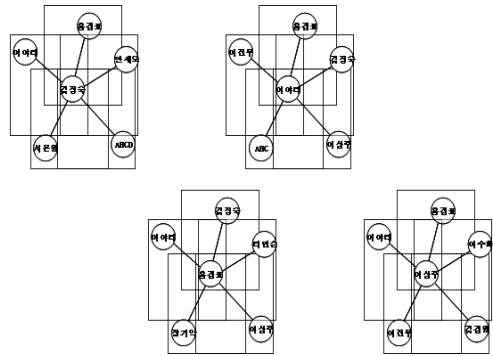


그림 15. 개인의 인맥과 스케치맵

각 인맥들을 스케치맵으로 표현한 후, [그림 16]과 같이 m'(김정숙, 최가박)의 맵을 검색하여 맵을 생성하고 가장 긴 경로를 사용하여 인맥의 스케치맵으로 사용할 수 있다.





그림 16. 스케치맵 m'(김정숙, 최가박) 검색 과정

V. 결론 및 향후 연구

본 논문의 스케치맵 시스템은 사용자의 간단한 목적지 정보에 대한 XML 데이터를 입력으로 하여 클러스터링을 수행한 결과로 표현하는 맵 서비스를 위한 제안이다. 따라서 본 연구에서는 스케치맵을 표현하기 위해 두 지점 사이에 도달할 수 있는 최다 경로를 가져야 한다는 전제하에 맵을 필요로 하는 사용자 정보를 축적하고 그 선호도를 가중치로 적용한 결과를 도출하였다.

본 스케치맵 시스템에서는 도달할 수 있는 경로에 대한 길의 처리방법을 통하여 스케치맵 XML로 표현해 준다. 따라서 생성된 스케치맵은 사용자가 제공한 경로 정보들의 집합을 이용하여 도달할 수 있는 최적의 길을 결과로 찾아낸다. 또한, 기존의 경로 탐색 서비스에서는 경로에 대한 고정 패턴의 수정이 쉽지 않았지만, 본 스케치맵 시스템에서는 필요로 하는 맵의 생성에 대하여 사용자의 직접 참여가 쉬워져 다양한 경로를 생성할 수 있다. 그러나 다수의 사용자들이 임의로 제공한 입력정보에 대한 객관적인 신뢰도 보장은 전제될 수 없다는 문제점을 갖고 있다. 하지만 웹 백과사전으로 일컬어지고 있는 ‘위키피디아’가 정보와 지식을 공유하는 미국 온라인 백과사전으로 현대 상식의 집합으로서 발전되어 가고 있다는 것과 비교한다면 다소 위안이 될 수 있다. 이러한 사실을 토대로, 본 스케치맵 시스템은 실제 보이는 맵을 표현하며 그 이상의 경우까지도 표현할 수 있다. 간단하게 표현된 이러한 맵들이 모여서 거대한 맵들이 형성될 수 있기 때문에 하나의 새로운 검색포털로서의 역할이 가능해 질 수 있을 것으로 기대된다.

향후 연구에서는 제안한 스케치맵 시스템에 대한 실

제적인 구현을 통해, 사용자들의 선호도 평가를 통해 유사 시스템과의 차별화를 이룰 수 있을 것이다. 또한 보다 다양한 유형의 맵 데이터들을 사용할 수 있도록 스케치맵을 구성하는 등의 시스템 확장에 대한 연구가 필요하다.

최다 경로를 가져야 한다는 전제하에 맵을 필요로 하는 사용자 정보를 축적하고 그 선호도를 가중치로 적용한 결과를 도출하였다.

본 스케치맵 시스템에서는 도달할 수 있는 경로에 대한 길의 처리방법을 통하여 스케치맵 XML로 표현해 준다. 따라서 생성된 스케치맵은 사용자가 제공한 경로 정보들의 집합을 이용하여 도달할 수 있는 최적의 길을 결과로 찾아낸다. 또한, 기존의 경로 탐색 서비스에서는 경로에 대한 고정 패턴의 수정이 쉽지 않았지만, 본 스케치맵 시스템에서는 필요로 하는 맵의 생성에 대하여 사용자의 직접 참여가 쉬워져 다양한 경로를 생성할 수 있다. 그러나 다수의 사용자들이 임의로 제공한 입력정보에 대한 객관적인 신뢰도 보장은 전제될 수 없다는 문제점을 갖고 있다. 하지만 웹 백과사전으로 일컬어지고 있는 ‘위키피디아’가 정보와 지식을 공유하는 미국 온라인 백과사전으로 현대 상식의 집합으로서 발전되어 가고 있다는 것과 비교한다면 다소 위안이 될 수 있다. 이러한 사실을 토대로, 본 스케치맵 시스템은 실제 보이는 맵을 표현하며 그 이상의 경우까지도 표현할 수 있다. 간단하게 표현된 이러한 맵들이 모여서 거대한 맵들이 형성될 수 있기 때문에 하나의 새로운 검색포털로서의 역할이 가능해 질 수 있을 것으로 기대된다.

향후 연구에서는 제안한 스케치맵 시스템에 대한 실제적인 구현을 통해, 사용자들의 선호도 평가를 통해 유사 시스템과의 차별화를 이룰 수 있을 것이다. 또한 보다 다양한 유형의 맵 데이터들을 사용할 수 있도록 스케치맵을 구성하는 등의 시스템 확장에 대한 연구가 필요하다.

참고 문헌

[1] G. Carsten, K. Alfons, K. Christoph, M. Guido, "Partition-Based Clustering in Object Bases,"

- in Proc. FODO '93, Lecture Notes in Computer Science, Vol.730 (Springer, Chicago), pp.301-316, 1993.
- [2] J. Banerjee, W. Kim, S-J. Kim, and J. F. Garaza, "Clustering a DAG for CAD databases," IEEE Transactions on Software Engineering, Vol.14, No.11, pp.1684-1699, 1998.
- [3] Jayavel Shanmugasundaram, Kristin Tufte, Chun Zhang, Gang He, David J.Dewitt, Jeffrey F. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," in Proc: VLDB '99 (Morgan Kaufmann, Edinburgh), pp.302-314, 1999.
- [4] D. Florescu and D. Kossman, "Storing and Querying XML, Data Using an RDBMS," IEEE Data Engineering Bulletin, Vpl.22, No.3, pp.27-34, 1999.
- [5] T. Feng, J. David, DeWitt. C. Jianjun, Z. Chun, "The Design and Performance Evaluation of Alternative XML Storage Strategies," SIGMOD Record, Vol.31, No.1, pp.5-10, 2000.
- [6] L. Bergroth, H. Hakonen, and T. Raita, "A Survey of Longest Common Subsequence Algorithms," SPIRE 2000 Proceedings. Seventh International Symposium, pp.39-48, 2000.
- [7] M. Xianfeng, L. Daofeng, M. L. Lee, A. Jing, "OrientStore: A Schema Based Native XML Storage System," in Proc: VLDB '03, (Morgan Kaufmann, Berlin), pp.1057-1060, 2003.
- [8] Jiefeng Cheng, Ge Yu, Guoren Wang, Jeffrey Xu Yu, "PathGuide: An Efficient Clustering Based Indexing Method for XML Path Expressions," in Proc. DASFAA '03, (IEEE Computer Society, Kyoto), pp.257-264, 2003.
- [9] 이정진, 원동기, 김법근, 두길수, 안동언, 정성중, "MMORPG에서 VML를 이용한 동적 지역분할 (Dynamic Local-Partition Using VML in MMORPG)", 한국정보처리학회, 제24회 추계학술발표대회논문집, pp.1487-1490, 2005.
- [10] 최일한, 문용기, 김형주, "XML 데이터의 경로 유사성에 기반한 클러스터링 기법", 정보과학회 논문지:데이터베이스, 제33권 제3호, pp.342-352, 2006.
- [11] R. Nayak and S. Xu. XCLS: A Fast and Effective Clustering Algorithm for Heterogenous XML Documents. in the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Singapore: LNCS pp.292-302, 2006.
- [12] Ludovic Denoyer, Patrick Gallinari, The Wikipedia XML corpus, ACM SIGIR Forum, Vol.40, No.1, June 2006.
- [13] Ludovic Denoyer, Patrick Gallinari, Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents, ACM SIGIR Forum, Vol.41, No.1, pp.79-90, June 2007.
- [14] 윤근정, 김혜영, 전철민, "모바일 GIS를 위한 벡터 데이터 경량화 기법", 한국GIS학회, Vol.16, No.2, pp.207-218, 2008.
- [15] <http://code.google.com/intl/ko/apis/kml/documentation/>
- [16] <https://developer.mozilla.org/ko/SVG>
- [17] <http://yeonisalive.net/javascript/MindWeb001.php>
- [18] <http://www.acm.org/sigmod/record/xml>

저 자 소 개

김 정 숙(Jung-Sook Kim)

정희원



- 1984년 2월 : 광운대학교 전산학과(이학사)
- 1988년 2월 : 동국대학교 교육대학원 전산학과(교육학석사)
- 1999년 2월 : 동국대학교 컴퓨터공학과(공학박사)

- 2000년 ~ 2001년 : 김포대학 컴퓨터계열 교수
- 2001년 3월 ~ 현재 : 삼육대학교 컴퓨터학부 교수
<관심분야>: 웹프로그래밍, 임베디드시스템, 모바일 컴퓨팅, 프로그래밍언어, 컴파일러

이 야 리(Ya-Ri Lee)

정회원



- 1990년 2월 : 고려대학교 전자전산학과(공학사)
- 1999년 2월 : 동국대학교 교육대학원 전산학과(이학석사)
- 2002년 8월 : 동국대학교 컴퓨터공학과(공학박사)
- 1990년 ~ 1992년 : (주)롯데캐논 기술연구소 연구원
- 2000년 ~ 2001년 : 경인여자대학 인터넷비즈니스학과 전임강사
- 2004년 4월 ~ 현재 : (주)이비전코리아 이사
<관심분야> : 프로그래밍언어, 컴파일러, 모바일 컴퓨팅, 웹프로그래밍, 디지털콘텐츠

홍 경 표(Kyung-Pyo Hong)

정회원



- 2000년 2월 : 동국대학교 컴퓨터공학과(공학사)
- 2002년 2월 : 동국대학교 컴퓨터공학과(공학석사)
- 2002년~2004년 : Wireless Japan 이사
- 2004년 4월 ~ 현재 : (주)이비전코리아 이사
<관심분야>: 프로그래밍언어, 시스템엔지니어링, 게임 운용 프로그램, 소프트웨어공학, 웹프로그래밍