

유비쿼터스 환경에서 향상된 파일 관리

현우석

한국성서대학교 정보과학부

e-mail: wshyun@bible.ac.kr

Enhanced File Management in Ubiquitous Environments

Woo-Seok Hyun

School of Information Science, Korean Bible University

요 약

유비쿼터스 컴퓨팅의 패러다임은 사용자 컨텍스트에 반응하는 컴퓨팅 환경을 구축하는데 있다. 유비쿼터스 환경에서 이상적인 파일 시스템은 현재 컨텍스트를 성공적으로 인식하여 파일관리를 자동화할 수 있는 시스템이다. 유비쿼터스 파일관리에서의 핵심은 휴리스틱(heuristics)에 기반한 클러스터링 방법을 시스템에 적용함에 의해서 수립될 수 있다. 적용된 휴리스틱은 전통적인 파일 시스템에서 파일을 손으로 다루기 위해 사용자에게 의해서 파일 속성으로 사용되어졌던 것들이다. 파일 속성은 파일들을 가장 적절한 작업 컨텍스트로 관련짓기 위해서 사용될 수 있고 파일간 관계(inter-file relationship)를 이끌어 낼 수 있다. 본 논문에서는 파일 시스템으로부터 주어진 파일 정보를 연결하는 방법에 대해 논의한다. 이 방법은 개별적인 파일보다는 컨텍스트 관련 작업 집합으로서 파일 관리를 가능하게 해 준다. 조사는 규칙적인 컴퓨터 사용자를 대상으로 수행되었으며 집계된 결과는 본 논문에서 제안한 컨텍스트 기반 파일 클러스터링 방법을 지지해 준다. 수행된 실험 결과는 파일 휴리스틱에서 본 논문에서 제안하는 모델이 파일간 의미있는 관계를 찾아주는 효과적인 결과를 보여주게 되었다.

1. 서론

유비쿼터스 컴퓨팅에서는 장면(scene)뒤에 복잡한 시스템 인터페이스와 제어 구조를 놓고 사용자에게는 사용하기 쉬운 인터페이스를 제공해 주고 있다. 이러한 인터페이스는 사용자와 기계들 사이에서 의사소통을 가능하게 해 준다. 유비쿼터스 환경에서 접근 가능한 인터페이스는 의사소통을 단순하고 자연스럽게 확장시켜 준다. 유비쿼터스 환경은 컨텍스트(context)에 반응함에 의해 인터페이스를 제공해 주어야 한다. 컨텍스트는 환경의 기능, 환경에서 사용자의 역할 그리고 유비쿼터스 컴퓨팅 프레임워크로부터 사용자가 필요로 하게 될 지도 모르는 관련된 서비스를 사용자에게 확장시켜 주는 것으로 정의될 수 있다. 프레임워크는 여러 개의 입력 센서들로부터 컨텍스트를 결정해 주어야 한다. 이러한 정보는 프레임워크 기능들 중에서 컨텍스트 모델을 구축하기 위해서 사용될 수 있다.

파일 관리 시스템은 컴퓨팅 환경에서 중요하다. 파일 시스템은 파일 관리를 통해서 사용자에게 인터페이스를 제공해 준다. 이러한 파일 관리 시스템은 시스템에서 다른 파일들 간의 의미없는 관계를 인식하고 그들 사이를 컨텍스트로 관련짓지 않는다. 유비쿼터스 환경을 위한 적절한 파일 관리 설계는 컨텍스트에 기반을 두고 있다. 전통적인 파일 시스템은 컨텍스트 혹은 파일 관계에서의 컨텍스트를 인식하지 못하여 컨텍스트 기반 파일 관리 설계를 수행하는데 실패를 하게 된다. 새로운 파일 관리 시스템은

파일들을 컨텍스트로 관련지을 수 있고 컨텍스트에 민감한 파일 유용성을 제공해 준다.

본 논문에서는 파일 관리에서 컨텍스트를 결합함에 의해서 유비쿼터스 컴퓨팅의 원칙을 수행하는 향상된 파일 관리 시스템을 제안한다. 하나의 파일은 태스크에 의해서 요구되어 지거나 생성되어지는 논리적인 데이터의 구별(separation)이다. 그러므로 데이터와 태스크는 같은 컨텍스트의 부분이다. 향상된 파일 관리 시스템은 파일과 컨텍스트 사이의 관계를 인식할 수 있어야 한다. 같은 컨텍스트와 관련된 파일 사이의 관계는 규칙적으로 저장된 파일 속성으로부터 유도된 의미(semantic)로부터 추출될 수 있다. 본 논문에서는 파일 이름, 파일 생성 날짜, 파일 유형, 최근 접근 날짜, 파일 접근 빈도, 파일 내용, 위치 등과 같은 파일 속성에 근거한 컨텍스트 결정 휴리스틱을 사용한다. 고립되어 사용되어진 각 휴리스틱은 파일들을 컨텍스트로 관련짓기 위하여 충분한 정보를 제공하지 않을 수도 있다. 다양한 속성들을 고려하기 때문에 파일 사이의 컨텍스트 관계를 잘 인식할 수 있게 해 준다. 그러므로 이러한 속성들에서 데이터 마이닝을 효과적으로 수행함에 의해서 파일을 컨텍스트로 구분지을 수 있게 된다.

본 논문에서 설계한 프레임워크는 위에서 언급된 기능들을 이루기 위해서 유비쿼터스 파일 시스템을 수행하고 사용자에게 컨텍스트 기반 파일 클러스터링을 제공해 준다. 이 프레임은 유사한 컨텍스트와 관련된 파일들을 함께 패키지와화하고 사용자 파일 접근 패턴을 발견하여 가장 적

질한 파일 집합을 제공해 줌으로써 현재 컨텍스트를 추론해 준다.

본 논문에서는 일반 컴퓨터 사용자를 대상으로 조사한 실험 결과를 가지고 있다. 이 결과들은 컨텍스트에 의해 파일들을 구조화하기 위한 시도에서 본 논문에서 제안하는 휴리스틱을 증명해 준다. 휴리스틱과 파일 접근 패턴 알고리즘에서 수행된 실험에서는 본 논문에서 제안하는 모델이 파일간 의미있는 관계를 찾아주는 효과적인 결과를 보여 주었다.

2. 연구배경

몇몇 연구들[1-3]은 파일들을 효과적으로 예측하는 것을 주제로 다룬다. 이 연구들은 저장되어지기 위한 명백한 파일 사용에 대해 논의한다. 응용이 사용되어질 때, 형성된 부분적인 파일 접근 패턴은 저장된 패턴과 비교되고, 트리에서 저수준의 파일을 검색하기 위해서 프리패치(prefetch)가 수행된다. 이런 연구들은 필요하다면 트리를 분배하는 저장 크기를 고려한다. [1] 연구의 경우에 일치(match)의 근접성은 휴리스틱에 의해서 정의되는데, 이 휴리스틱은 일치의 근접성이 일반적으로 40%의 파일을 가지게 된다. Kroeger et al.[4]는 확률 그래프와 같은 파일 접근 패턴을 형성하기 위해 다른 모델을 제안한다.

서버가 없는 파일 시스템[5]에서는 각각의 시스템이 파일과 그들과 관련된 리스트를 유지하면서 사용자 파일을 지역적으로 저장한다. 파일들은 최소한으로 반복되고 파일들이 자연스럽게 저장되는 곳으로부터 요청에 대해 처리해 준다.

Chen과 Finin[6]은 에이전트 기반 입는 컴퓨터(pervasive computing) 프레임워크에 대해 연구했다. 이 프레임워크에는 장치로부터 정보를 받아들이는 지능형 브로커(broker)와 에이전트 등이 있다. 본 연구에서는 브로커가 사용자 파일 접근 패턴과 관련된 정보를 받아들일 수 있고 파일간 의미있는 관계를 추론할 수 있는 브로커와 같은 프레임워크를 제공하고자 한다.

사용자 프로파일링(profiling)은 지역적인 파일들과 시스템에 제한되는 것이 아니다. Chen et al.[7]은 사용자의 웹 브라우징 습관을 분석하고 웹에서 사용자에게 좀더 좋은 서비스를 제공하기 위해서 이러한 접근법을 사용한다. Kim et al.[8]은 사용자에게 보다 좋은 웹 브라우징 경험을 제공하는데 있어서 사용자의 흥미있는 프로파일의 사용과 사용자 프로파일링 접근법을 재연구했다. 이러한 연구들은 파일 사이의 의미있는 관계들이 실제적인 내용에 의해서 잘 표현될 수 있다는 것을 보여준다. 본 논문의 프레임워크에서 휴리스틱은 파일이 특정 컨텍스트나 태스크에 속한다는 것을 결정하기 위해서 파일 내용(content)을 사용한다. 또한 파일 사이의 의미있는 유사성을 유도해 내기 위해서 파일 클러스터링 방법[9-10]을 적용한다.

3. 컨텍스트 기반 클러스터링

3.1 파일 클러스터링을 위한 휴리스틱

유비쿼터스 컴퓨팅의 패러다임은 사용자에게 컨텍스트

기반한 서비스를 제공하는 것을 목적으로 한다. 현재 파일 시스템에서 파일 관리 작업은 사용자 컨텍스트에 따른 파일 관리 시스템을 제공하기보다는 전적으로 사용자에게 남겨지고 있다. 본 논문에서는 사용자 컨텍스트를 가지면서 사용자 파일을 강하게 결합시키는 컨텍스트 관련 파일 집합의 개념을 제안한다. 컨텍스트 관련 파일 집합은 지능형 래퍼(wrapper)를 형성하는데, 이것은 파일 시스템에서 파일에 대한 유비쿼터스 환경에 대한 컨텍스트 기반 서비스를 제공한다. 컨텍스트는 사용자와 환경을 지니는 파일의 강력하고 의미있는 관계를 정의한다. 단순한 태스크와 관련된 파일들은 단순한 컨텍스트 관련 파일 집합으로 결합된다.

본 논문에서 제안하는 방법은 다음과 같이 서로 다른 파일 특성을 다루는 몇 가지 휴리스틱에 의존한다.

위치(locality): 많은 응용프로그램에서는 한 장소에 하나의 프로젝트와 관계된 파일들을 저장하곤 한다. 또한 사용자들은 일반적으로 다른 컨텍스트와 관련된 파일들을 다른 폴더에 위치시킴에 의해서 구분한다. 이 특성에 의해 상당한 가중치가 주어질 수 있고 파일사이의 컨텍스트 관계의 좋은 표시로 증명될 수 있다.

파일 명명법(File Nomenclature): 일반적으로 파일 이름은 파일의 내용 혹은 목적의 간결한 표시로 주어진다. 유사한 이름을 지니는 파일들은 유사한 목적을 지니거나 유사한 종류의 데이터를 지니는 것으로 암시된다. 이 정보는 정확한 컨텍스트와 파일을 관계 맺고 태스크 관련된 파일들을 검색하기 위해서 사용되어질 수 있다.

File Type(파일 유형): 이 휴리스틱은 시스템에서 파일을 조직화하기 위한 방법과 자주 관련된다. 하나의 파일 시스템은 파일 확장을 고려함에 의해서 어떤 사용자의 다른 파일유형을 위해서 다른 디렉토리 트리를 만들기도 한다.

File Count(파일 내용): 가장 복잡한 휴리스틱 중의 하나는 사용자 파일의 내용을 고려하는 것이다. 사용자 파일의 내용은 다른 파일들 사이에서 일치의 범위를 결정하기 위해서 다른 파일의 내용과 검색되기도 하고 조화되기(match)도 한다.

Creation time/date(생성 시간/날짜): 생성 날짜가 가까운 영역에 있는 파일들은 서로서로 관련지어질 수 있다. 일반적인 작업은 시간의 흐름에 따라 수행되어지고 단순한 태스크는 시간적으로 계속됨에 의해서 응집성을 유지하게 된다. 파일 내용(File Content) 휴리스틱이 함께 사용되어질 때 파일의 다른 버전을 해결하는데 도움을 줄 수 있다.

Access Patterns(접근 패턴): 병행 상태로 접근되는 파일들은 특별한 처리과정/응용과 관련된다. 이 휴리스틱은 파일 사이의 관계는 병행적으로 접근될 수 있다는 것을 제안한다. 이러한 파일과 파일 접근 특성은 휴리스틱을 위한 기본으로서 제안되는데, 이것은 관련된 파일들을 컨텍스트 관련 파일 집합으로 함께 클러스터링한다.

3.2 파일 클러스터링 알고리즘

본 논문에서는 사용자에게 가장 관련된 파일을 컨텍스트로 제공하기 위한 해결책을 제안한다. 이렇게 하기 위해

서 사용자 파일 접근 패턴은 가장 요구되는 파일을 결정하기 위해서 그리고 이러한 파일들로부터 컨텍스트 관련 파일 집합을 생성하기 위해서 분석된다. 그림 1은 파일 클러스터링 알고리즘의 흐름을 보여 준다.

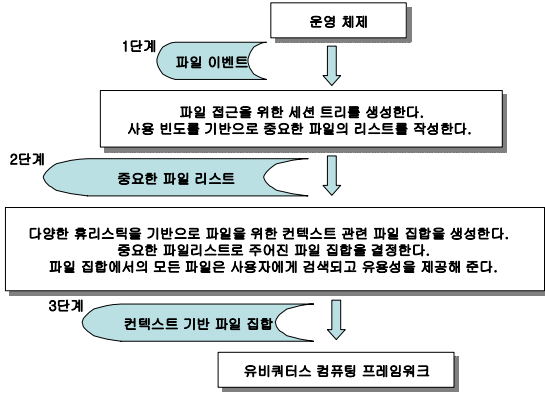


그림 1 파일 클러스터링 알고리즘

3.3 사용자 모델링

클러스터링을 위해서 사용되어지는 속성과 적용되는 휴리스틱의 실제적인 순서는 사용자에게 따라 달라지기도 한다. 사용자 모델은 각 사용자를 위해서 세워져야 하는데, 이것은 사용자를 위해서 따라가야 하는 클러스터링 접근법을 정의한다. 다음의 알고리즘은 하나의 모델이 구축될 수 있는 방법에 대해 개요를 보여준다.

사용자 모델을 구축하는 알고리즘

```
Define Final_set[] as the final set of clusters
Define Attr_set[] as set of attributes
Define dataset as set of instances of file events
```

```
Function Find_optimal_clusters(cluster, attribute)
  If size(cluster) <=Threshold Then
  Begin
    Final_set[++count]=(cluster, attribute)
  Exit
  End
  For(j=0; j<sizeofarray(Attr_set[]); j++)
  Clust[j][]=Cluster(cluster, Attr_set[j])
  If(distribution(clust[j][])=acceptable) then
  Begin
    For(k=0;k<sizeofarray(clust[j][]);k++)
    Find_optimal_clusters(clust[j][k], Attr_set[j])
  End For
  End If
  End For
End Function
```

Main method

Begin

```
REM Create initial clusters clus[1..n] using all
REM attributes
Clus[]=Cluster(dataset, all)
For(i=1;i<=n;i++)
  REM find the optimal clusters using recursive
  REM fundtion
  Find_optimal_clusters(clus[i], all)
End For
REM use Final_set to train user model such as a
REM Neural Network model
End
```

4. 평가와 실험

4.1 사용자 파일 시스템에서의 조사

조사는 파일 시스템에서 파일들을 조직화하는 방법을 결정하기 위하여 50명의 컴퓨터 공학 학생들을 대상으로 수행되었다. 조사의 결과는 표1과 같다.

표 1 사용자 파일 시스템 조사

| Category | Subcategory | Ranking |
|--|-----------------------------|----------|
| Application | Email | 1 |
| | Word Processing | 2 |
| | Compilers | 3 |
| | Spread Sheet | 4 |
| | | Quantity |
| Path | User Defined Path | 26 |
| | Default Path | 1 |
| Directory# (per user) | 1-5 | 5 |
| | 6-10 | 9 |
| | 11-20 | 10 |
| | Over 20 | 9 |
| Organizing files according to file type or file name | File type | 20 |
| | File name | 17 |
| Depth of file structure | 0-2 | 9 |
| | 3-5 | 18 |
| | 6-10 | 4 |
| Naming convention | Random | 3 |
| | Project name | 19 |
| | Generic name + Sequence# | 3 |
| | Generic name | 5 |
| | Name + date | 7 |

4.2 실험

본 논문에서는 .NET의 FileSystemWatcher 구성요소를

사용하여 파일 시스템 이벤트 추적 구성요소를 수행시켰고 추적된 파일 이벤트의 XML 표현이 생성되었다. 세션 트리 생성과 통합을 위한 구성요소는 마이크로소프트 C#.NET에서 수행되었다. 구성요소로부터 생성된 결과는 세션 트리를 기억장소에 갱신시키고 가장 최근에 자주 사용된 파일이 분석되기 위해 사용되었다.

본 논문에서는 다음과 같이 선택된 휴리스틱에 기반한 파일 클러스터링에 대한 실험을 하였다. 첫째는 파일이름과 클러스터링에 기반한 파일 패스이다. 둘째는 타임 스탬프(Time Stamp)와 접근 빈도에 의한 클러스터링이다. 이 실험을 위해서 컴퓨터 공학 학생 파일 접근 이벤트를 표현하는 데이터집합이 수집되었는데, 데이터집합은 2시간동안 수집된 5375개의 레코드로 구성되어 있다.

(1)파일이름과 파일패스에 기반한 클러스터링: 클러스터링 알고리즘에 의해서 클러스터링을 수행한 결과 10개의 클러스터가 형성되었다. 구분지어진 클러스터 안에 있는 파일들은 디렉토리 계층구조에서 파일 이름과 위치가 유사성을 지니고 있었다. 그림 2는 하나의 예를 보여 준다.

| | |
|---------------------------|---------------------------|
| 1)Instance:878 | 2)Instance:901 |
| Filename:glsnecb.ttf | Filename:gopika.ttf |
| Filepath:winnt\fonts\ | Filepath:winnt\fonts\ |
| Timestamp:7\29\2008_21:36 | Timestamp:7\29\2008_21:36 |
| Cluster:cluster5 | Cluster:cluster5 |

그림 2 클러스터된 파일의 예

(2)타임 스탬프와 접근 빈도에 기반한 클러스터링: 타임 스탬프와 접근 빈도에 기반한 클러스터링을 실험한 결과 6개의 클러스터가 형성되었다. 그림 3는 하나의 예를 보여 준다.

Instance:1093
 Filename: journal_paper.doc
 Filepath: document_and_settings\administrator\desktop\
 Timestamp: 7/29/2008_19:23
 Cluster:cluster2

Instance:1107
 Filename: architecture_diag.gif
 Filepath: document_and_settings\administrator\desktop\
 Timestamp: 2/29/2008_23:05
 Cluster:cluster2

그림 3 클러스터된 파일의 예

5. 결론

본 논문에서는 파일의 컨텍스트를 식별하고 관련된 파일을 파일 집합으로 선택해 주는 컨텍스트 기반 클러스터링 모델을 제안한다. 본 논문에서 제안된 모델은 유비쿼터스 시스템에 대한 파일 선택 능력을 제공해 주며, 파일 관리의 효율성을 높여 주고 역동적인 방식으로 컨텍스트 기

반 파일 선택 능력을 향상시켜 준다. 본 논문에서 수행한 조사는 휴리스틱이 파일 집합 선택과 관련된 컨텍스트로서 수행되어 지며, 데이터를 가지고 한 실험은 우리가 제안했던 다양한 휴리스틱에 의해서 파일의 성공적인 분류를 보여준다. 따라서 본 논문에서 제안한 방법은 효율성의 극대화를 요구하는 유비쿼터스 컴퓨팅에 기여할 수 있다.

참고문헌

- [1] C. C. Tait and D. Duchamp, "Detection and Exploitation of File Working Sets," In Proc. Eleventh Intl. Conf. on Distributed Computing Systems, pp. 2-9, IEEE, May 1991.
- [2] Hui Lei and Dan Duchamp, "An Analytical Approach to File Prefetching," USENIX Conference CA, January 1997.
- [3] Meenakshi Arunachalam, Alok chaudhary, Brad Rullman, "Implementation and Evaluation of Prefetching in the Intel Paragon Parallel File system," 10th International Parallel Processing Symposium (IPPS '96) April 15-19, 1996.
- [4] Thomas M. Kroeger and Darell D. E. Long. "The Case for Efficient File access Pattern Modeling," The Seventh Workshop on Hot Topics in Operating Systems, 1999.
- [5] Thomas E. Anderson, Michael D. Dahlin, Jeanna M. Neeffe, David A.Patterson, Drew S. Rosellim Randolph R, Wabg, "Severless Neowork File Systems," 15th ACM Symposium on OS principles, Dec. 1995.
- [6] Herry Chen and Tim Finin, "An Ontology for Context Aware Pervasive Computing Environments," IJCAI03 workshop on Ontologies and distributed systems.
- [7] P. Chan, "A non-invasive learning approach to building web user profile," KDD-99 Workshop on Web Usage Analysis and User Profiling, pp. 7-12, 1999.
- [8] Hyoung R. Kim and Philip K. Chan., "Learning Implicit User Interest Hierarchy for Context in Personalization, IUI '03, 2003.
- [9] Nicolas Anquetil, Timothy Lethbridge, "Extracting concepts from File names; a new file clustering criterion," The 20th International Conference on Software Engineering, 1998.
- [10] V. Kartik, G. Arvind, L. Yogyung, "Intelligent File Management in Ubiquitous Environments", Proceedings of the 2005 ACM symposium on Applied computing SAC '05, pp. 1621-1625, March 2005.