韓國컴퓨터情報學會 論文誌
第16卷 第5號, 2011. 05.

2011-16-5-4-3

그룹화된 폭소노미 구축을 위한 프레임워크와 지원도구의 개발

강 유 경*, 황 석 형**

An development of framework and a supporting tool for organizing Grouped Folksonomy

Yu-Kyung Kang*, Suk-Hyung Hwang**

요 약

폭소노미는 웹에 존재하는 리소스에 대해 구성원이 자유롭게 선택한 태그를 붙여서 정보를 체계화하는 새로운 분류 체계이다. Del.icio.us, Flickr, YouTube 등과 같은 최근의 폭소노미 기반 협력 태깅 시스템들을 살펴보면, 폭소노미 데이터를 체계화하고 유용한 서비스를 제공하기 위해 폭소노미의 각 구성요소들(사용자들, 태그들, 리소스들)을 그룹핑하기 위한 기능들을 추가하고 있다. 본 논문에서는, 일반적인 폭소노미에 "그룹(Group)"이라는 개념을 도입하여 확장한 그룹화된 폭소노미(Grouped Folksonomy) 모델과 그룹핑 관련 기본 연산들(Group Aggregation, Group Composition, Group Intersection, Group Difference)을 제안한다. 또한, 본 논문에서는 그룹화된 폭소노미의 구축과 기본연산 수행을 지원하는 도구(GFO)의 개발과 활용사례를 소개한다. 본 연구에서 제안된 모델과 기본 연산들로 구성된 그룹화된 폭소노미 구축을 위한 프레임워크는 폭소노미의 각 구성요소들을 그룹핑하여 그룹화된 폭소노미를 구성하고 보다 유용한 정보들을 추출 및 통합하여 체계화할 수 있는 토대를 제공한다.

▸ 키워드 : 폭소노미 기반 웹 어플리케이션, 그룹화된 폭소노미

Abstract

A folksonomy is a new classification approach for organizing information by users to freely attach one or more tags to various resources published on the web. Recently, in order to provide useful services and organize the folksonomy data, many collaborative tagging systems based on folksonomy offer additional functionalities for grouping each elements of a folksonomy. In this paper, organization framework for grouped folksonomy is proposed. That is, we suggest the grouped folksonomy model that is an extended folksonomy with the concept of "group" and

• 제1저자, 교신저자 : 강유경
• 투고일 : 2010. 11. 22, 심사일 : 2010. 12. 20, 게재확정일 : 2010. 12. 24.
* 선문대학교 컴퓨터공학부 연구교수(Dept. of Computer Engineering, Sunmoon University),
** 선문대학교 컴퓨터공학부 교수(Dept. of Computer Engineering, Sunmoon University)

fundamental operations(Group Aggregation, Group Composition, Group Intersection, Group Difference) for grouping of folksonomy elements. Also, we developed a supporting tool(GFO) that constructs grouped folksonomy and executes fundamental operations. And we introduce some cases using the fundamental operations for grouping of each elements of folksonomy. Based on suggested our approach, we can construct grouped folksonomy and organize and extract useful information from the folksonomy data by grouping each elements of a folksonomy.

▸ Keyword : Folksonomy based Web Application, Grouped Folksonomy, Group Aggregation, Group Composition, Group Intersection, Group Difference

I. 서 론

폭소노미(folksonomy)는 웹에 존재하는 리소스에 대해 구성원이 자유롭게 선택한 태그(tag)를 붙여서 정보를 체계화하는 새로운 분류 체계로써, 일반적인 대중을 의미하는 folk와 분류법을 의미하는 taxonomy의 합성어이다. 폭소노미는 흔히 협력 태깅(Collaborative tagging) 또는 소셜 태깅(Social tagging)으로 알려져 있으며, 사용자들이 자유롭게 참여하는 bottom-up 방식의 분류를 기반으로 하고 있어서, 시시각각으로 변하는 웹의 리소스를 분류하는데 적합한 방법으로 WWW 분야에 널리 사용되고 있다. 폭소노미 기반의 유명한 시스템으로는 북마크를 공유하는 Del.icio.us[1], 사진을 공유하는 Flickr[2], 동영상을 공유하는 YouTube[3], 논문 및 북마크를 공유하는 Bibsonomy[4] 그리고 멀티미디어를 공유하는 GroupMe[5] 등이 있다.

이와 같은 폭소노미 기반 시스템에서는 유용한 협력태깅을 지원하기 위해 특정 리소스에 여러 사용자가 자유롭게 태그를 할당 할 수 있을 뿐만 아니라 공통 관심사를 갖는 사용자들이 함께 리소스에 태깅 할 수 있고, 태그들을 재사용하고 공유할 수 있으며, 관련된 리소스들을 추천하는 등의 다양한 기능들을 제공한다. 특히, 최근 폭소노미 기반 협력 태깅 시스템들을 살펴보면, 폭소노미 데이터를 체계화하고 유용한 서비스를 제공하기 위해 폭소노미 구성요소들(사용자들, 태그들, 리소스들) 각각을 그룹핑하기 위한 기능들을 추가하고 있다[1-7]. 이와 같은 관련 연구들을 토대로, 본 연구에서는 폭소노미 구성요소들을 그룹핑하기 위한 프레임워크를 제안한다. 구체적으로, 본 논문에서는, 일반적인 폭소노미 정의에 “그룹(Group)”이라는 개념을 포함하여 확장한 그룹화된 폭소노미(Grouped Folksonomy)를 제안하고, 그룹화된 폭소노미를 구축하고 폭소노미의 각 구성요소들을 그룹핑하여 폭소노미 데이터를 체계화하기 위한 기본적인 연산들을 정의한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련연구에 대해 설명하고, 3장에서는, 본 연구에서 제안한 그룹화된 폭소노미 모델을 정의한다. 4장에서는, 그룹화된 폭소노미의 각 구성요소들을 그룹핑하고 구조화하기 위한 기본적인 연산들을 정의한다. 5장에서는 그룹화된 폭소노미를 구성하기 위한 기본 연산들을 지원하는 시스템의 개발과 사용사례를 설명하고, 6장에서는 결론 및 향후 연구과제에 대해서 설명한다.

II. 관련 연구

협력 태깅은 폭소노미 데이터를 체계화하기 위한 강력한 방법으로서, 대부분의 폭소노미 기반 애플리케이션들은 협력 태깅을 지원한다. 제2장 관련 연구에서는 본 연구와 관련된 기존의 폭소노미 기반의 시스템들[6-14]에 대하여, 처리대상 컨텐츠와 사용자들 사이에 연결성(Connectivity between Users), 태그들 사이에 연결성(Connectivity between Tags), 리소스들 사이에 연결성(Connectivity between Tags)과 같은 특징들을 종합분석하여 표1에 정리하였다.

각 시스템의 사용자들은 “Links”와 “Groups”에 의해 다른 사용자와 연결되어 있다. “Links”는 사용자가 다른 사용자와 직접적으로 연결되어 있음을 나타낸다. “Groups”는 사용자가 특정 목적에 따라 사용자들의 클러스터들을 만들 수 있음을 의미한다. Del.icio.us 시스템은 사용자들을 그룹핑하기 위해 “Network”와 “Network Bundle”기능을 제공한다. “Network”는 좋아하는 사용자들의 컬렉션(people aggregator)이다. “Network Bundle”은 사용자들이 만들어 놓은 “Network”안에서 사용자들을 체계화하기 위해 그룹핑하는 기능이다. Flickr는 사진이 업데이트될 때 이를 알 수 있도록 다른 사용자들을 이웃으로 추가 할 수 있으며, 이웃을 “friends”, “family” 또는 “friends and family”로 분류할 수 있다. YouTube와 Bibsonomy 시스템들은 좋아하는 사용자들의 클러스터로써 각각 “Friends”와 “MyFriends”를 제공한다. GroupMe에서는 사용자의 리소스 그룹에 대한 “기여자(Contributor)”로써 다른 사용자를 추가할 수 있다.

표 1. 폭소노미 기반 시스템들의 특징들
Table 1. Features of folksonomy-based systems

	Contents	Connectivity between		
		Users	Tags	Resources
del.icio.us[1]	Bookmarks	Links, Groups	System suggestive, User specific	None
Flickr[2]	Photos & Videos	Links, Groups	System suggestive	Groups
YouTube[3]	Videos	Links, Groups	None	Groups
Bibsonomy[4]	Bookmarks & Phblications	Links, Groups	System suggestive, User specific	None
GroupMe[5]	Multimedia	Groups	System suggestive	Groups

Del.icio.us, Flickr, Bibsonomy, GroupMe는 태그 추천과 태그 기반 정보검색을 지원하기 위해 태그들 사이의 연결성을 제공한다. “System suggestive”는 시스템이 같은 사용자에 의해 사용된 태그들과 다른 사용자에 의해 같은 리소스에 할당된 태그들을 기반으로 사용자에게 가능한 태그들을 추천해준다. “User specific”은 시스템이 사용자들이 만들어 둔 태그들의 컬렉션을 제공한다. Del.icio.us는 태그들의 컬렉션으로서 “Tag bundle”을 제공한다. “Tag bundle”은 태그들의 그룹을 체계화하기 위한 단순한 방법이다. 예를 들어, 사용자는 “사과”, “복숭아”, “포도” 태그들을 “과일”이라는 하나의 묶음으로 만들고자 할 때 “Tag bundle”기능을 사용하면 된다. 이와 같이 “Tag bundle”기능을 사용함으로써, 사용자는 “과일”로 태깅된 북마크들만 필터링하여 찾아 볼 수 있다. Bibsonomy에서는 사용자가 자신의 태그들 사이에 상·하위 관계를 설정할 수 있다. 사용자가 상위 태그로 검색했을 때, 상위 태그뿐만 아니라 상위 태그에 종속된 하위 태그들로 태깅된 모든 북마크들을 찾아낼 수 있다. 예를 들어, 사용자가 “ubuntu”를 “linux”의 종속 태그로 설정했을 때, 사용자가 “linux”태그로 검색을 하면, “linux” 태그뿐만 아니라 “ubuntu”태그가 할당된 북마크들도 검색 가능하다. Filckr에서는, 사용자가 이미 어떤 태그들이 할당된 사진에 새로운 태그들을 추가하고자할 때, 시스템은 그 사용자에 의해 사용된 태그들을 단지 보여주기만 한다. GroupMe는 다른 사용자가 같은 리소스에 할당한 태그들을 보여준다.

Flickr, YouTube, GroupMe에서는 리소스들의 그룹핑을 위해 사용자들에게 유용한 기능들을 제공한다. Flickr는 사진들을 모아놓는 “Set” 기능을 지원한다. YouTube에서는 사용자가 “Favorites”, “Playlists”, “QuickList” 기능을 사용하여 비디오들을 체계화 할 수 있다. “Favorites”은 좋아하는 비디오들의 리스트들이고, “QuickList”는 나중에 보기 위해 모아놓은 비디오들의 리스트이다. “Playlists”는 YouTube에서 봤던 비디오들의 컬렉션이다. GroupMe는 리소스들의 집합을 나타내는 “group”기능을 제공한다. 하나의 그룹 또한 리소스로서 취급되기 때문에 하나의 그룹은 여러 개의 다른 그룹을 포함할 수 있으며, 각 그룹에 태그들을 할당할 수 있다.

이상의 관련 연구에서처럼, 최근에는 다양한 협력 태깅 시스템들 사이에 최근 폭소노미 요소들을 그룹핑하는 것이 중요한 이슈가 되었음에도 불구하고 현재 폭소노미 데이터를 그룹핑하기 위한 공통적이고 표준화된 모델과 기본기능들이 제공되고 있지 않다. 따라서, 본 논문에서는 “그룹(Group)” 개념을 포함하여 일반적인 폭소노미를 확장한 그룹화된 폭소노미 모델을 정의하고, 제반 기본연산들을 제안한다.

III. 그룹화된 폭소노미

본 장에서는, 일반적인 폭소노미의 정의와 본 연구에서 제안한 그룹화된 폭소노미에 대한 기본적인 정의를 소개한다.

[정의1] 폭소노미 F:= (U, T, R, Y)는 사용자들(Users)의 집합 U와 태그들(Tags)의 집합 T, 리소스들(Resources)의 집합 R, 그리고 U와 T와 R사이의 관계 Y ⊆ U × T× R로 구성된다[3]

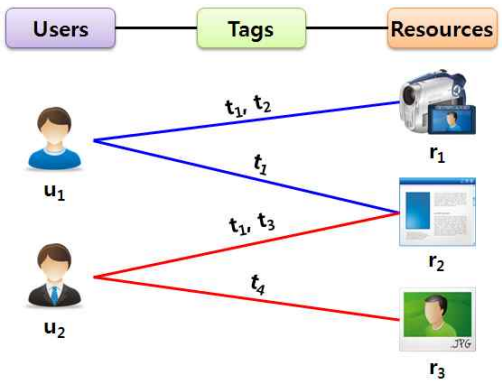


그림 1. 폭소노미의 예
Fig. 1. An example of folksonomy

del.icio.us, Flickr, YouTube, Bibsonomy 등과 같은

대부분의 협력 태깅 시스템들은 일반적인 폭소노미 모델(정의 1)을 기반으로 개발되었다. 폭소노미에서 사용자는 논문이나, 사진, 동영상등 웹에 존재하는 다양한 리소스들에 관련 태그들을 할당한다. 태그 할당(tag assignment)은 사용자, 태그, 리소스사이에 삼항관계(ternary relation)를 나타내며, (u, t, r)∈Y로 표현하고, "사용자 u는 리소스 r에 태그 t를 할당하였다."라는 것을 의미한다. 예를 들어, 그림 1에서, u_1은 리소스 r_1에 태그 t_1과 t_2를 할당하였으며, (u_1, {t_1, t_2}, r_1)과 같이 나타낸다.

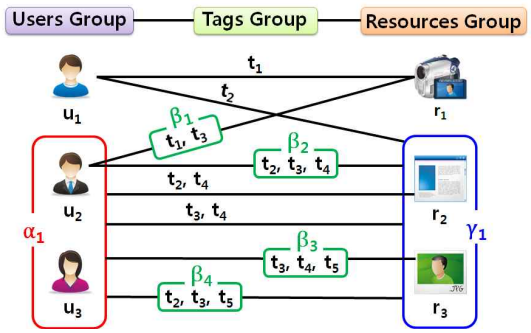


그림 2. 그룹화된 폭소노미의 예
Fig. 2. An example of Grouped Folksonomy

한편, 최근 폭소노미 기반 시스템들[1-5]은 "그룹" 개념을 도입하여 시스템의 기능을 확장하고 있다. 본 연구에서는 폭소노미의 구성요소들(사용자, 태그, 리소스)에 대하여 "그룹" 개념을 도입하여 사용자 그룹, 태그 그룹, 리소스 그룹을 다음과 같이 정의한다.

[정의 2] 사용자 그룹(user group)과 태그 그룹(tag group), 리소스 그룹(resource group)은 각각 사용자들의 집합, 태그들의 집합, 그리고 리소스들의 집합이다. Gu, Gt, Gr은 각각 사용자 그룹들의 집합, 태그 그룹들의 집합, 리소스 그룹들의 집합을 나타낸다($Gu \subseteq 2^U$, $Gt \subseteq 2^T$, $Gr \subseteq 2^R$).

본 논문에서는 정의2를 토대로 그룹화된 폭소노미 모델을 다음과 같이 정의한다.

[정의 3] 그룹화된 폭소노미 GF := (Ŭ, Ť, Ř, Ŷ)는 사용자 및 사용자 그룹의 집합 Ŭ = U∪Gu와 태그 및 태그 그룹의 집합 Ť = T∪Gt, 리소스 및 리소스 그룹의 집합 Ř = R∪Gr, 그리고 태그 할당 관계 Ŷ ⊆ Ŭ×Ť×Ř로 구성된다.

그룹화된 폭소노미는 그림 2와 같이 "그룹(group)"의 개념으로 확장된 폭소노미이다. 그룹화된 폭소노미에서는 사용자들, 태그들, 리소스들이 각각 그룹핑되어 내부적으로 각 구성요소들의 계층구조가 형성되고, 이를 토대로 하여, 그림 3과 같이 다양한 종류의 태그 할당 관계를 표현할 수 있다.

① 사용자는 리소스에 태그를 할당할 수 있다: (u_1, t_1, r_1). 예를 들어, 그림 3에서, 사용자 u_1은 리소스 r_1에 t_1이라는 태그를 할당하였다.

② 사용자는 리소스 그룹에 태그를 할당할 수 있다: (u_1, t_2, γ_1). 단 γ_1 = {r_2, r_3}. 예를 들어, 사용자 u_1은 r_2와 r_3로 구성된 리소스 그룹 γ_1에 태그 t_2를 할당하였다.

③ 사용자는 리소스에 태그 그룹을 할당할 수 있다: (u_2, β_1, r_1). 단, β_1 = {t_1, t_3}. 예를 들어, 사용자 u_2는 리소스 r_1에 t_1과 t_3로 구성된 태그 그룹 β_1을 할당하였다.

④ 사용자는 리소스 그룹에 태그 그룹을 할당할 수 있다:(u_2, β_2, γ_1). 단, β_2={t_2, t_3, t_4}, γ_1={r_2, r_3}. 예를 들어, 사용자 u_2는 리소스 r_2와 r_3로 구성된 리소스 그룹 γ_1에 t_2, t_3, t_4로 구성된 태그 그룹 β_2를 할당하였다.

⑤ 사용자 그룹은 리소스에 태그들을 할당할 수 있다 : (α_1, {t_2, t_4}, r_2). 단, α_1={u_2, u_3}. 예를 들어, u_2와 u_3 로 구성된 사용자 그룹 α_1은 리소스 r_2에 태그 t_2, t_4를 할당하였다.

⑥ 사용자 그룹은 리소스 그룹에 태그들을 할당할 수 있다: (α_1, {t_3, t_4}, γ_1). 단, α_1={u_2, u_3}, γ_1={r_2, r_3}. 예를 들어, u_2와 u_3 로 구성된 사용자 그룹 α_1은 리소스 r_2와 r_3로 구성된 리소스 그룹 γ_1에 태그 t_3, t_4를 할당하였다.

⑦ 사용자 그룹은 리소스에 태그 그룹을 할당할 수 있다: (α_1, β_3, r_3). 단, α_1={u_2, u_3}, β_3={t_3, t_4, t_5}. 예를 들어, u_2와 u_3 로 구성된 사용자 그룹 α_1은 리소스 r_3에 t_3, t_4, t_5로 구성된 태그 그룹 β_3을 할당하였다.

⑧ 사용자 그룹은 리소스 그룹에 태그 그룹을 할당할 수 있다: (α_1, β_4, γ_1). 단, α_1={u_2, u_3}, β_4={t_2, t_3, t_5}, γ_1={r_2, r_3}. 예를 들어, u_2와 u_3 로 구성된 사용자 그룹 α_1은 리소스 r_2와 r_3로 구성된 리소스 그룹 γ_1에 t_2, t_3, t_5로 구성된 태그 그룹 β_4를 할당하였다.

그룹화된 폭소노미의 각 구성요소들에 대해 다음과 같이 사용자, 태그, 리소스들의 집합들을 각각 정의할 수 있다.

[정의 4] 그룹화된 폭소노미 GF := (Ŭ, Ť, Ř, Ŷ)의 임의의 x∈Gu, y∈Gt, z∈Gr에 대하여, 사용자 그룹 x를 구성하는 사용자들의 집합 Users(x), 태그 그룹 y를 구성하는 태그들의 집합 Tags(y), 리소스 그룹 z를 구성하는 리소스들의 집합 Resources(z)는 각각 다음과 같이 정의한다.

- Users(x)={u ∈ Ŭ | u ∈ x ∧ x ∈ Gu},
- Tags(y)={t ∈ Ť | t ∈ y ∧ y ∈ Gt},
- Resources(z) = {r∈ Ř | r ∈ z ∧ z ∈ Gr}.

예를 들어, 그림 2에서, $\alpha_1 \in Gu$일 때, Users(α_1) = {u_2, u_3}이고, $\beta_4 \in Gt$일 때, Tags(β_4) = {t_2, t_3, t_5}이다.

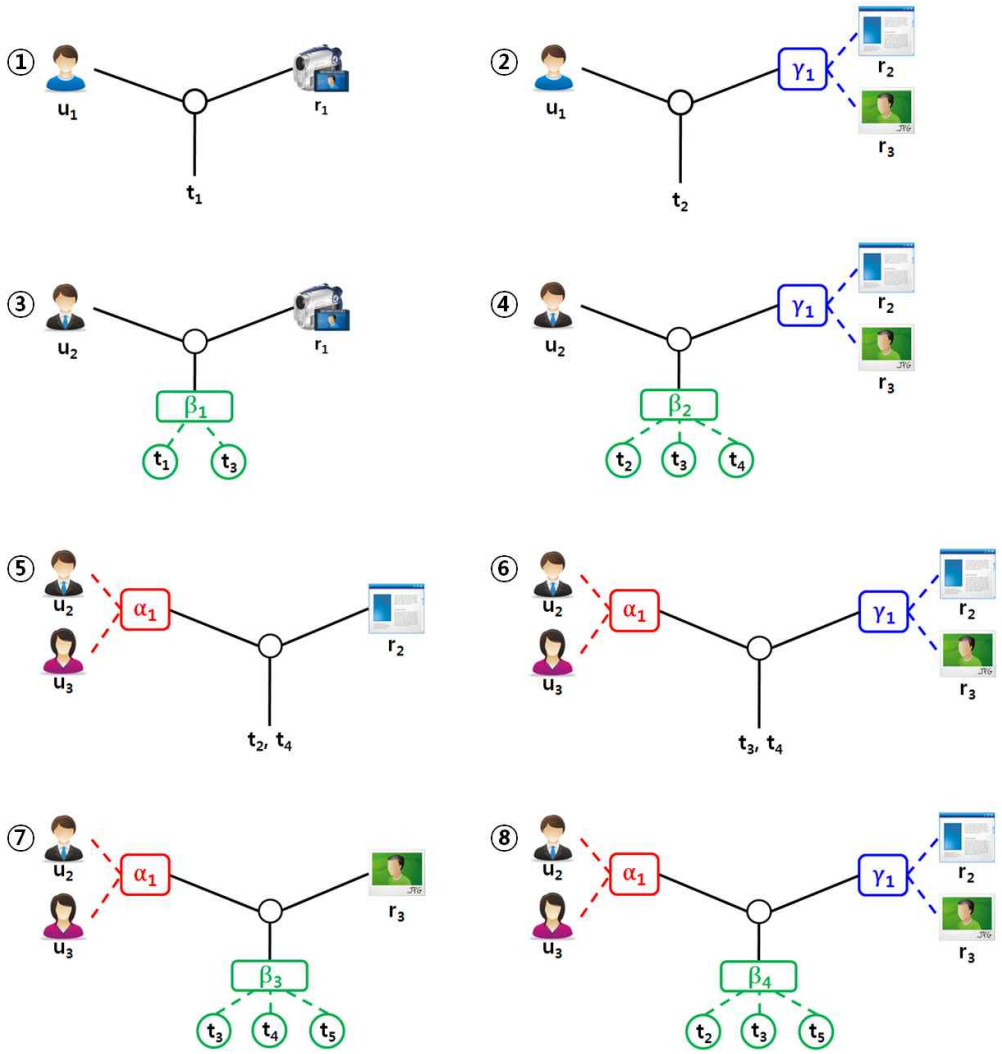


그림 3. 그룹화된 폭소노미의 3부 그래프(Tripartite graph)
Fig. 3. Tripartite graph of grouped folksonomy

또한, $\gamma_1 \in Gr$일 때, $Resources(\gamma_1) = \{r_2, r_3\}$이다.

그룹화된 폭소노미의 구성요소와 관련된 사용자(또는 사용자 그룹), 태그(또는 태그 그룹), 리소스(또는 리소스 그룹) 집합을 다음과 같이 정의한다.

[정의 5] 그룹화된 폭소노미 GF := (Ŭ, Ť, Ř, Ŷ)의 임의의 $u \in Ŭ$(즉, $u \in U \cup Gu$)에 대하여, 사용자(또는 사용자 그룹) u가 사용한 태그들의 집합 $T(u) = \{t \in Ť \mid \exists\, r \in Ř : (u, t, r) \in Ŷ\}$이고, 사용자 및 사용자 그룹 u가 태깅한 리소스들의 집합 $R(u) = \{r \in Ř \mid \exists\, t \in Ť : (u, t, r) \in Ŷ\}$이다.

[정의 6] 그룹화된 폭소노미 GF := (Ŭ, Ť, Ř, Ŷ)의 임의의 $t \in Ť$(즉, $t \in T \cup Gt$,)에 대하여, 태그(또는 태그 그룹) t를 사용한 사용자 그룹 $U(t) = \{u \in Ŭ \mid \exists\, r \in Ř : (u, t, r) \in Ŷ\}$이고, 태그 및 태그 그룹 t가 태깅한 리소스 집합 $R(t) = \{r \in Ř \mid \exists\, u \in Ŭ : (u, t, r) \in Ŷ\}$이다.

[정의 7] 그룹화된 폭소노미 GF := (Ŭ, Ť, Ř, Ŷ)의 임의의 $r \in Ř$(즉, $r \in R \cup Gr$)에 대하여, 리소스(또는 리소스 그룹) r을 태깅한 사용자 집합 $U(r) = \{u \in Ŭ \mid \exists t \in Ť : (u, t, r) \in Ŷ\}$이고, 리소스 및 리소스 그룹 r에 할당된 태그 집합 $T(r) = \{t \in Ť \mid \exists\, u \in Ŭ : (u, t, r) \in Ŷ\}$이다.

그림 2에서, 사용자 그룹 α1에 대하여 $T(\alpha_1) = \{t_2, t_3, t_4, t_5, \beta_3, \beta_4\}$이고, $R(\alpha_1) = \{r_2, r_3, \gamma_1\}$이다. 또한, 태그 그룹 β_3에 대하여 $U(\beta_3) = \{\alpha_1\}$이고, $R(\beta_3) = \{r_3\}$이다.

리소스 그룹 γ_1에 대하여 $U(\gamma_1)=\{u_1, \alpha_1\}$이고, $T(\gamma_1)=\{t_2, t_3, t_4, t_5, \beta_2, \beta_3, \beta_4\}$이다.

[정의 8] 그룹화된 폭소노미 GF := ($\breve{U}$, $\breve{T}$, $\breve{R}$, $\hat{Y}$)의 임의의 $u \in \breve{U}$, $t \in \breve{T}$, $r \in \breve{R}$에 대하여, u가 특정 리소스(또는 리소스 그룹) r에 태깅한 태그들의 집합 T(u, r)와 u가 특정 태그(또는 태그 그룹)를 사용하여 태깅한 리소스들의 집합 R(u, t), 특정 태그(또는 태그 그룹) t를 특정 리소스(또는 리소스 그룹) r에 태깅한 사용자들의 집합 U(t, r)를 다음과 같이 정의한다.

- $T(u, r) = \{t \in \breve{T} \mid (u, t, r) \in \hat{Y}\}$,
- $R(u, t) = \{r \in \breve{R} \mid (u, t, r) \in \hat{Y}\}$,
- $U(t, r) = \{u \in \breve{U} \mid (u, t, r) \in \hat{Y}\}$.

그림 2에서, 사용자 그룹 α_1이 리소스 r_2에 태깅한 태그 $T(\alpha_1, r_2) = \{t_2, t_4\}$이고, 사용자 그룹 α_1이 태그 그룹 β_4를 사용하여 태깅한 리소스 $R(\alpha_1, \beta_4) = \{\gamma_1\}$이다. 또한, 태그 그룹 β_2를 사용하여 리소스 그룹 γ_1을 태깅한 사용자 $U(\beta_2, \gamma_1)=\{u_2\}$이다.

IV. 그룹화된 폭소노미 구축을 위한 기본적인 연산들

본 장에서는, 기존의 폭소노미로부터 그룹화된 폭소노미를 구축하기 위하여, 사용자들, 태그들, 리소스들을 그룹핑하고 체계화하기 위한 기본적인 연산들을 소개한다.

표 2. 그룹화된 폭소노미를 체계화하기 위한 기본 연산들
Table 2. Fundamental operations for organization of grouped folksonomy

Group Aggregation	GA(GF, X, x1, x2)
Group Composition	GC(GF, X, x1, x2)
Group Intersection	GI(GF, X, x1, x2)
Group Difference	GD(GF, X, x1, x2)

3장에서 소개한 제반 정의와 집합이론에서의 합집합, 교집합, 차집합을 기반으로 하여 표2와 같이 4개의 기본적인 연산을 정의하였다. 모든 연산을 수행하기 위해서는 4개의 매개변수가 필요하다. GF는 그룹화된 폭소노미를 의미하며, $X \in \{U, T, R\}$는 사용자에 관한 연산인지, 태그에 관한 연산인지, 리소스에 관한 연산이지를 나타낸다. 각 연산들은 X에 의해 x_1과 x_2의 타입이 결정되고, x_1과 x_2는 각각 하나의 요소가 될 수도 있고 그룹이 될 수도 있다. 예를 들어, X가 U인 경우, $x_1, x_2 \in \breve{U}$이다.

4.1. Group Aggregation

Group Aggregation은 사용자 그룹, 태그 그룹, 리소스 그룹의 내부에 존재하는 계층구조를 그대로 유지하면서 집약하여 새로운 그룹을 생성하는 연산이다.

```
GA(GF, X, x1, x2, GN)
//입력 : GF=(Ŭ, Ť, Ř, Ŷ), X∈{U, T, R}, x1,x2∈Ŭ ∨ x1, x2∈Ť ∨ x1,x2∈Ř, GN은 새로 생성될 그룹의 이름이다.
//출력 : GFn = (Ŭn, Ťn, Řn, Ŷn).
1 begin
2   if (X ∈ U ∧ (x1∈ Ŭ∧x2∈ Ŭ)) :
3       Ŭn = Ŭ ∪ {GN};
4       GN = GN ∪ {x1, x2};
5       Ťn = Ť;
6       Řn = Ř;
7       Ŷn = TA(x1) ∪ TA(x2);
8   else if (X ∈ T ∧ (x1∈ Ť∧x2∈ Ť)) :
9       Ťn = Ť ∪ {GN};
10       GN = GN ∪ {x1, x2};
11       Ŭn = Ŭ;
12       Řn = Ř;
13       Ŷn = TA(x1) ∪ TA(x2);
14   else if (X ∈ R ∧ (x1∈ Ř∧x2∈ Ř)) :
15       Řn = Ř ∪ {GN};
16       GN = GN ∪ {x1, x2};
17       Ŭn = Ŭ;
18       Ťn = Ť;
19       Ŷn = TA(x1) ∪ TA(x2);
20   end if
21 end

function TA(x)
1 begin
2   if(x ∈ Ŭ)
3      for each r ∈ R(x) do
4         Ŷn = Ŷ ∪ {(GN, T(x, r), r)};
5      end for
6   else if(x ∈ Ť)
7      for each u ∈ U(x) do
8         Ŷn = Ŷ ∪ {(u, GN, R(u, x))};
9      end for
10   else if(x ∈ Ř)
11      for each t ∈ T(x) do
12         Ŷn = Ŷ ∪ {(U(t, x), t, GN)};
13      end for
```

```
14    end if
15 end
```

그림 4. Group Aggregation 알고리즘
Fig. 4. Group Aggregation Algorithm

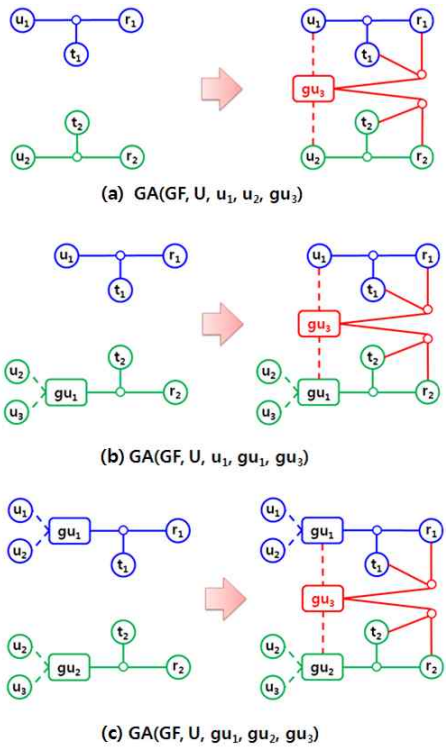


그림 5. 사용자 Group Aggregation의 예
Fig. 5. An example of User Group Aggregation

그림 4는 Group Aggregation을 위한 알고리즘으로, 사용자에 대한 Group Aggregation(1~7번째 줄)과 태그에 대한 Group Aggregation(8~13번째 줄), 그리고 리소스에 대한 Group Aggregation(13~21번째 줄)으로 구성된다. Group Aggregation은 기존의 태그 할당 관계는 그대로 유지하면서, function TA(x)를 사용하여 Group Aggregation 연산에 의해 새로 생성된 그룹에 대한 태그할당관계를 형성한다. 그림5는 사용자를 대상으로 Group Aggregation 연산을 수행한 예이다. 그림 5의 (a)는 사용자 u1과 u2에 대하여 Group Aggregation을 수행하는 연산으로, u1과 u2로 구성된 새로운 사용자 그룹 gu3을 생성한다. 그림 5의 (b)는 사용자 u1과 u2와 u3로 구성된 사용자 그룹 gu1에 대하여 Group Aggregation을 수행하는 연산으로, gu1의 내부 계층구조는 그대로 유지하면서, u1과 gu1로 구성된 새로운 사용자 그룹 gu3을 생성한다. 그림 5의 (c)는 두 개의 사용자 그룹 gu1과 gu2를 대상으로 Group Aggregation을 수행하는 연산으로, gu1과 gu2의 내부 계층구조는 그대로 유지하면서, gu1과 gu2로 구성된 새로운 사용자 그룹 gu3을 생성한다. (a), (b), (c) 모두 기존의 태그 할당 관계는 그대로 유지한다. 같은 방법으로 태그 또는 리소스를 대상으로 각각 Group Aggregation 연산을 수행할 수 있다.

Group Aggregation 연산에 의해 새로 생성된 그룹 내부에는 여러 개의 그룹으로 구성된 그룹이 존재할 수 있으므로, 복잡한 계층구조가 형성될 수 있다.

4.2 Group Composition

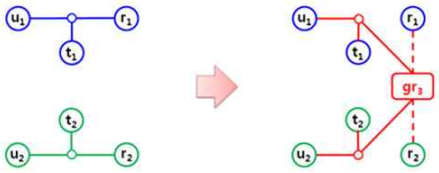
Group Composition은 사용자 그룹, 태그 그룹, 리소스 그룹의 내부에 존재하는 계층구조를 평탄화한 후 합성하여 새로운 그룹을 생성하는 연산이다.

```
GC(GF, X, x1, x2, GN)
//입력 : GF=(Ŭ, Ť, Ř, Ŷ), X∈{U, T, R}, x1,x2∈Ŭ ∨ x1,x2∈Ť ∨ x1,x2∈Ř, GN은 새로 생성될 그룹의 이름이다.
//출력 : GFn = (Ŭn, Ťn, Řn, Ŷn).
1 begin
2  switch(X){
3    case U :
4      Ŭn = Ŭ ∪ {GN};
5      if(x1∈U ∧ x2∈U)
6        GN = GN ∪ {x1, x2};
7      else if(x1∈U ∧ x2∈Gu)
8        GN = {x1} ∪ Users(x2);
9      else if(x1∈Gu ∧ x2∈Gu)
10       GN = Users(x1) ∪ Users(x2);
11     end if
12     for each u∈ GN do
13       GN = GN∪Composition(u);
14     end for
15     Ťn = Ť;
16     Řn = Ř;
17     Ŷn = TA(x1) ∪ TA(x2);
18     Ŭn = Ŭn - {x1, x2};
19   case T :
20     Ťn = Ť ∪ {GN};
21     if(x1∈T ∧ x2∈T)
22       GN = GN ∪ {x1, x2};
23     else if(x1∈T ∧ x2∈Gt)
```

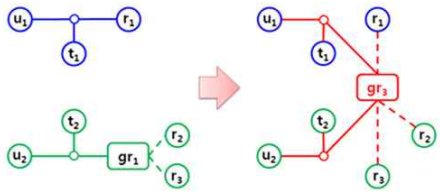
```
24          GN = {x1}∪Tags(x2);
25       else if(x1∈Gt ∧ x2∈Gt)
26          GN = Tags(x1) ∪ Tags(x2);
27       end if
28       for each t∈ GN do
29          GN = GN∪Composition(t);
30       end for
31       Ůn = Ů;
32       Řn = Ř;
33       Ŷn = TA(x1) ∪ TA(x2);
34       Ťn = Ťn - {x1, x2};
35    case R :
36       Řn = Ř ∪ {GN};
37       if(x1∈R ∧ x2∈R)
38          GN = GN ∪ {x1, x2};
39       else if(x1∈R ∧ x2∈Gr)
40          GN = {x1}∪Resources(x2);
41       else if(x1∈Gr ∧ x2∈Gr)
42          GN = Resources(x1) ∪ Resources(x2);
43       end if
44       for each r∈ GN do
45          GN = GN∪Composition(r);
46       end for
47       Ůn = Ů;
48       Ťn = Ť;
49       Ŷn = TA(x1) ∪ TA(x2);
50       Řn = Řn - {x1, x2};
51  }
52 end

function Composition(x)
1 begin
2    if(x ∈ U)
3       GN = GN ∪ {x};
4       return GN;
5    else if(x ∈ T)
6       GN = GN ∪ {x};
7       return GN;
8    else if(x ∈ R)
9       GN = GN ∪ {x};
10      return GN;
11   else if(x ∈ Gu)
12      for each y ∈ Users(x)
13         GN = GN ∪ Composition(y);
14      end for
15   else if(x ∈ Gt)
16      for each y ∈ Tags(x)
17         GN = GN ∪ Composition(y);
18      end for
19   else if(x ∈ Gr)
20      for each y ∈ Resources(x)
21         GN = GN ∪ Composition(y);
22      end for
23   end if
24 end

function TA(x)
1 begin
2    if(x ∈ Ů)
3       for each r ∈ R(x) do
4          Ŷn = Ŷ ∪ {(GN, T(x, r), r)};
5          Ŷn = Ŷn - {(x, T(x, r), r)};
6       end for
7    else if(x ∈ Ť)
8       for each u ∈ U(x) do
9          Ŷn = Ŷ ∪ {(u, GN, R(u, x))};
10         Ŷn = Ŷn - {(u, x, R(u, x))};
11      end for
12   else if(x ∈ Ř)
13      for each t ∈ T(x) do
14         Ŷn = Ŷ ∪ {(U(t, x), t, GN)};
15         Ŷn = Ŷn - {(U(t, x), t, x)};
16      end for
17   end if
18 end
```

그림 6. Group Composition 알고리즘
Fig. 6. Group Composition Algorithm

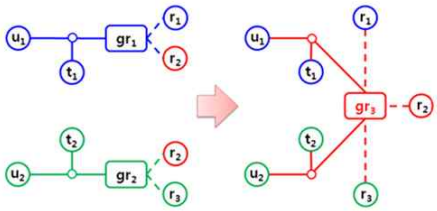
그림 6은 Group Composition을 위한 알고리즘으로, switch문의 X의 타입에 따라 case U 또는 case T 또는 case R이 수행된다. case U는 사용자에 대한 Group Composition 연산을 수행하는 부분(3~18번째 줄)이고 , case T는 태그에 대한 Group Composition 연산을 수행하는 부분(19~34번째 줄)이며, case R은 리소스에 대한 Group Composition 연산을 수행하는 부분(35~50번째 줄)이다. 각 case 내부에서는 두 개의 단일 요소를 대상으로 연산을 수행할 경우와 한 개의 단일 요소와 한 개의 그룹을 대상으로 연산을 수행하는 경우 그리고 두 개의 그룹을 대상으로 연산을 수행하는 경우로 나뉘어진다. Group Composition 연산에 의해 새로 생성된 그룹 안에는 어떠한 그룹도 존재할 수 없으므로 function Composition(x)을 사용하여 그룹 안의 요소들만 추출한다. function TA(x)는 기존의 태그 할당 관계를 토대로 연산에 의해 새로 생성된 그룹의 태그 할당 관계를 형성한 후 전체 태그 할당 관계 Ŷn에서 기존의 태그 할당 관계를 삭제한다. 그림7은 리소스를 대상으로 Group Composition 연산을 수행한 예이다. 그림 7의 (a)는 리소스 r1과 r2에 대하여 Group Composition을 수행하는 연산으로, r1과 r2로 구성된 새로운 리소스 그룹 gr3을 생성한다. 그림 7의 (b)는 리소스 r1과 r2와 r3로 구성된



(a) GC(GF, R, r_1, r_2, gr_3)



(b) GC(GF, R, r_1, gr_1, gr_3)



(c) GC(GF, R, gr_1, gr_2, gr_3)

그림 7. 리소스 Group Composition의 예
Fig. 7. An example of Resource Group Composition

태그 그룹 gr1에 대하여 Group Composition을 수행하는 연산으로, gr1의 내부 계층 구조는 유지하지 않고, gr1의 구성요소 r2, r3과 r1로 구성된 새로운 리소스 그룹 gr3을 생성한다. 그림 7의 (c)는 두 개의 리소스 그룹 gr1과 gr2를 대상으로 Group Composition을 수행하는 연산으로, gr1과 gr2의 내부 계층 구조는 유지하지 않고, gr1과 gr2의 구성요소 r1, r2, r3로 구성된 새로운 리소스 그룹 gr3을 생성한다.

Group Composition 연산에 의해 새로 생성된 그룹 내부에는 어떠한 그룹도 존재하지 않으며, 연산 시 입력으로 들어온 그룹 및 태그 할당 관계는 연산 후 모두 삭제된다.

4.3 Group Intersection

Group Intersection은 사용자 그룹, 태그 그룹, 리소스 그룹의 내부에 존재하는 공통된 원소를 추출하여 새로운 그룹을 생성하는 연산이다.

그림 8은 Group Intersection을 위한 알고리즘으로, Group Composition 연산에서와 같이 switch문의 X의 타입에 따라 사용자에 대한 Group Intersection 연산을 수행하는 부분(3~18번째 줄)과 태그에 대한 Group Intersection 연

```
GI(GF, X, x1, x2, GN)
//입력 : GF=(Ŭ, Ť, Ř, Ŷ), X∈{U, T, R}, x1,x2∈Ŭ ∨ x1,x2∈Ť ∨ x1,x2∈Ř, GN은 새로 생성될 그룹의 이름이다.
//출력 : GFn = (Ŭn, Ťn, Řn, Ŷn).
1  begin
2   switch(X){
3     case U :
4       Ŭn = Ŭ ∪ {GN};
5       if(x1∈U ∧ x2∈U)
6           exit;
7         else if((x1∈U∧x2∈Gu) ∧ {x1}∩Users(x2)≠∅)
8           GN = GN ∪ {x1};
9           Ŭn = Ŭ ∪ {GN};
10          Ťn = Ť;
11          Řn = Ř;
12          Ŷn = TA(x1) ∪ TA(x2);
13    else if((x1∈Gu∧x2∈Gu)∧Users(x1)∩Users(x2)≠∅)
14          GN = Users(x1) ∩ Users(x2);
15          Ťn = Ť;
16          Řn = Ř;
17          Ŷn = TA(x1) ∪ TA(x2);
18      end if
19   case T :
20      Ťn = Ť ∪ {GN};
21      if(x1∈T ∧ x2∈T)
22          exit;
23      else if((x1∈T∧x2∈Gt) ∧ {x1}∩Tags(x2)≠∅)
24          GN = GN ∪ {x1};
25          Ťn = Ť ∪ {GN};
26          Ŭn = Ŭ;
27          Řn = Ř;
28          Ŷn = TA(x1) ∪ TA(x2);
29       else if((x1∈Gt∧x2∈Gt)∧Tags(x1)∩Tags(x2)≠∅)
30          GN = Tags(x1) ∩ Tags(x2);
31          Ŭn = Ŭ;
32          Řn = Ř;
33          Ŷn = TA(x1) ∪ TA(x2);
34      end if
35    case R :
```

```
36      Řn = Ř ∪ {GN};
37      if(x1∈R ∧ x2∈R)
38          exit;
39      else if((x1∈R∧x2∈Gr)∧{x1}∩Resources(x2)≠
∅)
40          GN = GN ∪ {x1};
41          Řn = Ř ∪ {GN};
42          Ůn = Ů;
43          Ťn = Ť;
44          Ŷn = TA(x1) ∪ TA(x2);
45      else if((x1∈Gr∧x2∈Gr) ∧ Resources(x1)∩
Resources(x2)≠∅)
46          GN = Resources(x1) ∩ Resources(x2);
47          Ůn = Ů;
48          Ťn = Ť;
49          Ŷn = TA(x1) ∪ TA(x2);
50      end if
51 }
52 end

function TA(x)
1  begin
2    if(x ∈ Ů)
3      for each r ∈ R(x) do
4        Ŷn = Ŷ ∪ {(GN, T(x, r), r)};
5      end for
6    else if(x ∈ Ť)
7      for each u ∈ U(x) do
8        Ŷn = Ŷ ∪ {(u, GN, R(u, x))};
9      end for
10   else if(x ∈ Ř)
11     for each t ∈ T(x) do
12       Ŷn = Ŷ ∪ {(U(t, x), t, GN)};
13     end for
14   end if
15 end
```

그림 8. Group Intersection 알고리즘
Fig. 8. Group Intersection Algorithm

산을 수행하는 부분(19~34번째 줄), 리소스에 대한 Group Intersection 연산을 수행하는 부분(35~50번째 줄)으로 구성된다. Group Intersection은 두 개의 단일 요소를 대상으로는 연산은 수행하지 않으며, 한 개의 단일 요소와 한 개의 그룹을 대상으로 연산을 수행하거나 두 개의 그룹을 대상으로 연산을 수행한다. Group Intersection은 기존의 태그 할당 관계는 그대로 유지하면서 이를 토대로 function TA(x)를 사용하여 Group Intersection 연산에 의해 새로 생성된 그룹의 태그 할당 관계를 형성한다. 그림9는 태그를 대상으로 Group

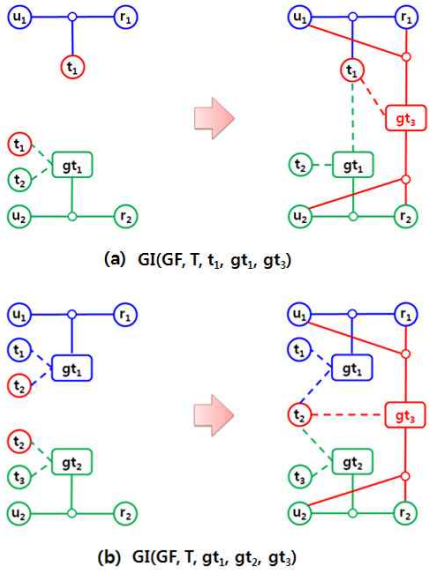


그림 9. 태그 Group Intersection의 예
Fig. 9. An example of Tag Group Intersection

Intersection 연산을 수행한 예이다. 그림9의 (a)는 태그 t1과 t1과 t2로 구성된 태그 그룹 gt1에 대하여 Group Intersection을 수행하는 연산으로, gt1의 내부에 존재하는 공통된 원소 t1을 추출하여 새로운 태그 그룹 gt3을 생성한다. 그림9의 (b)는 태그 t1과 t2로 구성된 태그 그룹 gt1과 태그 t2와 t3로 구성된 태그 그룹 gt2를 대상으로 Group Intersection을 수행하는 연산으로, gt1과 gt2의 내부에 존재하는 공통된 원소 t2를 추출하여 새로운 태그 그룹 gt3을 생성한다. (a), (b) 모두 기존의 태그 할당 관계를 토대로 새로 생성된 태그 그룹 gt3의 태그 할당 관계를 형성한다.

Group Intersection은 공통 원소를 추출하여 새로운 그룹을 생성하는 연산으로서, 공통 관심사를 갖는 커뮤니티를 추출하고자 할 때 유용하게 사용될 수 있다.

4.4 Group Difference

Group Difference는 사용자 그룹, 태그 그룹, 리소스 그룹의 내부에 존재하는 공통된 원소를 삭제한 후 배타적 특성을 갖는 원소를 추출하여 새로운 그룹을 생성하는 연산이다.

```
GD(GF, X, x1, x2, GN)
//입력 : GF=(Ů, Ť, Ř, Ŷ), X∈{U, T, R}, x1,x2∈Ů ∨
x1,x2∈Ť ∨x1,x2∈Ř, GN은 새로 생성될 그룹의 이름이다.
//출력 : GFn = (Ůn, Ťn, Řn, Ŷn).
```

```
1 begin
2  switch(X){
3      case U :
4        Ůn = Ů ∪ {GN};
5        if((x1∈U ∧ x2∈U) ∧ (x1≠x2))
6           exit;
7        else if((x1∈Gu∧x2∈U) ∧ Users(x1)≠{x2})
8           GN = Users(x1) - {x2};
9              else if((x1∈Gu∧x2∈Gu)∧Users(x1)≠
Users(x2))
10          GN = Users(x1) - Users(x2);
11       end if
12       Ťn = Ť;
13       Řn = Ř;
14       Ŷn = TA(x1);
15    case T :
16       Ťn = Ť ∪ {GN};
17       if((x1∈T ∧ x2∈T) ∧ (x1≠x2))
18          exit;
19       else if((x1∈Gt ∧ x2∈T) ∧ Tags(x1)≠{x2})
20          GN = Tags(x1) - {x2};
21           else if((x1∈Gt∧x2∈Gt) ∧ Tags(x1)≠
Tags(x2))
22          GN = Tags(x1) - Tags(x2);
23       end if
24       Ůn = Ů;
25       Řn = Ř;
26       Ŷn = TA(x1);
27    case R :
28       Řn = Ř ∪ {GN};
29       if((x1∈R ∧ x2∈R) ∧ (x1≠x2))
30          exit;
31        else if((x1∈Gr∧x2∈R) ∧ Resources(x1)≠
{x2})
32          GN = Resources(x1) - {x2};
33       else if((x1∈Gr∧x2∈Gr) ∧ Resources(x1)≠
Resources(x2))
34          GN = Resources(x1) - Resources(x2);
35       end if
36       Ůn = Ů;
37       Ťn = Ť;
38       Ŷn = TA(x1);
39 }
40 end
```

```
function TA(x)
1 begin
2    if(x ∈ Ů)
3       for each r ∈ R(x) do
4          Ŷn = Ŷ ∪ {(GN, T(x, r), r)};
5       end for
6    else if(x ∈ Ť)
7       for each u ∈ U(x) do
8          Ŷn = Ŷ ∪ {(u, GN, R(u, x))};
9       end for
10   else if(x ∈ Ř)
11      for each t ∈ T(x) do
12         Ŷn = Ŷ ∪ {(U(t, x), t, GN)};
13      end for
14   end if
15 end
```

그림 10. Group Difference 알고리즘
Fig. 10. Group Difference Algorithm

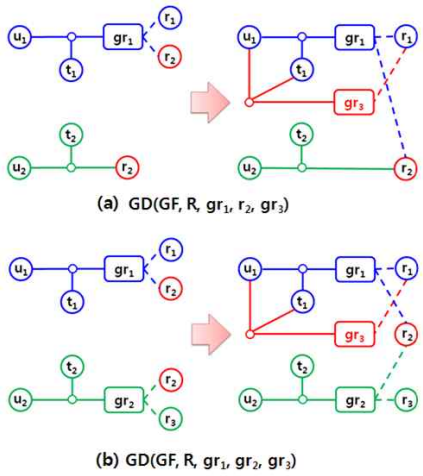


그림 11. 리소스 Group Difference의 예
Fig. 11. An example of Resource Group Difference

그림 10은 Group Difference를 위한 알고리즘으로, 사용자에 대한 Group Difference(3~14번째 줄)와 태그에 대한 Group Difference(15~26번째 줄), 리소스에 대한 Group Difference(27~38번째 줄)로 구성된다. Group Difference는 기존의 태그 할당 관계는 그대로 유지하면서, function TA(x)를 사용하여 연산에 의해 새로 생성된 그룹에 대한 태그할당관계를 형성한다. 그림11은 리소스에 대한 Group Difference 연산을 수행한 예이다. 그림11의 (a)는 리소스 r1과 r2로 구성된 리소스 그룹gr1과 리소스r2에 대하여 Group Difference를 수행하는 연산으로, gr1에서 공통

원소인 r2를 삭제한 후 나머지 원소 r1로 구성된 새로운 리소스 그룹 gr3을 생성한다. 그림11의 (b)는 리소스 r1과 r2로 구성된 리소스 그룹 gr1과 리소스 r2와 r3로 구성된 리소스 그룹 gr2를 대상으로 Group Intersection을 수행하는 연산으로, gr1에서 gr1과 gr2의 공통 원소인 r2를 삭제한 후 나머지 원소 r1로 구성된 새로운 리소스 그룹 gr3을 생성한다. (a), (b) 모두 기존의 태그 할당 관계를 토대로 새로 생성된 리소스 그룹 gr3의 태그 할당 관계를 형성한다.

V. 지원도구의 개발 및 사용사례

본 장에서는, 본 연구에서 제안한 그룹화된 폭소노미 모델을 기반으로 기본적인 연산을 수행하기 위해 개발한 지원도구(GFO : Grouped Folksonomy Operator)와 실제 데이터를 대상으로 지원도구를 사용하여 각 요소들을 그룹핑하는 사용사례를 소개한다.

5.1 지원도구의 개발

본 절에서는, 폭소노미의 각 요소들을 그룹핑하고 체계화하기 위해 제3장의 제반 정의들과 제4장의 알고리즘을 토대로, 그룹화된 폭소노미를 구축하기위해 기본연산을 지원하는 도구(GFO : Grouped Folksonomy Operator)를 개발하였다. GFO의 전체적인 아키텍처는 그림 12와 같이 Bibsonomy 사이트에서 폭소노미 정보를 가져와서 데이터베이스(Database)에 저장하기 위한 "Bibsonomy Crawler"와 사용자가 Bibsonomy 데이터를 대상으로 기본연산들을 수행하기 위해 제공되는 "UI Modules", 실제 연산을 수행하는 "Processing Modules"로 구성되어 있다.

Bibsonomy Crawler는 Bibsonomy에서 제공되는 API를 사용하여 Bibsonomy에 축적되어있는 폭소노미 정보를 GFO 내부의 데이터베이스에 저장하는 모듈이다.

UI Modules는 Search 모듈, Tag Assignments View 모듈, Results View 모듈로 구성되어 있다. Search 모듈은 그림13의 (b)와 같이 사용자가 선택한 폭소노미 요소 중 사용자가 입력한 키워드에 관한 정보를 데이터베이스에서 검색하는 기능을 제공한다.

Tag Assignments View 모듈은 그림13의 (c)와 같이 Search의 결과를 테이블 형태로 보여주고 연산에 필요한 정보를 입력하는 기능을 제공한다. Results View 모듈은 그림13의 (d)와 (e)와 같이 새로 생성하고자 하는 그룹의 이름과 새로 생성될 그룹의 요소를 입력하여, 연산이 수행된 결과를 테이블 형태로 보여주며, 그룹화된 정보를 데이터베이스에 저장하는 기능을 제공한다.

Processing Modules은 그림13의 (a)와 같이 Group Aggregation 모듈, Group Composition 모듈, Group Intersection 모듈, Group Difference 모듈로 구성되어 있다. Group Aggregation 모듈은 Tag Assignments View 모듈에서 입력된 정보를 토대로, 입력된 그룹의 내부 계층구조를 그대로 유지하면서 집약하여 새로운 그룹을 생성한다. Group Composition 모듈은 Tag Assignments View 모듈에서 입력된 정보를 토대로, 입력된 그룹의 내부 계층구조를 평탄화한 후 합성하여 새로운 그룹을 생성한다. Group Intersection 모듈은 입력된 그룹의 내부에 존재하는 공통된

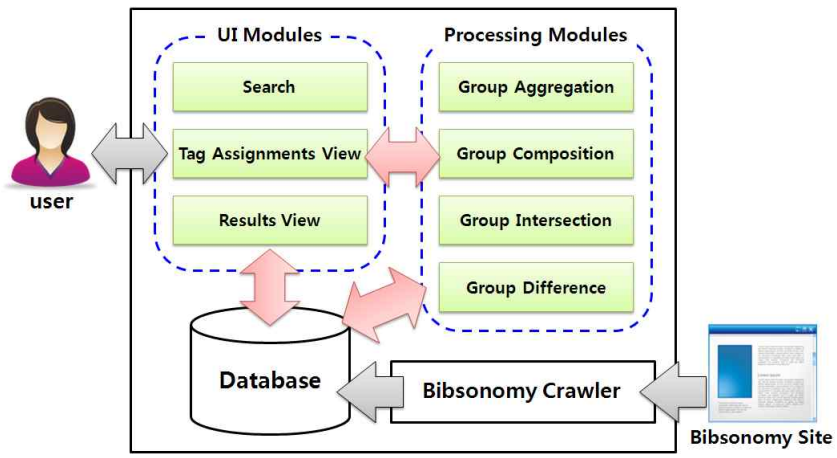


그림 12. GFO(Grouped Folksonomy Operator)의 아키텍처
Fig. 12. Architecture of Grouped Folksonomy Operator

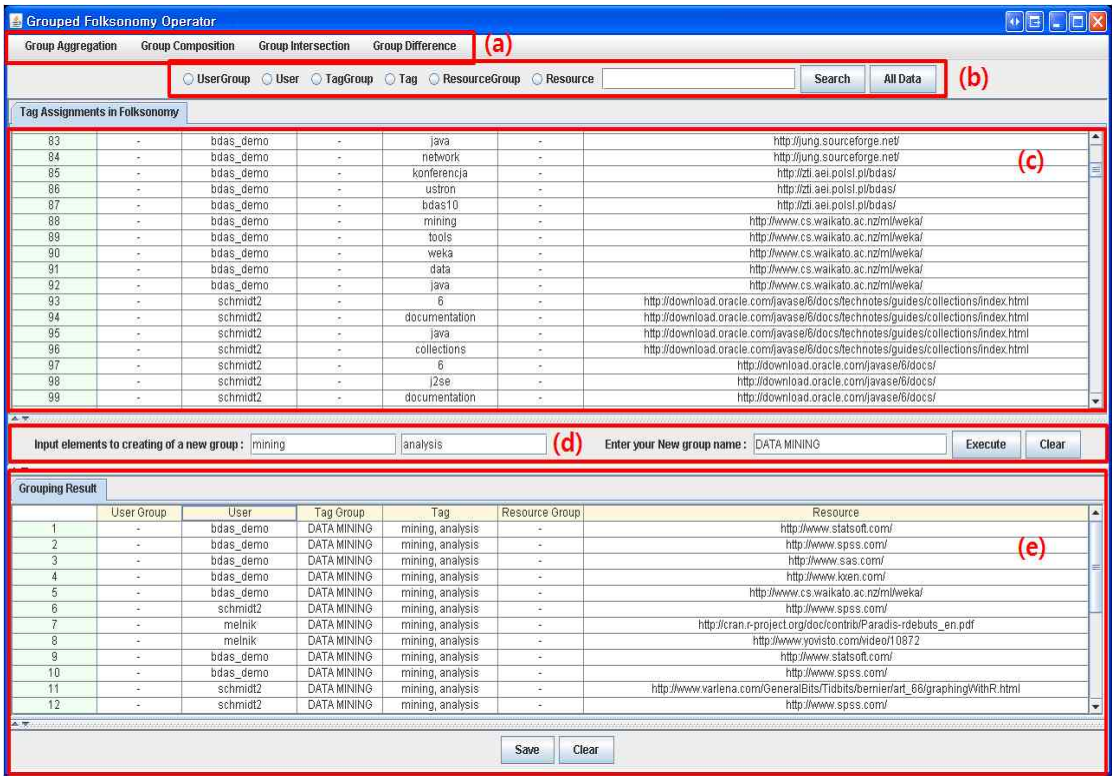


그림 13. Grouped Folksonomy Operator의 실행 화면
Fig. 13. A screenshot of Grouped Folksonomy Operator

원소를 추출하여 새로운 그룹을 생성한다. Group Difference 모듈은 입력된 그룹의 내부에 존재하는 공통된 원소를 삭제한 후 배타적 특성을 갖는 원소를 추출하여 새로운 그룹을 생성한다.

표3은 본 논문에서 개발한 시스템과 기존 시스템들과의 기능 비교를 나타내고 있다. 기존의 폭소노미 기반 시스템들은 표3과 같이 폭소노미의 각 요소들을 그룹핑하기 위한 일부 기능만을 제공하고 있다. 본 연구에서 개발한 GFO는 그룹화된 폭소노미 모델을 기반으로 사용자, 태그, 리소스 각각을 그룹핑하기 위한 모든 기능을 제공하고 있으며, 그룹화된 정보를 토대로 폭소노미 데이터를 체계화하고 이를 기반으로 정보를 검색할 수 있다. 현재는 Bibsonomy로부터 폭소노미 데이터를 가져오기 위한 Bibsonomy Crawler만 개발되어 있지만 다양한 폭소노미 기반 시스템에 맞게 Crawler들을 개발한다면 기존의 폭소노미 기반 시스템에 축적되어 있는 폭소노미 데이터를 대상으로 GFO를 사용하여 그룹화된 폭소노미를 구축할 수 있다

표 3. 본 연구결과와 기존 시스템과의 기능비교
Fig. 3. Function comparison among our tool and existing systems

	user grouping	tag grouping	resource grouping
del.icio.us[1]	O	O	X
Flickr[2]	O	X	O
YouTube[3]	O	X	O
Bibsonomy[4]	O	O	X
GroupMe[5]	O	X	O
GFO(our tool)	O	O	O

5.2 사용사례

본 절에서는, bibsonomy에서 “java” 태그 또는 “analysis” 태그를 사용한 사용자 중에서 임의의 100명 사용자의 태깅 정보(24399건)를 추출하여, 추출된 태그에 대하여 기본적인 연산을 수행하는 사례를 소개한다.

(1) Tag Group Composition

sql 또는 oracle과 관련 있는 리소스들을 하나의 그룹으로 그룹핑 하고자 할 때, 그림 14의 (a)와 같이 “sql” 태그와

“oracle” 태그를 사용한 사용자의 태깅 정보를 검색한 후 Group Composition 연산을 수행하여 그림 14의 (b)와 같이 새로운 태그 그룹 “DB”을 생성할 수 있다. Group Composition은 그룹을 평탄화한 후 합성하여 새로운 그룹을 생성하기 때문에 Group Composition에 의해 생성된 새로운 그룹 내부에는 그룹이 존재 할 수 없다.

⑵ Tag Group Aggregation

데이터베이스 관련 리소스와 톰캣(tomcat)관련 리소스를 하나의 그룹으로 만들고자 할 때, Group Aggregation을 사용할 수 있다. 그림 15의 (a)와 같이 “tomcat” 태그를 사용한 사용자의 태깅 정보와 그림 14의 (b)와 같이 Group Composition에 의해 생성된 태그 그룹 “DB”를 대상으로 Group Aggregation을 수행함으로써, 그림 15의 (b)와 같이 새로운 태그 그룹 “SERVER”를 생성할 수 있다. Group Aggregation은 기존에 그룹핑 된 태그 그룹 “DB”의 내부 계층 구조는 그대로 유지하면서 새로운 그룹 “SERVER”를 생성하기 때문에 Group Aggregation에 의해 생성된 그룹 내부에는 다른 그룹이 존재 할 수 있으며, 자동적으로 계층구조가 형성이 된다.

⑶ Tag Group Intersection

기존에 존재하는 태그 그룹들 사이에 공통적으로 사용된 태그들만 그룹핑 하고자 할 때, Group Intersection을 사용하여 새로운 그룹을 생성 할 수 있다. 그림 16의 (a)와 같이 “free”, “sourceforge”, “opensource” 태그들을 Group Composition하여 태그 그룹 “OPENSOURCE”를 생성하고, 같은 방법으로 그림 16의 (b)와 같이 “opensource”, “code”, “sourceforge” 태그를 그룹핑 하여 태그 그룹 “CODE”를 생성한다. 태그 그룹 “OPENSOURCE”와 “CODE”를 대상으로 Group Intersection을 수행한 결과 그림 16의 (c)와 같이 공통적으로 사용된 태그 “sourceforge”와 “opensource”로 구성된 새로운 태그 그룹 “SOURCE CODE”를 생성할 수 있다.

⑷ Tag Group Difference

기존에 존재하는 태그 그룹에서 특정 태그를 제외한 다른

User Group	User	Tag Group	Tag	Resource Group	Resource
-	jil	-	sql	-	http://www.postgresql.org/docs/7.4/interactive/ddl-alter.html
-	cschenk	-	sql	-	http://www.greensql.net/
-	fsteeg	-	sql	-	http://talklikeaduck.denhaven2.com/2010/08/04/when-i-say-no
-	brightbyte	-	sql	-	http://hashmysql.org/index.php?title=Trees_and_hierarchical_data_in_SQL
-	nosebrain	-	sql	-	http://ibatis.apache.org/docs/dotnet/datamapper/ch03s09.html
-	dbenz	-	sql	-	http://www.mongodb.org/
-	dolefulrabbit	-	SQL	-	http://hsqldb.org/
User Group	User	Tag Group	Tag	Resource Group	Resource
-	schmidt2	-	oracle	-	http://download.oracle.com/javase/tutorial/index.html
-	schmidt2	-	oracle	-	http://download.oracle.com/javase/6/docs/
-	schmidt2	-	oracle	-	http://java.sun.com/security/seccodeguide.html

(a) “sql” 태그와 “oracle” 태그 검색 결과

User Group	User	Tag Group	Tag	Resource Group	Resource
-	jil	DB	sql, oracle	-	http://www.postgresql.org/docs/7.4/interactive/ddl-alter.html
-	cschenk	DB	sql, oracle	-	http://www.greensql.net/
-	fsteeg	DB	sql, oracle	-	http://talklikeaduck.denhaven2.com/2010/08/04/when-i-say-no
-	brightbyte	DB	sql, oracle	-	http://hashmysql.org/index.php?title=Trees_and_hierarchical_data_in_SQL
-	nosebrain	DB	sql, oracle	-	http://ibatis.apache.org/docs/dotnet/datamapper/ch03s09.html
-	dbenz	DB	sql, oracle	-	http://www.mongodb.org/
-	dolefulrabbit	DB	sql, oracle	-	http://hsqldb.org/
-	schmidt2	DB	sql, oracle	-	http://download.oracle.com/javase/tutorial/index.html
-	schmidt2	DB	sql, oracle	-	http://download.oracle.com/javase/6/docs/
-	schmidt2	DB	sql, oracle	-	http://java.sun.com/security/seccodeguide.html

(b) 태그 “sql”과 “oracle”을 대상으로 Group Composition 연산을 수행하여 생성된 태그 그룹 “DB”

그림 14. Tag Group Composition 사용사례
Fig. 14. Use case of Tag Group Composition

User Group	User	Tag Group	Tag	Resource Group	Resource
-	jil	-	tomcat	-	http://www.eclipse.org/equinox/server/http_in_container.php
-	jil	-	tomcat	-	http://tomcat.apache.org/tomcat-5.5-doc/class-loader-howto.html
-	jil	-	tomcat	-	http://www.crazysquirrel.com/computing/java/eclipse/reducing-tomcats-thread-count...
-	jil	-	tomcat	-	http://www.informit.com/articles/article.aspx?p=31443&seqNum=9
-	jil	-	tomcat	-	https://issues.apache.org/bugzilla/show_bug.cgi?id=45015
-	cschenk	-	tomcat	-	http://code.google.com/p/psi-probe/
-	macek	-	Tomcat	-	http://www.eclipsezone.com/eclipse/forums/t73908.html
-	macek	-	Tomcat	-	http://tomcat.apache.org/tomcat-6.0-doc/logging.html
-	ne	-	Tomcat	-	http://www.it-schulungen.com/seminare/softwareentwicklung/java/index.html

(a) “tomcat” 태그 검색 결과

User Group	User	Tag Group	Tag	Resource Group	Resource
-	jil	SERVER	tomcat, DB	-	http://www.eclipse.org/equinox/server/http_in_container.php
-	jil	SERVER	tomcat, DB	-	http://tomcat.apache.org/tomcat-5.5-doc/class-loader-howto.html
-	jil	SERVER	tomcat, DB	-	http://www.crazysquirrel.com/computing/java/eclipse/reducing-tomcats-thread-count.jspx
-	jil	SERVER	tomcat, DB	-	http://www.informit.com/articles/article.aspx?p=31443&seqNum=9
-	jil	SERVER	tomcat, DB	-	https://issues.apache.org/bugzilla/show_bug.cgi?id=45015
-	cschenk	SERVER	tomcat, DB	-	http://code.google.com/p/psi-probe/
-	macek	SERVER	tomcat, DB	-	http://www.eclipsezone.com/eclipse/forums/t73908.html
-	macek	SERVER	tomcat, DB	-	http://tomcat.apache.org/tomcat-6.0-doc/logging.html
-	ne	SERVER	tomcat, DB	-	http://www.it-schulungen.com/seminare/softwareentwicklung/java/index.html

(b) 태그 “tomcat”과 태그 그룹 “DB”를 대상으로 Group Aggregation 연산을 수행하여 생성된 태그 그룹 “SERVER”

그림 15. Tag Group Aggregation 사용사례
Fig. 15. Use case of Tag Group Aggregation

User Group	User	Tag Group	Tag	Resource Group	Resource
-	springnet	OPENSOURCE	free, sourceforge, opensource	-	http://www.seopher.com/articles/the_70_coolest_free_applications_in_existence
-	melnik	OPENSOURCE	free, sourceforge, opensource	-	http://www.openclipart.org/
-	hkorte	OPENSOURCE	free, sourceforge, opensource	-	http://www.mpi-inf.mpg.de/yago-naga/javatools/index.html
-	hkorte	OPENSOURCE	free, sourceforge, opensource	-	http://www.microsoft.com/student/de-DE/Products/e-books/e-book.html
-	hkorte	OPENSOURCE	free, sourceforge, opensource	-	http://www.gutenberg.org/wiki/Main_Page
-	jil	OPENSOURCE	free, sourceforge, opensource	-	http://linuxwiki.de/MathematikProgramme
-	jil	OPENSOURCE	free, sourceforge, opensource	-	http://packages.debian.org/experimental/browser-plugin-lightspark
-	jaeschke	OPENSOURCE	free, sourceforge, opensource	-	http://patentabsurdity.com/
-	brightbyte	OPENSOURCE	free, sourceforge, opensource	-	http://opengovernmentdata.org/catalogues/

(a) 태그 그룹 "OPENSOURCE" 검색 결과의 일부분

User Group	User	Tag Group	Tag	Resource Group	Resource
-	cschenk	CODE	opensource, code, sourceforge	-	http://sourceforge.net/projects/oojmerge/
-	cschenk	CODE	opensource, code, sourceforge	-	http://webdav-servlet.sourceforge.net/
-	cschenk	CODE	opensource, code, sourceforge	-	http://cloc.sourceforge.net/
-	gresch	CODE	opensource, code, sourceforge	-	http://nanoshot.sourceforge.net/
-	gresch	CODE	opensource, code, sourceforge	-	http://getgreenshot.org/
-	gresch	CODE	opensource, code, sourceforge	-	http://sourceforge.net/projects/hl7socketreader/
-	gresch	CODE	opensource, code, sourceforge	-	http://lawleecher.sourceforge.net/
-	gresch	CODE	opensource, code, sourceforge	-	http://sourceforge.net/projects/wmsopera/
-	gresch	CODE	opensource, code, sourceforge	-	http://nanoshot.sourceforge.net/

(b) 태그 그룹 "CODE" 검색 결과의 일부분

User Group	User	Tag Group	Tag	Resource Group	Resource
-	cschenk	SOURCE CODE	sourceforge, opensource	-	http://sourceforge.net/projects/oojmerge/
-	cschenk	SOURCE CODE	sourceforge, opensource	-	http://webdav-servlet.sourceforge.net/
-	cschenk	SOURCE CODE	sourceforge, opensource	-	http://cloc.sourceforge.net/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://nanoshot.sourceforge.net/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://getgreenshot.org/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://sourceforge.net/projects/hl7socketreader/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://lawleecher.sourceforge.net/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://sourceforge.net/projects/wmsopera/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://nanoshot.sourceforge.net/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://getgreenshot.org/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://sourceforge.net/projects/hl7socketreader/
-	gresch	SOURCE CODE	sourceforge, opensource	-	http://lawleecher.sourceforge.net/

(c) 태그 그룹 "OPENSOURCE"와 "CODE"를 대상으로 Group Intersection 연산을 수행하여 생성된 태그 그룹 "SOURCE CODE"

그림 16. Tag Group Intersection 사용사례
Fig. 16. Use case of Tag Group Intersection

User Group	User	Tag Group	Tag	Resource Group	Resource
-	cschenk	-	sourceforge	-	http://sourceforge.net/projects/oojmerge/
-	cschenk	-	sourceforge	-	http://webdav-servlet.sourceforge.net/
-	cschenk	-	sourceforge	-	http://cloc.sourceforge.net/
-	gresch	-	sourceforge	-	http://nanoshot.sourceforge.net/
-	gresch	-	sourceforge	-	http://getgreenshot.org/
-	gresch	-	sourceforge	-	http://sourceforge.net/projects/hl7socketreader/
-	gresch	-	sourceforge	-	http://lawleecher.sourceforge.net/

(a) 태그 "sourceforge" 검색 결과의 일부분

User Group	User	Tag Group	Tag	Resource Group	Resource
-	springnet	FREE SOURCE	free, opensource	-	http://www.seopher.com/articles/the_70_coolest_free_applications_in_existence
-	melnik	FREE SOURCE	free, opensource	-	http://www.openclipart.org/
-	hkorte	FREE SOURCE	free, opensource	-	http://www.mpi-inf.mpg.de/yago-naga/javatools/index.html
-	hkorte	FREE SOURCE	free, opensource	-	http://www.microsoft.com/student/de-DE/Products/e-books/e-book.html
-	hkorte	FREE SOURCE	free, opensource	-	http://www.gutenberg.org/wiki/Main_Page
-	jil	FREE SOURCE	free, opensource	-	http://linuxwiki.de/MathematikProgramme
-	jil	FREE SOURCE	free, opensource	-	http://packages.debian.org/experimental/browser-plugin-lightspark
-	jaeschke	FREE SOURCE	free, opensource	-	http://patentabsurdity.com/
-	brightbyte	FREE SOURCE	free, opensource	-	http://opengovernmentdata.org/catalogues/
-	brightbyte	FREE SOURCE	free, opensource	-	http://sparkleshare.org/
-	sac	FREE SOURCE	free, opensource	-	http://thinking-forth.sourceforge.net/

(b) 태그 그룹 "OPENSOURCE"와 태그 "sourceforge"를 대상으로 Group Difference 연산을 수행하여 생성된 태그 그룹 "FREE SOURCE"

그림 17. Tag Group Difference 사용사례
Fig. 17. Use case of Tag Group Difference

태그들만 그룹핑 하고자 할 때, Group Difference를 수행할 수 있다. 그림 16의 (a)와 같이 "free", "sourceforge", "opensource"태그들로 구성된 태그 그룹 "OPENSOURCE"와 그림 17의 (a)와 같이 "sourceforge" 태그를 사용한 사용자의 태깅정보를 토대로 Group Difference 연산을 수행한 결과 그림17의 (b)와 같이 태그 그룹 "OPENSOURCE"의 구성요소에서 "sourceforge"태그를 제외한 "free", "opensource" 태그로 구성된 새로운 태그 그룹 "FREE SOURCE"을 생성할 수 있다.

위와 같은 방법으로, 사용자를 대상으로 또는 리소스를 대상으로 4종류의 기본적인 연산을 수행하면 공통 관심사를 갖는 사용자 그룹뿐만 아니라 특정 주제와 관련된 리소스 그룹을 생성할 수 있다.

Ⅵ. 결론

폭소노미는 웹에 존재하는 리소스에 대해 구성원이 자유롭게 선택한 태그를 붙여서 정보를 체계화하는 새로운 분류 체계로써, 사용자들이 자유롭게 참여하는 bottom-up 방식의 분류를 기반으로 하고 있어서, 시시각각으로 변하는 웹의 리소스를 분류하는데 적합한 방법으로 WWW 분야에 널리 사용되고 있다. Del.icio.us, Flickr, YouTube 등과 같은 최근의 폭소노미 기반 협력 태깅 시스템들을 살펴보면, 폭소노미 데이터를 체계화하고 유용한 서비스를 제공하기 위해 폭소노미의 각 구성요소들(사용자들, 태그들, 리소스들)을 그룹핑하기 위한 기능들을 추가하고 있다.

본 논문에서는, 일반적인 폭소노미 정의에 "그룹" 개념을 도입하여 확장한 그룹화된 폭소노미 모델을 제안하고 그룹화된 폭소노미를 기반으로 각 요소들을 체계화하기 위한 기본적인 연산들을 정의하였다. 또한, 본 연구에서 제안한 기법을 지원하는 시스템을 개발하였으며, 기본적인 연산들을 사용하여 각 요소들을 체계화하는 사용사례를 설명하였다. 본 연구에서 제안된 모델과 연산들을 사용함으로써, 공통 관심사를 갖는 사용자 그룹과 같은 의미를 갖는 태그 그룹, 특정 주제와 관련된 리소스 그룹 등과 같은 폭소노미 데이터에 숨겨져 있는 유용한 정보들을 수월하게 추출해 낼 수 있다. 또한, 사용자가 서로 다른 폭소노미기반 사이트에서 각각 사용한 태그들 또는 리소스들을 통합하여 태깅하고자 할 때 본 연구에서 제안한 기본 연산들이 유용하게 사용될 수 있다.

향후 연구에서는, 현존하는 다양한 폭소노미 기반 시스템에 본 연구에서 제안한 기본 연산을 적용하여 그룹화된 폭소노미를 구축하기 위해 다양한 crawler를 개발할 예정이며, 그룹화된 결과를 효율적으로 표현하기 위한 방법에 대하여 추가적인 연구가 필요하다.

참고문헌

[1] Del.icio.us, *http://delicious.com*
[2] Flickr, *http://www.flickr.com*
[3] YouTube, *http://www.youtube.com*
[4] Bibsonomy, *http://www.bibsonomy.org*
[5] GroupMe!, *http://www.groupme.org*
[6] P. Anon and L. Kristina, "Constructing folksonomies from user-specified relations on flickr", In Proceedings of the 18th Int. Conf. on World wide web, pp. 781-790, Madrid, Spain, April. 2009.
[7] A. Fabian, H. Nicola and K. Daniel, "A NOVEL APPROACH TO SOCIAL TAGGING: GROUPME! Enhancing Social Tagging Systems with Groups", In 4th Int. Conf. on Web Information Systems and Technologies, pp.42-49, Funchal, Portugal, May. 2008.
[8] A. Hotho, R. J·aschke, C. Schmitz, and G. Stumme, "BibSonomy: A Social Bookmark and Publication Sharing System", In Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th Int. Conf. on Conceptual Structures, pp. 87-102, Aalborg, Denmark, July, 2006.
[9] C. Marlow, M. Naaman, D. Boyd, and M. Davis, "HT06, tagging paper, taxonomy, flickr, academic article, to read", In HYPERTEXT '06: Proceedings of the 7th conference on Hypertext and hypermedia, pp. 31-40, Odense, Denmark, August, 2006.
[10] B. Pierpaolo, G. Domenico, L. Filippo and S. Giovanni, "Recommending Smart Tags in a Social Bookmarking System", Bridging the Gep between Semantic Web and Web 2.0, pp.22-29, September, 2007.
[11] X. Shengliang, B. Shenghua, F. Ben, S. Zhong and Y. Yong, "Exploring folksonomy for personalized search", In Proceedings of SIGIR'2008. pp.155-162, singapore, July, 2008.
[12] B. Simone, Z. Vlentin, and H. Hans-Jorg, "Social Semantic Bookmarking", the 7th Int. Conf. on Practical Aspects of Knowledge Management, LNAI 5345, pp.62-73, Yokohama, Japan, November, 2008.
[13] Y. Dong-Hee, "A System Framework and Research Challenges for the Semantic Web Applications," Journal of the Korea Society of Computer and Information, v.14, no.12, pp.255-266, December, 2009.
[14] L. Jun-Hee , "Development of e-PBL Based on Web 2.0," Journal of the Korea Society of Computer and Information , v.13, no.2, pp.59-66, March, 2008.

저 자 소 개

강 유 경

2003년 : 선문대학교 컴퓨터정보학부 (이학사)

2005년 : 선문대학교 일반대학원 전자계산학과 (이학석사)

2009년 : 선문대학교 일반대학원 컴퓨터정보학과 (이학박사)

2009년 3월~현재 : 선문대학교 컴퓨터공학부 BK21 박사후과정생, 연구교수

관심분야 : Formal Concept Analysis, Semantic Web Mining, 시맨틱 웹, 소프트웨어공학 등

Email: yukyung.kang@gmail.com

황 석 형

1991년 8월 : 강원대학교 전자계산학과 조기졸업 (이학사)

1993년 4월 : 일본 오사카대학교 대학원 정보공학과 (공학석사)

1997년 4월: 일본 오사카대학교 대학원 정보공학과 (공학박사)

1997년 3월~현재 : 선문대학교 컴퓨터공학부 전임강사, 조교수, 부교수, 정교수

2007년 1월~2008년 1월 : 아일랜드국립대학교 DERI 객원연구원

관심분야 : 소프트웨어공학, 객체지향, 온톨로지공학, 시맨틱 웹, Formal Concept Analysis, Semantic Web Mining

Email: shwang@sunmoon.ac.kr