Multidiszciplináris tudományok, 3. kötet. (2013) 2. sz. pp. 289-296.

FORMÁLIS FOGALMAKRA ÉPÜLŐ ÁLLOMÁNYKEZELŐ RENDSZER

Piller Imre
doktorandusz, Miskolci Egyetem, Alkalmazott Matematikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: piller@jerry.uni-miskolc.hu

Fegyverneki Sándor
egyetemi docens, Miskolci Egyetem, Alkalmazott Matematikai Tanszék
3515 Miskolc, Miskolc-Egyetemváros, email: matfs@uni-miskolc.hu

Összefoglalás

Az állomány- és dokumentumkezelésnek kulcsfontosságú szerepe van az informatikában. Ugyan a cél egy általánosan használható állomány szintű adatszervezési mód megvalósítása, a jelen munka mégis elsősorban egy alternatív adatszervezési- és visszakeresési módszerrel foglalkozik. A javasolt megoldás formális fogalmakat használ jegyzékek és a szigorúan hierarchikus osztályozási mód helyett. Ez a megközelítés számos problémát megold (mint például az átfedéses kategóriák, vagy a szükségtelen redundancia kezelését), de új technikák használata szükséges a rendszer felhasználóbarát kialakításához. A formális fogalomelemzés tárgyköre adja a szükséges matematikai modelleket, a rendszer elvi hátterét.

Kulcsszavak: *állománykezelő rendszerek, fogalomhálók, információ menedzsment*

Abstract

The file and document management has a key role in current Informatics. Its main principles are derived from the first computer system implementations. This paper shows an alternative method to organize and retrieve information. The proposed method uses formal concepts instead of directories and strict hierarchical structures. This approach is able to solve many problems of classical information organizational problems (for example overlapping categories and unnecessary redundancy) but requires some new techniques for user friendly design. The theory of formal concept analysis provides a great mathematical foundation for these types of systems.

Keywords: *file handling systems, concept lattices, information management*

1. Bevezetés

Az állománykezelő rendszerek a leggyakrabban használt alkalmazások közé tartoznak. Az Internet terjedésének és fejlődésének köszönhetően a webböngészők kezdik háttérbe szorítani őket. Amíg az elektronikusan tárolt dokumentumainkat saját birtokunkban szeretnénk tudni (nem pedig egy harmadik féltől származó szolgáltatást veszünk igénybe) addig ezek alkalmazására szükség lesz.

Az új technológiáknak köszönhetően a felhasználóknak egyre kevesebbet kell foglalkozniuk az állományok kezelésével kapcsolatos szoftveres és hardveres megvalósítással. A fájlok elnevezésére vonatkozó kötöttségek lazultak, az ikonokkal való megjelenítés kellő absztrakciót nyújt a nem elsődlegesen informatikával foglalkozó felhasználók számára is.

A tipikus állománykezelő rendszerekben az elsődleges rendszerezési elvet a könyvtárstruktúra adja. A felhasználó tetszőleges hierarchikus szerkezetet építhet fel, a számítógépen tárolt tartalmakat osztályozhatja. Az átlapolódó kategóriák kezelése, illetve az egymástól független (*ortogonális*) szempontok azonban problémát jelentenek. Elsősorban ezeknek a megoldása miatt szükséges új rendszerezési elveket és módszereket keresni.

2. Fájlok és dokumentumok

Érdemes megvizsgálni, hogy ténylegesen mi is a különbség a között, ha fájlokról, vagy ha dokumentumokról beszélünk. A fájlok és a dokumentumok is az adatok egy kezelési egységét jelölik. A fájl esetében ezt a számítógépes megvalósítás szemszögéből nézzük, míg a dokumentumoknál az adatok felhasználási módja a mérvadó. Jellemzően a dokumentumoknak egy-egy fájl felel meg. Vannak azonban olyan esetek, amelyek miatt a két fogalmat még sem tekinthetjük azonosnak.

- Számos olyan speciális célú fájl létezik, amely nem dokumentum, és az átlagos felhasználónak tudomása sincs a létezésükről.
- Az operációs rendszerek a jegyzékeket/mappákat is speciális állományokként kezelhetik.
- Bizonyos dokumentumok (például weboldalak) felhasználói szemszögből ugyan egy dokumentumként kezelhetők, de több fájl is tartozik hozzájuk.
- Az új dokumentumformátumok (*például a docx, xlsx, odt, ods kiterjesztésűek*) speciális tömörített fájlok, melyek szintén több, szabványos formátumú állományból épülnek fel, és a tömörítés csak egy keretbe foglalja őket.

A két fogalom között tehát jelentős különbségek vannak, és emiatt a dokumentum- és fájlkezelési módok is eltérnek. A kettőt jelen esetben az a tendencia kapcsolja össze, hogy a dokumentumok kezelésénél megjelenő új módszereket fokozatosan átveszik az állománykezelő rendszerek is. A fájlrendszer szintű megvalósítás mindenképpen növeli az adott módszer hatékonyságát.

3. Információ visszakeresés

Az információkat elsősorban azért tároljuk, hogy szükség esetén gyorsan vissza tudjuk keresni azokat. Az információ megléte mellett az idő is egyre kritikusabb tényező. Fontos továbbá, hogy a keresés kényelmes, a kereséshez használt felület intuitív legyen. A fában történő keresés nagy adathalmazok esetében nehézkessé válhat. A kulcsszavas, illetve a

címkézésre vagy hasonló meta-információkra építkező keresőrendszerek ekkor jutnak szerephez.

A könyvtárstruktúrák használatával kapcsolatban már készültek felmérések, mint például Sarah Henderson kutatása, melyben hat személy adatkezelési szokásait vizsgálta meg, különös tekintettel a jegyzékek és a fájlok elnevezési módjára vonatkozóan [5]. Ez alapján sikerült meghatározniuk azokat a fő csoportokat, melyekbe besorolhatók a napi gyakorlatban használt osztályozási szempontok. Ilyenek például az idő, a feladat, a témakör vagy a fájlok típusa. Ami a felmérésből kitűnik, hogy különböző felhasználók egymástól függetlenül nagyon hasonló struktúrákat hoznak létre, a különbségek pedig a szigorúan hierarchikus osztályozási mód korlátait jelzik.

A meta-információkat a fájlokhoz maga a felhasználó vagy az állományokat kezelő rendszer automatikusan is hozzárendelheti. Egy fontos kérdés az, hogy a rendszer mennyiben képes arra, hogy a felhasználó számára lényeges információkat kinyerje az adott állományokból. Az elterjedt kulcsszavas keresőszolgáltatások jelenlegi hatékonysága azt mutatja, hogy a feladat nagy megbízhatósággal megoldható. Az indexelés elvégzéséhez szükséges algoritmusok és adatok egy része azonban publikusan nem érhető el, illetve nagyobb számítási kapacitást igényelnek, mint amit a jelenlegi személyi számítógépek biztosítani tudnak, ezért más, kisebb erőforrásokat igénylő módszereket kell alkalmazni.

A címkézés (*tagging*) egy olyan elterjedt módszer, amely kiegészíti és megkönnyíti az állományok visszakeresését. Az állományokat a rájuk leginkább jellemző címkékkel láthatjuk el. Az osztályozást ekkor maguk a címkék jelentik, így az átlapolódó osztályok kezelése adott. A keresésnél a címkék sorrendjére szintén nem kell tekintettel lennünk. A könyvtárstruktúrával ellentétben itt nem szükséges az általános jellemzőktől a speciálisak felé haladnunk.

A címkézéses módszerek manapság főként a tartalomkezelő (*Content Management System*) illetve a levelező rendszerekben vannak jelen. Gyakorlatilag az elterjedtebbek közül már az összesben megjelenik a címkézés, mint elérhető funkció. Vannak speciálisan dokumentumok kezeléséhez készített alkalmazások is (például *OpenKM*), melyekben a kategóriák kialakításához és a kereséshez kulcsszavakat adhatunk meg. Számos más alkalmazási területen is jelen van. A szemantikus web- és desktop koncepciókba nagyon jól illeszkedik, és az adatbáziskezelő rendszerekkel a címkék, fájlok és kapcsolataik adatainak kezelése is megoldható. Nyitott problémaként többek között az szerepel, hogy mennyire alkalmas nagy mennyiségű adat kezelésére egy ilyen rendszer, illetve hogy hogy nézne ki az a felület, amelyet a laikus felhasználó is könnyedén át tud tekinteni. A címkézési módszer szélesebb körben való elterjedését elsősorban az nehezíti, hogy nincsenek még meg azok az egységes konvenciók, mint a könyvtárstruktúra esetében. A felületek általában bonyolultabbak, amivel az átlagos felhasználók már kevésbé boldogulnak.

4. Formális fogalomelemzés

A hierarchikus fájlrendszerekről említett problémáiról, illetve a koncepciókon alapuló fájlrendszerek lehetőségeiről a [1] cikkben találhatunk egy rövid összefoglalót. Ebben külön problémaként említi a hagyományos fájlrendszerekben a fájlok elnevezésének fontosságát, mivel a rendelkezésre álló eszközök többsége erre hagyatkozik. Amennyiben nem tudjuk a keresett fájl nevét, típusát, vagy a létrehozásának, módosításának időpontját, akkor kénytelenek vagyunk végigjárni a hierarchiát, és külön megnézni a szóbajöhető lehetőségeket. A javasolt megoldás egy koncepciókra épülő állománykezelő rendszer (*Concept File Mana-*

gement) létrehozása. A "koncepció" az említett cikkben még csak egy szempont szerinti kategorizálási egységet jelöl.

A háló alapú információs rendszer ötlete már az 1960-as években felmerült [7]. A probléma a gyakorlati alkalmazhatóságával volt. A FaIR nevű rendszer egy alkalmazási mintaként szolgál, amely megmutatja, hogy a logikai kifejezésekkel megadott lekérdezések hogyan kapcsolódnak a hálóelmélethez.

4.1. A matematikai modell

A formális fogalomelemzés tárgyköre a hétköznapi értelemben használt fogalmaknak egy kellően letisztult matematikai formalizmust biztosít. Bevezeti a formális kontextus fogalmát, amely három halmazt foglal magába [4].

- A G az objektumok halmaza. Jelen esetben ebben a fájlokhoz tartozó, rendszerezés szempontjából érdekes adatok kerülnek bele.
- Az M az attribútumok halmaza. Az előzőekben említett címkék vagy kulcsszavak tartoznak ebbe a halmazba.
- Az I incidencia reláció írja le a G és M halmazok közötti kapcsolatot.

A formális kontextusban a G és M halmazok részhalmazain értelmezett derivációs alapművelet, már egy egyszerű lekérdezési módnak tekinthető. Ez a G → M, illetve M → G irányban is értelmezett. Az első esetben az A objektumhalmazhoz azon attribútumokat rendeljük, melyek mindegyike kapcsolatban van az adott objektumokkal, vagyis

$$A' = \{m \in M | (g, m) \in I, \forall g \in A\}$$

és analóg módon

$$B' = \{g \in M | (g, m) \in I, \forall m \in B\}.$$

A deriváció jelöléséhez itt is vesszőt használhatunk. Abban az esetben, ha két halmazra teljesül, hogy A = B' és B = A' akkor az (A, B) párt formális fogalomnak nevezzük, amelynek az objektum részhalmazát *attribútum-extenzió*nak, az attribútum részhalmazát pedig *objektum-intenció*nak nevezzük.

A kontextusban lévő fogalmak között kétirányú tartalmazási kapcsolatot értelmezhetünk, amely egy részbenrendezés az összes fogalomra nézve, mégpedig

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \iff B_2 \subseteq B_1.$$

A formális kontextushoz tartozó összes fogalom ezzel a részben rendezéssel eredményezi majd a fogalomhálót, amit szokás $\mathfrak{B}(G, M, I)$ formában is megadni.

4.2. A fogalomhálók szerepe és megjelenési módjai

A formális fogalomelemzés különféle módokon megjelenik az informatikában [8]. A 90'-es évek végétől vált jellemzővé, hogy a korábban használt kifejezések (például dokumentum-kifejezés hálók (*document-term lattices*)) helyett a fogalomelemzés elnevezései kerültek előtérbe. A módszerek általánosságát bizonyítja, hogy egyre szélesebb körben alkalmazzák, mint például a tudásábrázolásban, információ visszakeresésben, mesterséges intelligenciával kapcsolatos területeken, ontológiákban.

Az állománykezelésben való alkalmazhatóságával kapcsolatban folynak aktív kutatások. A fogalomháló alapú fájlrendszer implementálása lényegesen bonyolultabb feladat,

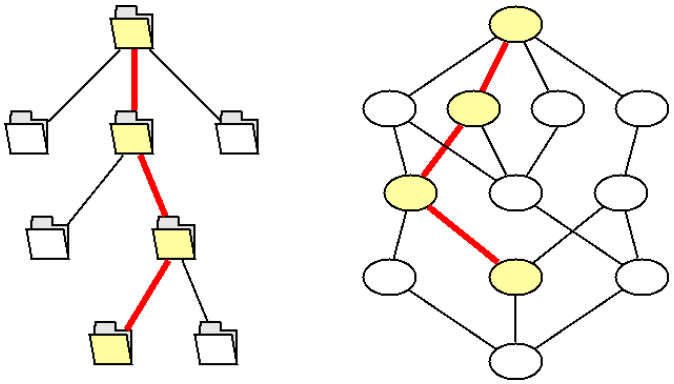
mint a hierarchikus fájlrendszereké. Ben Martin tézisében [6] megvizsgálja a lehetséges megvalósítási módok egy részét, illetve összehasonlítja azok hatékonyságát. Ezek szükségesek annak a belátására, hogy a fogalomhálón elvégzendő műveletek egy elvárható maximális válaszidőn belül lesznek nagyobb kontextusok esetében is.

5. Navigáció a fogalomhálóban

S. Ferré és O. Ridoux [2] cikkében már az előzőeknek megfelelő matematikai modellt találunk. Említés esik benne a hierarchikus rendszer linkekkel való kibővítésének lehetőségéről is. Ez a UNIX rendszerekben már az első változatoktól kezdve elérhető funkció. A problémát a linkek karbantartása jelenti, mivel azok érvénytelenné válhatnak, illetve nehéz őket más típusú rendszerekbe átvinni. A cél tehát mindenképpen egy tisztán fogalmi hálókra építkező modell kialakítása.

A hálós szerkezet rugalmas tudásábrázolási és kezelési módja a felhasználói felület kialakításánál gondot okozhat. A lekérdezések közvetlen megadása nem várható el, így valamilyen navigációs módszerre van szükség. (A navigáció alatt azt az iteratív folyamatot értjük, melynek során a felhasználó lépésenként módosítja a lekérdezést.)

A navigációban a fő különbséget a könyvtárszerkezethez képest az jelenti, hogy a hálóban nincs egyedi szülő, több lehetséges irány is adott általában. Az 1. ábrán egy nagyon leegyszerűsített példát láthatunk arra, hogy az útvonalak hogyan jelennek meg az adott struktúrában.



1. ábra. *Példa útvonalakra jegyzékstruktúra és fogalmi háló esetében.*

A nagy adathalmaz áttekinthetőségét, és is így magát a navigációt az is nehezíti, hogy az adatok megjelenítése közel sem annyira magától értetődő a fogalomháló esetében, mint a jegyzékstruktúránál. A fogalmaknak rendelkezésre kell hozzá állniuk, amelyek felsorolása szintén egy számításigényes művelet. A hálók Hasse diagramjának ábrázolására nincs egyértelműen jó módszer, a fogalmi hierarchia illetve az adott feladat dönti el, hogy melyik ábrázolási mód az előnyös. Az egész háló kirajzolása csak nagyon kis kontextusoknál jelenthet segítséget a felhasználónak, mivel már egy 20-30 fogalmat tartalmazó háló áttekintése sem egyszerű feladat.

5.1. Navigációs műveletek

A fogalom orientált rendszerben a hagyományos jegyzékstruktúrában elvégezhető műveletekhez hasonlóakra van szükség. Ehhez először definiálni kell az újszerű struktúrához tartozó alapvető fogalmakat.

Az abszolút- és relatív útvonalak, mint a lekérdezésként használt címkehalmaz elemeinek cseréjére vagy módosítására vonatkozó leírás jelenik meg. Az abszolút útvonallal közvetlenül megadhatjuk, hogy mely címkék szerepeljenek benne, míg a relatív esetében jelezzük, hogy melyeket kell hozzáadni vagy elvenni a halmazból, esetleg negált címkeként szerepeltetni. Ezeket rendre a +, − és ! prefixekkel jelölhetjük. A megoldásnak az az előnye is megvan, hogy a lekérdezés és az útvonal tulajdonképpen egyet jelent. Megtehetjük azt, hogy a rendszerbe újonnan bekerülő állományt úgy helyezzük el, mint ha azt keresnénk, mivel a lekérdezés eredményéhez azt közvetlenül hozzá lehet adni.

Ahhoz, hogy a lekérdezéssel történő keresés hatékony legyen, be kell vezetni a negációt is. Ezt a lekérdezésekben ellentett attribútumként adhatjuk meg. Ezek olyan, újonnan bevezetett attribútumoknak is tekinthetők, melyek akkor kerülnek hozzárendelésre az objektumokhoz, ha az eredeti attribútum nem tartozott hozzá. A számítógépes megvalósításban ezek nem feltétlenül jelennek meg mint adatok, hanem csak a lekérdezést kiértékelő komponens veszi figyelembe az attribútumra szerepére vonatkozó módosítót.

A gyökér jegyzék analógiája a fogalomhálóban a legáltalánosabb fogalom. Ehhez semmilyen attribútum sem tartozik, viszont minden objektum benne van. Az elemi navigációs lépést a kereséshez egy attribútum hozzáadása vagy elvétele jelenti. Mindig van tehát egy aktuális fogalom, és egy másik fogalomba való eljutáshoz szükséges lépések minimális száma már távolságdefinícióként értelmezhető.

6. Parancssoros felhasználói felületek

A grafikus felhasználói felületek térhódítása ellenére még manapság is fontos szerephez jutnak a parancssoros kezelőfelületek (*Command Line Interface*). Ennek a következő okai vannak.

- A maihoz hasonló rendszerek kezdeti változatainál még csak ezek álltak rendelkezésre. Ez már egy, az ember számára is érthető és kezelhető réteget képvisel. Használata az átlagfelhasználó számára kényelmetlen és tanulást igényel. Az eltávolítása nem indokolt, és kompatibilitási okok miatt sem lenne célszerű megtenni.
- Bizonyos esetekben egyszerűbb vele a komplikált feladatok elvégzése, mint ha azt grafikus felület segítségével tennénk meg.
- A rendszereket úgy készítik el, hogy minden grafikus felület segítségével kiadott utasításnak megfeleltethető legyen a parancssoros változata.

A fogalomháló kezeléséhez az előző szakaszban említett elemeket parancsnyelvi formában is meg kell adni. Mivel a navigációs mód sok hasonlóságot mutat a hagyományos rendszerekkel, ezért érdemes azok elterjedt parancsnyelveit mintáknak tekinteni.

A jegyzékstruktúra listaként megadott útvonalaihoz képest itt halmazokról van szó, amely különbséget a jelölésekben is célszerű kiemelni. Az abszolút útvonalra megadásához a kiválasztott jelölés szögletes zárójeleket használ, melyben az attribútumok idézőjelek között vannak felsorolva. Egy ilyen abszolút útvonal lehet például: `["attribútum 1", "attribútum 2"]`. A relatív útvonal a címkék hozzáadására vagy eltávolítására vonat-

kozik, amit így a + és − jelekkel adhatunk meg. Az ellentett attribútumok csak hozzáadásnál lényegesek, így annál jelölésben a `+!` helyett a `!` prefix is elegendő. Egy relatív útvonal például a következő formában nézhet ki: `+"attribútum 1", -"attribútum 2", !"attribútum 3"`.

Az aktuális fogalomban lévő objektumok listázásához az `ls` parancs használható. Ez együtt alkalmazható az előbbi két útvonaltípussal a könyvtárstruktúrák esetében használt `ls` parancshoz hasonlóan. Itt szükség van még arra a felhasználási módjára is, hogy adott objektumokhoz megkereshessük vele, hogy milyen attribútumok tartoznak. Ehhez az objektumok nevét fel kell sorolni mint paramétereket, például az `ls "objektum 1" "objektum 2" "objektum 3"` lekérdezés azoknak az attribútumoknak a listáját eredményezi, amelyek mind az `"attribútum 1"`, `"attribútum 2"` és `"attribútum 3"` objektumokon rajta vannak.

A `cd` parancs helyett itt a `tag` parancs az, amelyik az aktuális fogalom váltására alkalmas. E mögött egyszerűen csak meg kell adni az útvonalat, például: `tag +"attribútum 1" -"attribútum 2"`.

A `tag` parancs a `mv` (move) áthelyezési művelet helyett használható. A paraméterezésének így két része lesz; az elsőben azon objektumok listáját kell megadni, amelyekre vonatkozik a művelet, a másodikban pedig az útvonalat. Az `"objektum 1"` és `"objektum 2"` áthelyezését egy speciálisabb, `"attribútum 1"`-el bővített fogalomba a `tag "objektum 1" "objektum 2" +"attribútum 1"` parancs kiadásával lehet megoldani.

A `cp` (copy, másolás) és az `rm` (remove, törlés) parancsok használata tulajdonképpen már adódik az előzőek mintájára. Az átnevezésre ebben a konstrukcióban a mv már nem alkalmas (mivel a fájlnév nem képezi az útvonal részét), ezért ahhoz egy name parancsot is be kell vezetni. Ennek a szintaktikája: `name "aktuális név" "új név"`.

7. További fejlesztési lehetőségek

A parancssoros felhasználói interfész a rendszer használatát már lehetővé teszi, viszont az általános felhasználási módhoz a későbbiekben egy grafikus felhasználói felület is szükséges lesz. A jelenleg elterjedt felhasználói felület kialakítások ezekhez mintaként használhatók, hogy a felhasználók számára ne jelentsen nehézséget a fogalomhálóra épülő rendszer használata sem.

A kontextus tárolásához és a lekérdezések megvalósításához vannak ugyan általános feladatokra elkészített eszközök, de fokozatosan le kell cserélni azokat olyanokra, amelyek hatékonyabban támogatják a bemutatott, általánosabb szerkezetet. Az egyik lehetőséget az ontológiákban használt RDF erőforrás leíró- és SPARQL lekérdező nyelv használata jelentheti [3].

8. Köszönetnyilvánítás

A kutató munka a Miskolci Egyetem stratégiai kutatási területén működő Mechatronikai és Logisztikai Kiválósági Központ keretében valósult meg.

9. Irodalom

[1] Ali Sajedi Badashian, Mehregan Mahdavi, Seyyed Hamidreza Afzali:Supporting Multiple Categorization using Conceptual File Management, American Journal of Scientific Research, ISSN 1450-223X Issue 35 (2011), pp. 129-136

[2] Sébastien Ferré, Olivier Ridoux: A File System Based on Concept Analysis, Research riport, no 3942, 2000

[3] Sébastien Ferré: Conceptual Navigation in RDF Graphs with SPARQL-like Queries, International Conference on Formal Concept Analysis LNCS 5986 (2010) 193-208.

[4] Bernhard Ganter, Gerd Stumme, Rudolf Wille: Formal Concept Analysis, Foundations and Applications, Springer-Verlag Berlin Heidelberg 2005

[5] Sarah Henderson: Genre, Task, Topic and Time: Facets of Personal Digital Document Management, CHINZ '05, July 6-8, 2005 Auckland, New Zealand

[6] Ben Martin: Formal concept analysis and semantic file systems, University of Wollongong Thesis Collections, 2008

[7] Uta Priss: Lattice-based Information Retrieval, Knowledge Organization, Vol. 27, 3, 2000, p. 132-142.

[8] Uta Priss: Formal Concept Analysis in Information Science, Cronin, Blaise (ed.), Annual Review of Information Science and Technology. Vol 40, 2006, p. 521-543.