

## 1.3. Távközlési közönségszolgálat gépi támogatásának lehetőségei - Specifikáció

### Bevezetés

A MEO projekthez kapcsolódó távközlési (hibakezelő) szakontológia és a hozzá kapcsolódó hibakezelő rendszer specifikációját tartalmazza ez a dokumentum.

Az A hibakezelési ontológia konszolidálása c. dokumentummal megegyező szóhasználatot alkalmazunk.

### Célok

Elsősorban az egyéni ügyfeleknél felmerülő, az internethasználattal kapcsolatos hibák elhárítására alkalmas rendszer felépítése a célunk. Ezen célok elérése érdekében az [A szakontológiával támogatott hibakezelés alapfogalmai](#) című dokumentumban megfogalmazott célok közül a hibákkal kapcsolatos részeket modelleztük. A szervezeti, működési felépítés alapján történő segítségnyújtással ez a rendszer nem foglalkozik. A hibakezelés azon részei, melyek nem a hiba felderítéséhez kapcsolódnak (pl.: beszélgetés rögzítése) szintén nem képezik ennek a rendszernek részét.

Tehát az alábbiakat igyekszik megvalósítani rendszerünk:

- Hibák, hibatípusok fogalmi szinten történő leírása, elméleti és gyakorlati szempontok szerinti osztályozása, kategóriák létrehozása.
- A hibák lehetséges okainak megnevezése és leírása.
- Az okozati összefüggések feltárása.
- A hibajelenségek, azaz a hibák észlelési módjának és következményeinek rögzítése.

### Fogalmak

Néhány központi jelentőségű fogalmat írunk le röviden. A hozzájuk kapcsolódó bővebb leírás, mely a tulajdonságaikat mutatja be, az A hibakezelési ontológia konszolidálása című dokumentumban található.

Az itt fel nem sorolt fogalmakról részletesebben az [Ontológia-építő nyelvek összehasonlítása](#) c. dokumentumban található leírás.

### Gyanú

A gördülékeny és adaptív működés igényével bevezetjük a Gyanú fogalmát is. Első közelítésben annyit takar ez a kifejezés, hogy a tapasztalt [Hibajelek](#), [Konfiguráció](#), lefuttatott [Tesztek](#) és kipróbált [Megoldások](#) és azok eredményeinek függvényében célszerűnek látszik menedzselni egy mértéket, mellyel a rendszer futása közben mérjük a [Problémák](#) fennállásának lehetőségét. Azaz az a (gyöngye) elvárásunk, hogy a rendszer futás közben minden lehetséges [Problémához](#) nyilvántartsunk egy „gyanú-mértéket”, mely minél nagyobb, annál inkább legyen megalapozott az a feltevés, hogy az adott [Probléma](#) fennáll a rendszerben.

### Hibajel

Olyan rendellenes jelenség, melyet az ügyfél közvetlenül megtapasztal. Minden [Problémához](#) társul legalább egy, öt feltétlenül kísérő [Hibajel](#). A kísérő [Hibajelek](#) hiánya alapján zárhatunk ki potenciális [Problémákat](#).

## Kódolási séma

Adattartalmi megszorítás, olyan leírás, amely egy mező tartalmára vonatkozóan megszorításokat határoz meg, azok betartását segíti. Ez a technikai segédeszköz a helyes adatbevitelt segíti azáltal, hogy nem engedi nem megfelelő adatok bevitelét a felületen, valamint strukturált adat bevitelét. Ennek segítségével speciális megjelenítés is lehetséges. Például egy kapcsolat lehetséges kódolási sémája: „azonosító, majd / jel, majd azonosító”.

## Kódolási séma tár

A [kódolási sémák](#) tárolása központosított módon történik.

## Konfiguráció

Azon rendszer leírása, melyben az ügyfél az általa leírt [Hibajelek](#)eket tapasztalta. Része lehet adott esetben nemcsak az ügyfél alhálózatának hardware- és software-konfigurációja, de pl. adott weboldal el nem érése esetén a weboldalt üzemeltető webszolgáltató szerverének és magának az adott weboldalnak leírása is (amennyire szükséges és lehetséges).

## Konfigurációs feltétel

Minden [Probléma](#) fennállásának vannak szükséges feltételei. Legáltalánosabb esetben ez annyi, hogy legyen olyan software- vagy hardware egység az ügyfél rendszerében, melyre az adott [Probléma](#) vonatkozhat. Azaz pl. egy wireless hálózatban nem fordulhat elő egy „Kábelszakadás” probléma, így mondhatjuk, hogy ennek a problémának szükséges feltétele (vagy konfigurációs feltétele), hogy legyen kábel az adott hálózatban.

Ilyen elemek leírásával érhető el, hogy a rendszer kevesebb „buta” kérdést tegyen fel azáltal, hogy nagyobb ütemben lesz képes kizárni a biztosan nem fennálló [Problémákat](#).

## Központi kereső

A [központi metatár](#)ba begyűjtött adatokban közös keresési lehetőséget biztosít (kereső motor).

## Központi metatár

A központi metatár adatbázisa az egyes [metatár](#)akból begyűjtött adatokat tárolja. Ez szolgál a [központi keresés](#) alapjául.

## Megnevezés és konfiguráció tár

A rendszer központosított eleme, mely a névvel ellátott rendszerelemek megnevezését (és kapcsolódó információkat) tárol. Fő elemei a [minősítő tár](#), a [kódolási séma tár](#), valamint a kezelt [metatár](#)ak, és [névter](#)ek elnevezései, konfigurációi.

## Megoldás

Egy-egy [Problémához](#) rendelhetünk egy vagy több lehetséges Megoldást is. Ezen megoldásoktól elvárt, hogy *elképzelhető* legyen, hogy megoldják azt a problémát, amihez csatolták őket.

## Metaleírás (vagy metarekord)

A [metatár](#)ban tárolt egyed (rekord). Szerkezetét a metatár sémája határozza meg.

## Metatár

Feladata, hogy a szerkesztőfelületet nyújtson az ontológia valamint a konkrét probléma felderítéséhez és megoldásához. Többféle séma alapján képes az adatokat tárolni és elérni. Az

egy metatárak kapcsolódnak egymáshoz szintén metatárakat használva. Ennek használatával jön létre az ontológia és ennek segítségével lehet egy egységes felületen keresztül az ügyféllel kapcsolatos információkat megjeleníteni.

A rendszer ezen egysége, egy meghatározott [metatár sémával](#) leírt adatokat kezel. Ezen adatokat tárolja (a metatár adatbázisban), karbantartásukra lehetőséget ad (a metatár karbantartó alkalmazással), és az adatokat a központ felé szolgáltatja (a metatár szolgáltató segítségével). Egy egyedi metatár jellemzően egy nyilvántartás (gyűjtemény) adattartalmának kezeléséért felelős. A [névtér](#)ek speciális metatárak. [központi keresés](#) alapjául.

### **Metatár séma**

Sémák határozzák meg az [metatár](#)akban, [névtér](#)ekben tárolt [metarekord](#)ok formai szabályait. A sémák formája alapvetően: [mMEO](#).

### **Minősítő tár**

A [metatár sémák](#)ban használható minősítők nyilvántartása. Minősítők alapján történik a [metarekord](#)ok jellemzőinek leírása.

### **mMEO**

A minősített [Dublin Core](#) metaleírás ajánlásnak (mDC) a MEO jelen projektje számára kialakított, kibővített és jelen rendszerben használt változata.

### **Névtér**

Speciális [metatár](#), mely referencia-jellegű adatot tartalmaz (pl. eszközök megnevezései, különböző tulajdonságai). Egy névtér elemeire metatárak elemei ([metarekord](#)) hivatkozhatnak. A névtér tekinthető speciális [kódolási sémának](#), ahol a névtér a lehetséges értékek teljes körű felsorolását tartalmazza.

### **Objektumtár**

Olyan központi tároló, ahol tetszőleges állomány tárolható (pl. HTML dokumentum, ábra). A [metatár](#)ak elemei ([metarekord](#)ok) hivatkozhatnak objektumtári elemekre.

### **Proxy**

Egy [névtér](#) elem kiválasztását segítő komponens, mely jellemzően a metatár karbantartóban felhasználói felülettel is rendelkezik.

### **Probléma**

A felmerült hiba forrását, okát írja le; ilyen lehet pl. az „elszakadt a koaxiális-kábel az ügyfél számítógépe és modeme között” probléma.

### **Teszt**

Amennyiben úgy döntünk, hogy egy potenciális [Probléma](#) jelenlétét nem az okozott [Hibajel](#)ekkel, hanem egy olyan vizsgálattal ellenőrizzük, ami közvetlenül és specifikusan az adott [Problémára](#) vonatkozik, akkor egy, az adott [Problémához](#) csatolt [Tesztet](#) hajtunk végre. Azaz, minden [Problémához](#) rendelhetünk egy vagy több [Tesztet](#), melyek végrehajtását kérés esetén ajánljuk az ügyfélnek. Ezen tesztekkel kapcsolatos elvárás az, hogy ha a [Probléma](#) fennáll, azt a [Teszt](#) mindenképp mutassa ki (ám ha a [Teszt](#) eredménye eltér a normálistól, az nem kell feltétlenül azt jelentse, hogy az adott [Probléma](#) valóban fennáll). Ilyen [Teszt](#) lehet pl. „Ellenőrizzük, hogy az általunk begépett IP-cím egyezik-e a szolgáltató által megadottal.”

## Modell

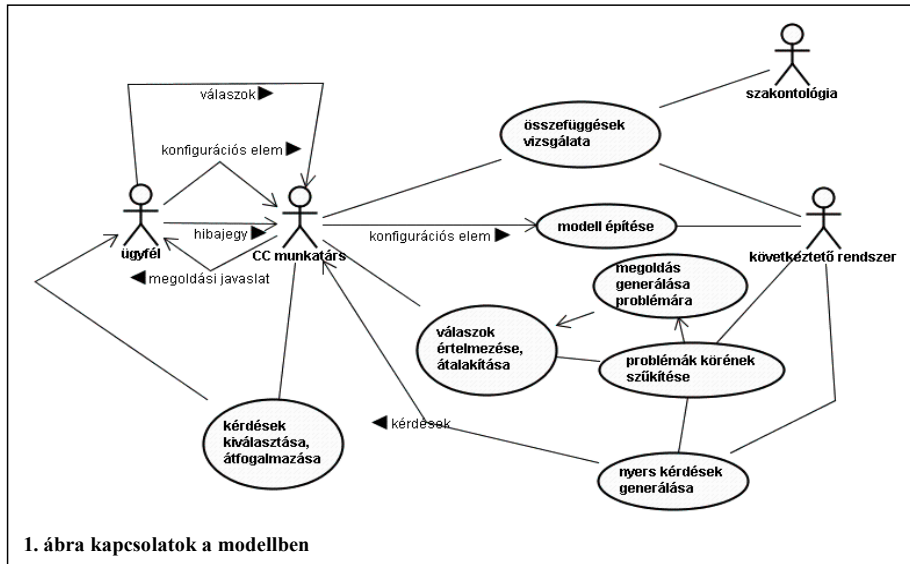
A CC munkatárs által tett kijelentések és az ebből következtetett lehetséges [Problémák](#), valamint kérdésésre érdemes [Teszt](#)ek, [Hibajelek](#), [Konfigurációs feltételek](#) együttese.

## Szerepek

**Ügyfél** – olyan ember, aki a Call Centertől/Contact Centertől várja problémája megoldását.

**CC munkatárs/felhasználó** – egy képzett ember, aki képes kezelni a hibakezelést támogató rendszert és segít elhárítani/megoldani az ügyfél problémáját.

**Távközlési hibakezelést támogató rendszer** – a CC munkatársat segítő rendszer, ezt a rendszert írja le ez a dokumentum.



## Részek

A rendszer az alábbi főbb részekből áll:

- Ontológia szerkesztő
- Következtető rendszer
- Ontológia
- Segédalgoritmusok
- Felhasználói felület a CC munkatárs és az ontológia/következtető rendszer között

A fejezet további részeiben ezen elemek funkcióinak leírása található.

## Ontológia szerkesztő

Az ontológia szerkesztéséhez két eszközt használnánk. Mivel az adatmodellünk frame-eket használ, így frame-szerkesztő rendszereket választottunk erre a feladatra. A kezdeti adatfeltöltéshez a Scriptum Informatika Rt. Metatár szerkesztőjének egy módosított változata, míg a későbbi frissítésekhez a Protégé szerkesztő frame-es modulja lenne használható ([xml-es adatsere formátum](#) használatával).

A szerkesztő a következő funkciókkal rendelkezik:

- Keresési lehetőség a fogalmak és kapcsolataik között.
- Teljes szövegű keresés a megjegyzésmezőkben.
- Naplózza a szerkesztés közben végzett lépéseket.
- Jogosultsági szintekhez köti a változtatási lehetőségeket.
- A szerkesztendő résznek megfelelő felület. (Tehát például a hierarchia kialakításakor nem mutatja a frame-ek kapcsolatait.)
- Az ontológia állapota alapján bizonyos helyeken felajánlja a prefixek alapján a lehetséges kiegészítéseket.
- Ontológia elmentése, betöltése
- Ontológia exportálása, importálása a [Protégé frame leíró xml formátum](#)ába.

### **Következtető rendszer**

A következtető rendszer képes kell a következő feladatok ellátására:

- Betölteni egy frame modellt (a neki megfelelő szintaxis szerint; az alább definiált összefüggések leírásához elegendő bonyolultságú állításokat tölti csak be)
- Betöltés után is lehessen kijelentéseket tenni
- Egy frame egyedeinek kilistázása
- Frame leszármazottainak illetve őseinek meghatározása (eldönteni, hogy két frame milyen viszonyban áll egymással; frame-ek között lehet többszörös öröklődés)
- Meghatározott kapcsolatok mentén történő keresés (pl.: slot érték egyezések mentén azonos őszű frame-ekben)
- Új kapcsolatok létesítése Horn szabályok alapján (pl. a generikus hierarchia lezárásakor, illetve az egymással kompatibilis software-eket tartalmazó klikkek meghatározásakor)

Nem feladata azonban az ontológia ellentmondásmentességének biztosítása. Így ez egy kellően egyszerű és hatékony alrendszer lehet rendszerünkben.

### **Ontológia**

Az ontológia leírását az A hibakezelési ontológia konszolidálása c. dokumentum tartalmazza, itt csak néhány rövid megjegyzéssel utalunk a tervezett tulajdonságaira:

- Nem tartalmaz metaosztályokat, bár – a Problémák és Hibajelek hierarchizálásából fakadóan – szükség van olyan relációkra is, melyek nem két egyed között futnak, hanem osztályok közt, ill. egy osztály és egy egyed közt; így a „standard” elsőrendű logikában való leírás nem kivitelezhető
- Frame alapú reprezentáció
- Bizonyos helyeken zárt, míg másutt nyílt világszemléletet alkalmaz.
- A főbb osztályai: Probléma, Hibajel, Megoldás, Teszt, Konfiguráció

### **Segédalgoritmusok**

Ezek az algoritmusok szolgálják a modell konzisztenciájának megőrzését. Az alábbi algoritmusokat készítjük el a rendszerhez, melyek a különböző állítások utáni modelljavításokat végzik el.

- Egy Teszt eredményétől függően gyanú változtatása a modellben, illetve néhány elem törlése a gyanús problémák közül.
- Egy Hibajel ellenőrzését követően a Hibajel meglététől függően bizonyos Problémákkal kapcsolatos gyanút növeljük, vagy töröljük a modelltől a kizártakat.
- Egy Konfiguráció komponens meglététől függően szintén változtatja bizonyos Problémák gyanú-értékét, vagy töröl néhány Problémát a modelltől.

## **Felhasználói felület**

A felhasználói felület a CC munkatárs és a rendszer többi része közti kommunikációt biztosítja. Főbb feladatai alapján az alábbi részekre bontható:

- Bőngésző a szakontológiához
- A lehetséges problémákat böngésző felület
- Konfiguráció pontosító felület
- Eldöntendő kérdések (konfigurációk) böngészése
- Lehetséges megoldási módok böngészése + állítás lehetősége
- Eldöntendő kérdések (tesztek eredményei) böngészése + állítás lehetősége
- Eldöntendő kérdések (hibajel fennáll-e) böngészése + állítás lehetősége

Ezek a felületek egységes képet mutatnak, könnyen lehet váltani köztük. (Kivéve az ontológia szerkesztő eszközt, mivel az egy külön komponens.)

A felhasználói felület a Scriptum Metatár termékének egy továbbfejlesztett változatát fogja használni.

## **Működés**

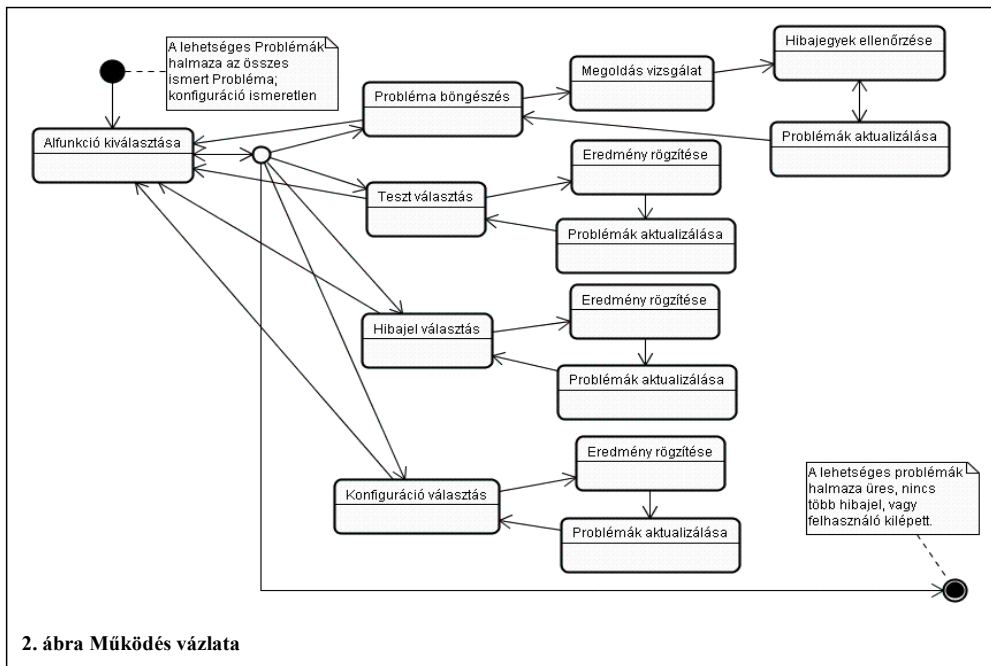
Ebben a fejezetben – egy rövid általános működési leírás után – a részrendszerek működését írjuk le alaposabban.

### **Általános működés**

Az alkalmazás használatának elképzelt forgatókönyve a következő:

- I. az ügyfél elmondja a CC munkatársnak, hogy milyen hibajelenségeket tapasztalt (ezeket a felhasználó átfordítja Hibajelekké) és valamennyit a saját rendszerének konfigurációjáról (melyekből Konfiguráció komponenseket készít a felhasználó)
- II. a rendszer az információk alapján leszűkíti a Problémák lehetséges körét (első cél: a Probléma minél pontosabb felismerése)
- III. ha a Probléma kellően meghatározott, létezik hozzá (ismert) Megoldás, a felhasználó kiválaszt egy ilyet, akkor a program kiajánlja a Megoldásokat (ha valamelyik Megoldást kipróbálta az ügyfél (természetesen a felhasználó fordításában és az ő javaslatára), akkor az eddig tapasztalt Hibajegyekre a IV. a.-nak megfelelően újra rákérdezzük); egyébként folytassuk a IV. ponttal
- IV. a program a további szűkítés céljából az alábbi dolgok valamelyikét teszi:
  - a. rákérdezhet egy olyan Hibajelre, ami legalább egy potenciális Probléma esetében felmerül, és még nem ismert, hogy fellépett-e a Hibajel. Amennyiben ez a Hibajel nincs, úgy az adott Problémákat ki lehet zárni; ha van, akkor „erősödik a gyanú”, hogy ezen Problémák valamelyikével állunk szemben.
  - b. kérheti egy bizonyos Teszt végrehajtását a felhasználótól; a tesztek sajátossága, hogy egy-egy Probléma vagy Problémaosztályra vonatkozóan sikeres végrehajtásuk egyértelműen megállapítja, hogy az adott Probléma(osztály) valamely eleme fellép-e vagy sem.
  - c. rákérdezhet a felhasználó az ügyfél Konfigurációjának valamely tulajdonságára, mely tulajdonság szükséges egy nem-kizárt probléma fennállásához.
- V. Ezek után menjünk vissza a II. pontra.

A főbb átmeneti lépéseket mutatja a következő ábra.



## Ontológia szerkesztő

Az ontológia szerkesztő leghasznosabb tulajdonságai a helyes működés és a keresési lehetőség, így ezekre különösen nagy hangsúlyt fektetünk.

A könnyű kezelhetőség érdekében a szerkesztő támogatja a kontextus-függő keresési lehetőséget, valamint a szakontológia böngészésekor a megjegyzésekbeli teljes szövegű keresést.

Célszerűnek tűnik egy prototípus készítése és az alapján a felhasználók igényeihez igazítani a rendszert.

## Következtető rendszer

Ez a rész az ontológia kezelésének egy kulcsa. Mivel csak egyszerű következtetési lépésekre lesz szükségünk, leginkább a keresés hatékonysága a döntő szempont.

A specifikáció ezen fázisában korai lenne elköteleznünk valamilyen következtető rendszer mellett, valószínű, hogy egy adatbáziskezelő rendszer és egy, a hierarchiát kezelő rendszer ötvözése egy kellően hatékony megoldás lenne erre a feladatra. (Hisz a lekérdezések vannak túlsúlyban és a következtetések többségét a generikus hierarchiából származó tudás felhasználásával nyerjük. A kapcsolatok mentén vett kereséseket szintén támogatja az adatbáziskezelő.)

Az eddig vizsgált rendszerek közül a rule engine-ek, a Prolog alapú rendszerek (pl. FLORA-2), és azon tételbizonyítók, melyek támogatják a felsorolást is, egyaránt megfelelőek erre a feladatra, ám kétséges, hogy a viszonylag jelentős mennyiségű konvertálás és a Java interface hiánya miatt nem egyszerűbb-e egy ehhez a rendszerhez illő következtető rendszert megvalósítani az alábbi módon:

1. A hierarchia és a kapcsolatok betöltése

2. Lekérdezés esetén a hierarchiákat kezelő modul meghatározza, hogy mely frame-ek illenek a lekérdezés bizonyos pontjaira (melyek azok, melyek általánosabbak), majd az adatbáziskezelő az ennek megfelelő egyedeket szolgáltatja.
3. Ha kapcsolatokat is meghatározott a lekérdezés, akkor a kapott lehetséges kötéseket mentén megvizsgáljuk, hogy mely elemek vannak valóban összekötve. (Szintén adatbázis lekérdezésekkel.)
4. Amennyiben egy szabályt kap a következtető rendszer, akkor a 2. és 3. pontnak megfelelően kikeresi azokat az elemeket, melyekre illik a feltétel és a szabály eredmény része alapján újabb kapcsolatokat alakít ki (az adatbázisban). (Tehát lényegében egy hátra következtető rendszerről van szó, mely tablózza a következtetés eredményeit.)

## Segédalgoritmusok

Célunk a következtető rendszer funkcióinak meghatározásakor természetesen az is, hogy minél pontosabban működjön, ám elsődlegesen azt tartjuk fontosnak, hogy gyorsan szolgáltatson válaszokat. Korábbi vizsgálataink tanulsága az, hogy minél nagyobb kifejezőerőt adunk egy következtető rendszer input nyelvének, annál lassabban működik (ez természetes: a nagyobb kifejezőerő nagyobb állapotteret generál, amiben az inkonzisztencia ellenőrzése akár exponenciálisan is tovább tarthat, sőt az eldönthetlenségig is elviheti a rendszert). Ezért úgy döntöttünk, hogy magának a következtető rendszer magjának, a motornak egy csekély kifejezőerejű nyelvet adunk (ami megfelel körülbelül a Horn-logikának, leszámítva, hogy itt osztályok is szerepelhetnek relációk mindkét oldalán). Az ezzel a motorral előállított választ pedig tovább szűrjük olyan algoritmusok segítségével, melyek kívül állnak a következtető motor képességein, azonban szintén gyorsak. Így a következtető motor és külső szűrők segítségével elérhetjük a megkívánt teljesítményt, miközben a válaszok köre nem változik – ezt annak köszönhetjük, hogy a válaszok ebben a speciális szakontológiában valóban megkaphatóak ezzel a sémával. Azt a keretalgoritmust, mely összefogja ezeket a speciális algoritmusokat, egy korábbi szakaszban bemutattuk.

Az algoritmusokban a lehetséges Problémák halmazát  $P$ -vel, míg annak elemeit/valamely elemét  $p$ -vel jelöljük.

## Szűrés Hibajel alapján

Válasszunk egy  $h$  Hibajelet, amire van olyan elem a  $P$  problémaosztályban, ami mindenképp rendelkezik vele vagy leszármazottjával, és olyan is, ami nem.<sup>1</sup> (A felhasználó böngészheti a potenciális Hibajeletet anélkül, hogy választania kellene közülük.) Kérdezzünk rá erre a  $h$  Hibajelre<sup>2</sup>.

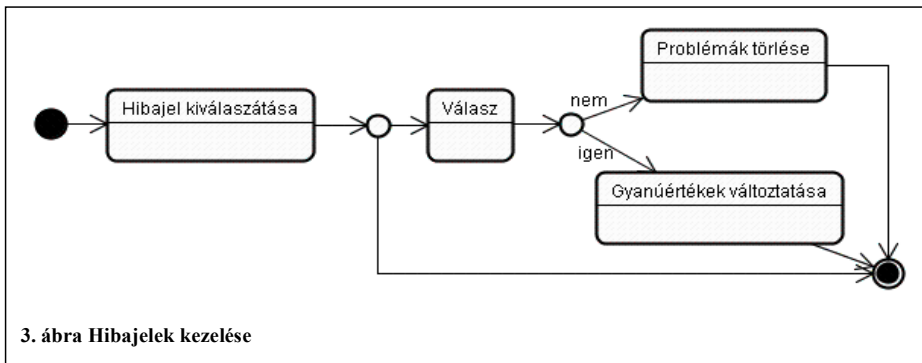
1. Ha a válasz „igen”, akkor erősítsük meg gyanúnkát<sup>3</sup> minden olyan  $p$ -vel szemben, aminek valamely felmenője rendelkezik ezzel a  $h$  Hibajellel vagy leszármazottjával.
2. Ha a válasz „nem”, akkor távolítsuk el  $P$ -ből az összes olyan problémát, melynek van olyan felmenője, aki rendelkezik  $h$  valamely leszármazottjával.

<sup>1</sup> Természetesen a felhasználó választhat másikat is, azonban a rendezés esetén célszerű így rendezni a Hibajeletet.

<sup>2</sup> Itt (és később is) a „kérdés” kétféle jelentésben szerepel. A felhasználó (CC munkatárs) felé egy számára érthető, nem feltétlenül természetes nyelvű szöveg/ábra, míg az ügyfél felé a munkatárs ebből egy érthető kérdést készít.

<sup>3</sup> Itt és a későbbiekben is a gyanú értékének változtatásának pontos algoritmusosa még nem alakult ki: a kész rendszer vizsgálatokor majd néhány próba alapján döntjük el, majd az összes adattal feltöltött rendszerben további finomításokat végzünk.

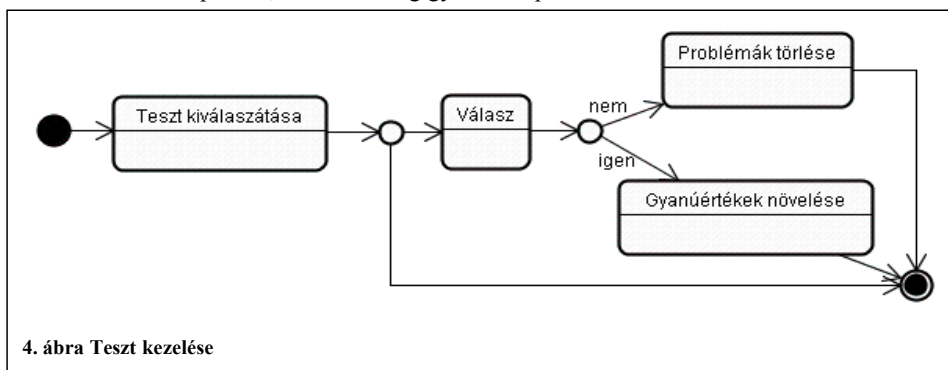




### Szűrés Teszt alapján

Válasszunk egy olyan Tesztet, mely lehetséges (és gyanús) Problémához kapcsolódik. Ezt hajtassuk végre az ügyféllel. (A Teszteket ettől függetlenül is vizsgálhatja a felhasználó. Azonban a rendszer nem képes szimulálni egy-egy teszt következményeit (modem bekapcsolva a teszt végén, illetve visszatérni egy olyan állapotba, mintha nem végezte volna el az ügyfél a tesztet).)

1. Ha a Teszt negatív, távolítsuk el  $p$ -t és összes leszármazottját  $P$ -ből.
2. Ha a Teszt pozitív, erősítsük meg gyanúnkat  $p$ -vel és leszármazottaival szemben.

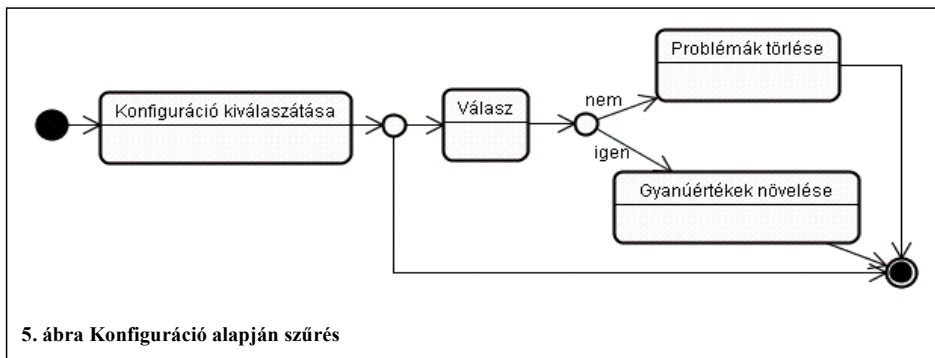


### Szűrés Konfiguráció alapján

Válasszunk egy olyan  $E$  Konfiguráció-komponenst, amire van  $P$ -ben olyan Probléma, ami igényli egy ilyen komponensnek a jelenlétét, és egy olyan is, ami nem<sup>4</sup>. Ez a komponens lehet osztály is, vagy két már meglévő Konfigurációs elem közti kapcsolat. Kérdezzünk rá  $E$ -re.

1. Ha a válasz „igen”, akkor erősítsük meg gyanúnkat minden olyan Problémával szemben, ami igényli  $E$  (vagy ha  $E$  osztály, akkor valamely leszármazottja) jelenlétét. Menjünk a 2. pontra.
2. Ha a válasz „nem”, akkor töröljünk minden olyan elemet  $P$ -ből, ami igényli  $E$  vagy leszármazottja jelenlétét.

<sup>4</sup> Legalábbis ilyen célszerű választani, a rendezés is ezt követi. Természetesen választható olyan is, mely nem teljesíti ezt a feltételt.

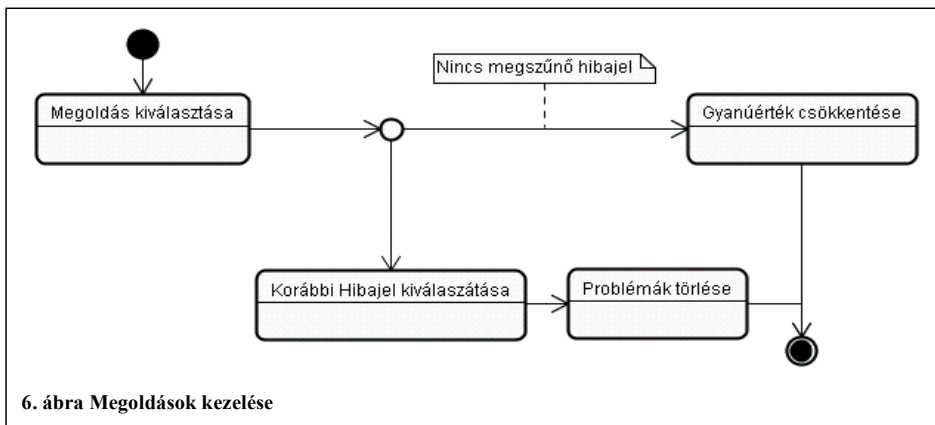


5. ábra Konfiguráció alapján szűrés

### Megoldás végrehajtása után szükséges lépések

Válasszuk ki egy  $p$  elemet a  $P$  problémaosztályból, amihez van csatolt Megoldás. ( $P$  – vagy leggyanúsabb elemei – böngészhetőek.) Hajtassuk végre a Megoldást az ügyféllel. Ellenőriztessük az ügyféllel, hogy az eddig fennálló Hibajelek továbbra is jelen vannak-e.<sup>5</sup>

1. Ha van olyan Hibajel, ami emiatt megszűnt, hajtsuk végre a Szűrés Hibajel alapján részt az összes ilyen Hibajelre (azaz töröljük a megfelelő problémákat).
2. Ha nincs ilyen Hibajel, akkor csökkentjük gyanúnkat  $p$ -vel és leszármazottaival szemben.



6. ábra Megoldások kezelése

### Felhasználói felület

A felhasználói felület korábban említett részeit mutatjuk be ezen fejezet szakaszaiban. Mivel a felhasználó ezen keresztül kommunikál, ezért ehhez kapcsolódóan nem a pontos leírást, képernyőtervet mellékeljük, hanem azt határozzuk meg, hogy az egyes elemek milyen lehetséges funkcionalitással bírhatnak, hogyan tervezzük a működését.

### Szakontológia böngésző

Első megközelítésben egy, a következtető rendszerrel közvetlen kommunikációt lehetővé tevő interface-t elegendőnek tartunk. Azonban az ontológia megváltoztatására lehetőséget adó lekérdezéseket nem engedjük meg.

<sup>5</sup> Feltételezzük, hogy megoldás végrehajtása nem hozhat létre új hibákat, így nem kérdezzünk olyan Hibajelekre, melyek korábban nem álltak fel.

A válasz-elemeket szintén a következtető rendszer kimenetei adják, nem végzünk rajta módosításokat.

Valószínű, hogy ez a felületet nem gyakran használja a felhasználó, hisz az ontológia nem kell érdekelje. Ennek a felületnek a célja bizonyos ritka/kevésbé ismert információ keresése. Ha erre nincs szükség, akkor a végleges változatba nem kerül bele.

### **Ügyfél környezetét leíró eszköz**

Ez az alrendszer a Metatárat használja az ügyfél konfigurációjának pontosítására.

A felhasználó állíthatja tetszőleges elemről, hogy létezik-e a modellben, vagy sem, valamint amennyiben létezik, akkor a működéséről, jellemzőiről is nyilatkozhat. Az elemek a következők lehetnek:

- eszközök (hardver és szoftver);
- eszközök közti kapcsolatok.

A működés vázlatos leírása:

1. Néhány kérdést feltesz a rendszer, melyet a felhasználó – az ügyféltől kapott információk alapján – megválaszol.
2. Ezen információk alapján a felhasználó kiválasztja a megfelelő konfiguráció vázat a lehetséges változatok közül, mely legközelebb áll a felhasználó konfigurációjához.
3. Későbbi lépésekben a felhasználó pontosíthatja az ügyfél konfigurációjára vonatkozó ismereteit, azonban új kapcsolatokat nem adhat meg és nem is törölhet, valamint csak speciálisabb elemet adhat meg a korábbi helyett. (Tehát egy böngészőt nem cserélhet le egy vezeték nélküli kapcsolatra a modellben, azonban az általános böngésző osztályt lecserélheti az Internet Explorer 6.0 változatra.)

### **A lehetséges problémákat böngésző felület**

Ez egyszerűen a szűkítéssel kapott problémák rövid leírását takarja. Szintén a Metatár felületet használja a böngészéshez, kereséshez.

Egy adott Probléma kiválasztása esetén annak további tulajdonságait is bemutatja. A rendezés alapértelmezetten a gyanú alapján történik. Hierarchikusan is böngészhető a lehetséges Problémák halmaza, de lehet a megjegyzések mezőkben teljes szövegű keresést is végezni.

Lehetséges rendezési feltételek:

- Gyanú
- Hierarchia
- Illeszkedő Hibajegyek száma szerint
- Szükséges Konfigurációs feltételek száma szerint (melyekről tudjuk, hogy szükségesek és rendelkezik is vele a modell)
- Megoldás (első) leírása
- Megjegyzés szövege

Indexelt mezők:

- Megjegyzés
- Konfiguráció
- Hibajegy
- Megoldás

## Konfigurációs kérdések böngészése

A Metatár felület egy újabb alkalmazása, mely a konfigurációval kapcsolatos nem-vizuális kérdéseket mutatja meg.

Az eszközök közti kapcsolatra kérdez. (Esetleg az – egyébként pusztán háttérképként funkcionáló, az ügyfél rendszerét jó közelítéssel modellező (előre megrajzolt) – grafikus ábrán is jelöli a kérdést. Ez esetben ez a rész csak a rendezés miatt lehet érdekes.)

A rendezés alapértelmezett szempontja az adott kérdéssel eldönthető problémák mélységének (súlyozott) összege alapján történik.

További lehetséges rendezési szempontok:

- A konfigurációs kérdéshez tartozó Problémák gyanú átlaga
- A konfigurációs kérdéshez tartozó Problémák száma
- A hozzá tartozó Problémák mélységéből képzett súlyozott összeg
- A konfigurációs kérdések esetén jelen lévő hierarchia esetében a hierarchia felső szintű komponenseire kérdezzük rá előbb

## Tesztek megjelenítése

A Metatár segítségével a tesztelemelek rövid leírását mutatja, kiválasztás esetén a részletesebb leírást mutatja. (Az esetleges még részletesebb leírások linkeként vannak benne. A hypertextes szöveget egy böngésző jeleníti meg.)

Végrehajtás esetén [Igen] [Nem] [Mégsem] elemek közül kiválasztja a felhasználó az eredményt.

Lehetséges rendezési szempontok:

- Legfeljebb mennyi Problémát zárhat ki
- Kizárható Problémák mélységének súlyozott összege

## Hibajelek megjelenítése

A Metatár felületen tetszőleges Hibajel meglétéről nyilatkozhat a felhasználó (természetesen az ügyféltől kapott információk alapján). Böngészhető hierarchiák. Intelligens (szinonimákat (kompatibilis elemek) és fölérendelt címszavakat hozzátéveszi) kereső. Rendezés szempontja változtatható. (Pl.: gyanú alapján, vagy hierarchia szerint is böngészhető.)

Ehhez a részhez is hasznos az ontológia használata. (Például a keresőkifejezések meghatározásához.)

Lehetséges rendezési szempontok:

- Kizárható Problémák száma
- Kizárható Problémák mélysége alapján képzett súlyozott összeg
- Kizárható Problémák gyanújának átlaga
- A hierarchia felső szintű elemeit vegyük előre
- Az ismert eredményű Hibajeleket csoportosítsa külön (vagy esetleg ki se írja)

## Metatár

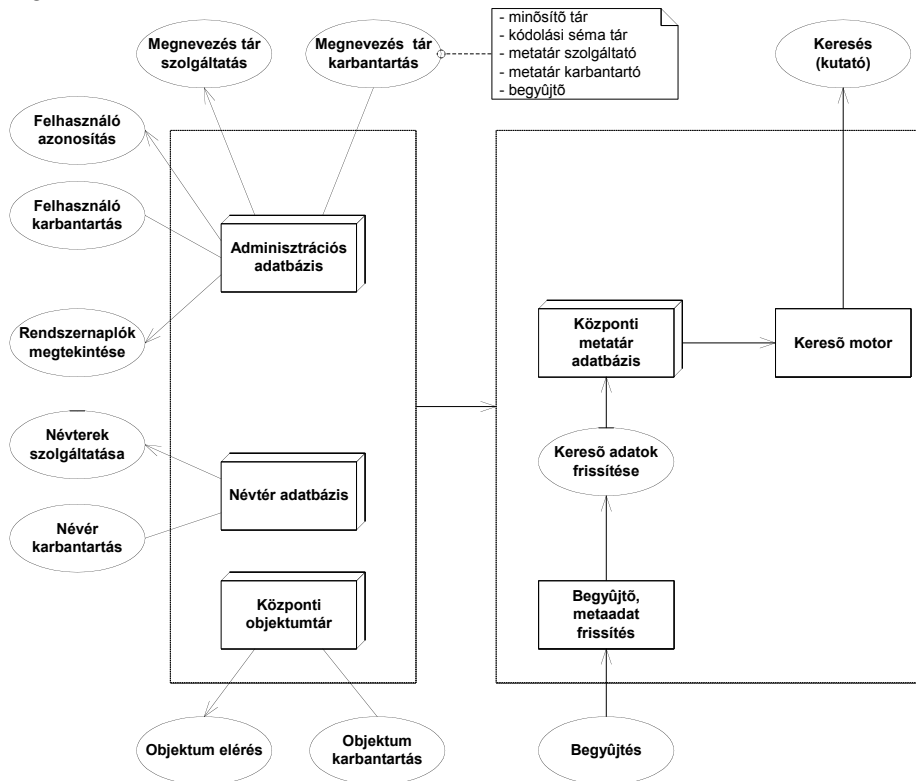
Ezt az eszközt a következő szerkesztői modulokban használjuk:

- Ügyfél környezetét leíró eszköz
- Probléma-böngésző
- Teszt megjelenítő
- Hibajel megjelenítő
- Ontológia szerkesztő

A következtető rendszer, a segédalgoritmusok egy olyan adatszerkezeteket használ, amely eltér a metatár adatábrázolásától. A Metatár biztosítja a szinkront a többféle adatábrázolás között. A Metatár a többi rendszer kimenetét is felhasználhatja a megjelenítéskor, szerkesztéskor.

## Központi szolgáltatások

Az alábbi ábra a központi szolgáltatásokat szemlélteti. Bal oldalon az általános központi szolgáltatások, jobb oldalon az adatbegyűjtéssel és kereshetővé tétellel kapcsolatos szolgáltatások láthatóak.



7. ábra: Az egyedi metatárak számára központi szolgáltatások: adminisztráció, névtér, objektumtár. Az adatbegyűjtés és kereshetővé tétel szintén a központ feladata.

A továbbiakban részletesen tárgyaljuk az egyes központi szolgáltatásokat.

## **Felhasználó adminisztráció és a jogosultságrendszer**

A rendszer felhasználói központosítva kezeltek. Felhasználók adatainak adminisztrálása, új felhasználók felvétele biztosított.

Felhasználók jellemzői:

- Felhasználói azonosító.
- Teljes név.
- Jelszó.
- Jogosultság (lásd lejjebb).
- Státusz. („Aktív”, „Inaktív”, „Törölt”)

A felhasználói jogosultságok megadására alapvetően metatár szinten lehetséges, szükség esetén ezt a metatár séma egy-egy minősítője szintjére finomítva.

## **Központi megnevezés és konfiguráció tár**

A központi megnevezés tár alapvető szerepe, hogy a rendszer elemeit egyedi módon azonosítsa, és a kapcsolódó központi adminisztratív feladatokat támogassa. Emellett kapcsolódó konfigurációs információkat is tárol.

- Minősítő tár.
- Kódolási séma tár.
- Metatár szolgáltatók azonosítása és konfigurációja.
- Metatár karbantartók azonosítása és konfigurációja.
- Begyűjtők azonosítása és konfigurációja.

A fenti elemek egy-egy csoporton belül egyedi rövid névvel kell, hogy rendelkezzenek.

## **Központi naplózás**

A központi naplózás a rendszer központi szolgáltatásaival kapcsolatos aktivitásokat tárolja.

- Bejelentkezések.
- Begyűjtések kezdete, vége, begyűjtött adatok mennyisége.
- Központi megnevezés tárhoz kapcsolódó adminisztráció naplózása.
- Konfigurációs tárhoz kapcsolódó naplózás.
- Felhasználó adminisztráció naplózása.

A központi naplózási funkció mellett az egyedi metatárak önállóan végzik el a metatár szintű naplózást.

## **Névterek**

A névterek speciális metatárak. Karbantartásuk a metatárakhoz hasonló módon, hasonló kliens felületen történik.

Névtér szerepe:

- Egyértelműsíti az egyedeket, id-ket rendelhet hozzájuk.
- Névteres keresés során segítséget ad a megfelelő egyed kiválasztásához.
- Navigációs lehetőséget biztosít.

A navigáció átláthatósága, világos megvalósítása érdekében a névterek „fa” struktúrájúak.

## Objektumtár

Az objektumtár tekinthető egy speciális metatárnak is, melyben nem csak metaleírások (objektum leírások) szerepelnek, hanem ahhoz csatolt objektumok is.

Az objektumtár a központi szolgáltatások részét képezi. Az objektumtár elemei az alábbi „párok”:

objektum leíró, az objektum metaleírása.

objektum (fájl).

Megjegyzés a központi objektumtár használatával kapcsolatosan:

Az objektum leírók megfelelő tartalmú kitöltése javasolt, a könnyebb és tisztább hivatkozások létrehozásának segítésére.

Az objektumtárban az objektumok duplikátummentessége nem garantálható, ennek elkerülése alapvetően az objektumtár – szerkesztő és véglegesítő jogosultságú – felhasználóinak felelőssége.

Új objektumtári elem beszurásakor a rendszer figyelmeztető üzenetet ad, ha az alábbi szempontok szerint azonosnak vélt elemet talál az objektumtárban:

Fájlnév és kiterjesztés,

Fájl méret.

## Funkciók

Az objektumtár alapvető funkciói az alábbiak:

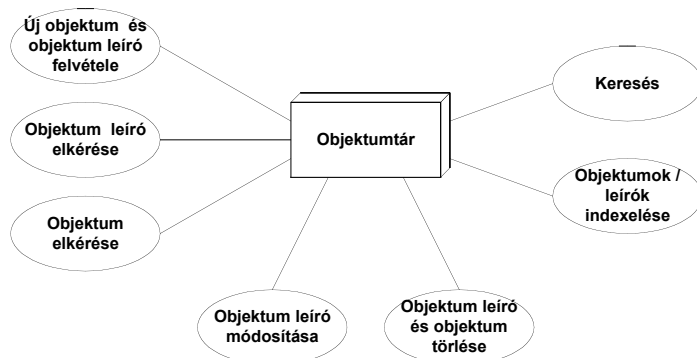
Új objektum és objektum leírás felvétele

Létező objektum törlése az objektumtárból.

Objektum leírás módosítása.

Objektum vagy objektum leírás elkérése.

Indexelés, keresés az objektumtárban tárolt leírásokban és az objektumokban.



8. ábra: Az objektumtár és a hozzá kapcsolódó fő funkciók. Az objektumtárban objektum leírók és objektumok párijai szerepelnek.

Az objektumtárban tárolt elemekhez szorosan kapcsolódó feladatok és funkciók:

Szövegiemelés objektumból, melynek hatására a szöveges objektumból (.doc, .pdf, ...) formázatlan szöveg kiemelésre kerülhet. Ez alapján készül a dokumentumok tartalmi indexelése (full-text kereséshez).

Indexelés, mely hatására indexek készíthetők az objektum leíróban tárolt értékekre (megadható, hogy mely jellemzők értékeire készüljön index) és az objektumokból kiemelt szövegekre (full text).

Az index készítésekor szótó képzés megtörténik.

Keresés, a képzett index alapján történik.

A találati lista megjeleníthető, a lista elemei megtekinthetők.

A találati „lista” tartalmazhat:

egy objektum leírot (id alapján történő objektum leíró keresése)

egy objektumot (id alapján történő objektum keresése)

objektum leírók egy listáját (más feltétel alapján történő keresés)

Az objektumtár esetén konfigurálható, hogy mely kiterjesztésű állományok feltöltésére legyen lehetőség.

#### **Objektumtár karbantartás**

Az objektumtár karbantartása alapvetően az objektumtár karbantartó felületén történik.

Emellett új objektumtári elem felvételére lehetőség van az egyedi metatárak karbantartó klienséről kezdeményezve is. Például ha az aktuálisan szerkesztett metarekordhoz egy dokumentumot szeretnénk csatolni, akkor azt a karbantartó felületről – jogosultság esetén – hozzá kell, hogy tudja adni a felhasználó.

Amennyiben egy olyan felhasználó bővíti az objektumtárat, aki szerkesztő jogosultsággal rendelkezik, de jóváhagyó jogosultsággal nem, akkor a bővítés megtörténik, de az objektumtári elem nem „véglegesítődik”. Ekkor a szerkesztett elem hivatkozik az újonnan beszűrt objektumtári elemre. Amennyiben a jóváhagyó jogosultságú felhasználó elveti az objektumtár bővítési javaslatot, akkor az objektumtári elem törlődik és így a szerkesztett elem nem valódi objektumtári hivatkozást tartalmaz. Az invalid hivatkozást a következő ellenőrzési folyamat törli, így helyreáll a konzisztencia.

#### **Objektum leíró**

Az objektum leíró az alábbi adatokat tartalmazza:

Azonosító (objektumtáron belül egyedi).

Objektum neve (fájlnév)

Kategória (mappa), mely az objektum csoportosítását teszi lehetővé.

Többszörözhető jellemző, egy objektum több kategóriába is besorolható.

Egyszintű csoportosítást tesz lehetővé.

Megjelenítendő név.

Objektum típusa (MIME).

Objektum mérete.

Objektumtári elemet beillesztő felhasználó neve és a bekerülés dátuma.

Objektumtári elemet jóváhagyó felhasználó neve és a jóváhagyás dátuma.

Megjegyzés.

#### **Központi keresés**

Web-es felületű keresésre a központi metatárban tárolt adatokban van lehetőség. A feladatot a központi metatárhoz kapcsolódó kereső motor és a hozzá kapcsolódó felhasználói felület látja el.



A rendszer az alábbi központi keresési lehetőségeket biztosítja:

Keresés a begyűjtött, központi keresőtárban tárolt metarekordokban.

Keresés lehetősége az objektumtárban tárolt dokumentumokban.

A kereséshez kapcsolódó speciális lehetőségek:

Keresés során lehetőség van névtér támogatás használatára.

Keresés történhet minősítőre történő szűkítéssel, elemre történő szűkítéssel vagy szűkítés megadása nélkül.

Névteres keresésénél, ha hierarchikus névtér szerepel a háttérben, akkor a keresés során a hierarchia kihasználásra kerül (egy elem megadása esetén annak gyermekeiben is keres).

Megjegyzés:

Központi keresés során a begyűjtött metarekordok megjelenítésekor a minősítőknél a minősítő tárból definiált „megjelenő nevet” használjuk az adatmegjelenítés során. Ez eltérhet attól a névtől, mely a karbantartó alkalmazásban használt, amennyiben a karbantartó alkalmazás szerkezetének definiálásakor az metatár specifikus módon felülírásra került.

Navigációs támogatás biztosítható a központi kereső felületen, melynek segítségével a kereső elemei (hierarchikus) struktúrában láthatók és annak segítségével elemek kiválaszthatók.

Meghatározandó egy névtérben egy kategória és egy névtér elem, mely a kiválasztott kategóriába tartozik. A navigációs fa gyökere az adott elem, és a navigációs fa a megadott kategóriába tartozó, adott elem alatti elemeket tartalmazza (hierarchikus névtér esetén).