

Networkshop'17

**3D modell alapú, statisztikai
lekérdezőnyelv és futtató környezet
parametrikus lekérdezések futtatására**

Király Roland, Balla Tamás,
Bózvály Bence

Célok

- ❖ A vizsgált terület a felsőoktatás, és a tudományos kutatások világa.
- ❖ Arra vagyunk kíváncsiak, hogy a rendszer résztvevői mely pontokon működnek együtt.
- ❖ Vizsgáljuk az összefüggéseket, kapcsolatokat és azt, hogy a munka hogyan tehető hatékonyabbá
- ❖ A probléma megoldása, vagyis a megfelelő adatbázis szerkezet kialakítása, a lekérdezések elkészítése és parametrizálása nem egyszerű feladat.

A problémák, amikbe ütköztünk

- ❖ Inhomogén, strukturálatlan adatok állnak rendelkezésre, nagy mennyiségben.
- ❖ A hagyományos, többnyire statikus lekérdező eszközök segítségével az adatbázisokban megbúvó, nagy mennyiségű információ megtalálása nehéz feladat
- ❖ A korábbi adatbázis kezelő rendszerek statikus szerkezetűek, az új eszközök és lekérdező nyelvek pedig túl általánosak
- ❖ speciális területekről kell információt gyűjteni.

A megoldás

- ❖ Szükség van egy grafikus felületre, megfelelő adatgyűjtési, szűrési, rendezési módszerekre és egy jól struktúrált adatbázisra,
- ❖ amelyek segítségével nagy mennyiségű adathalmazon kényelmesen, gyorsan , de professzionális módon lehet információt gyűjteni,
- ❖ Az eredményt 3D felületen, több nézőpontból és szögből is kell vizsgálni, értelmezni.
- ❖ Kompakt eszközt kell készíteni, amely az átlagos felhasználók számára is lehetővé teszi az elemzések futtatását.

A megoldás

- ❖ A célunk elérése érdekében kidolgoztunk egy adatlekérdező és feldolgozó rendszert, amely rendelkezik
- ❖ saját adatbázis modellel,
- ❖ tartozik hozzá saját lekérdező nyelv,
- ❖ grafikus felület, amely valós időben képes megjeleníteni a lekérdezések eredményeit.
- ❖ A program kimenete 3D-ben előállított, élcímkezett, irányított gráf, ahol a csomópontok és az élek egyértelműen (vizuálisan) leírják a vizsgálni kívánt információt

A modell három komponense

- ❖ Az első az információ tárolására használt adatbázis és a rajta értelmezett szemantikus információ
- ❖ A második komponens a szemantikus információ kinyerésére használható lekérdező nyelv
- ❖ A harmadik komponense a grafikus megjelenítő réteg, amely a lekérdezések eredményeit 3D gráf formában ábrázolja

1. Az adatbázis

Az információ tárolására használt adatbázist két logikai szintre osztható:

- ❖ *Az alacsonyabb szint a nyers adatok szintje:*
- ❖ Nyers adatnak nevezzük azokat az objektumokat leíró egyedeket, amelyek a kapcsolatok végpontjain helyezkednek el.
- ❖ Ilyen végpontok lehetnek pl. a kutatást végző személyek, a kutatott projektek, vagy egyéb, a kapcsolatok alanyát képző objektumok.
- ❖ Végpont lehet az eredmény is, ami egy további cikk, vagy szöveget tartalmazó objektum.

Az adatbázis

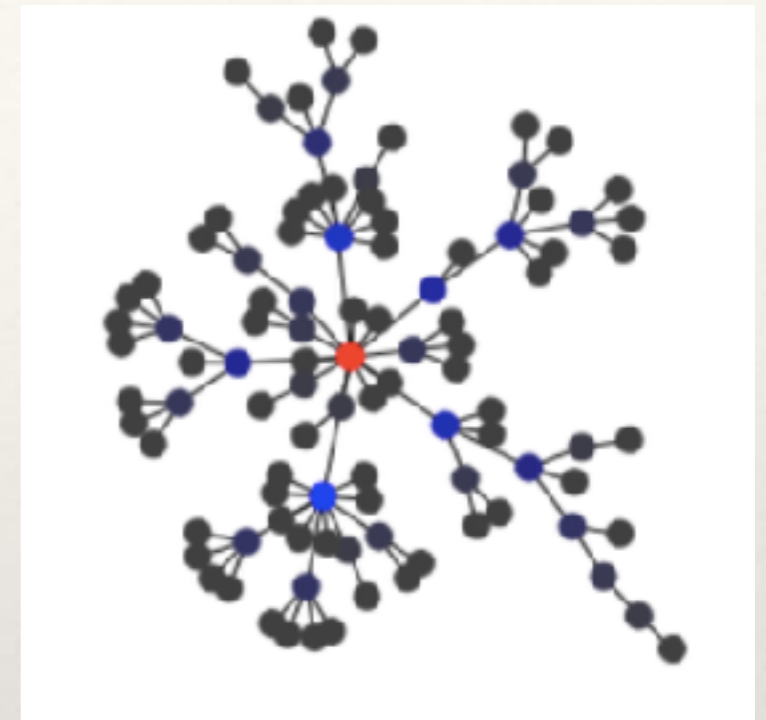
- ❖ *A második az összefüggések szintje, vagyis a kapcsolatoké (szemantikus információ):*
- ❖ A végponti objektumok közötti kapcsolatokat hivatott reprezentálni.
- ❖ Két végpont között a kapcsolat egy harmadik, összekapcsoló objektum segítségével valósul meg.
- ❖ A kapcsolat objektum több (kapcsolatokat leíró) attribútumát virtuális élek tekintjük a két végpont között.
- ❖ Az élek száma a virtuális élek erőssége (élkötegek),
- ❖ Az erősség függ a kapcsolat objektum attribútumainak a számától, és a kapcsolat erősségét adja.

Az adatbázis

- ❖ Az adatbázis nyers adatai, vagyis a végpontok és a rajtuk értelmezett kapcsolatokat leíró információk összessége a **szematikus gráf**, amelynek egyes részgráfjait, vagy a teljes rendszert ábrázolhatjuk 3D-ben

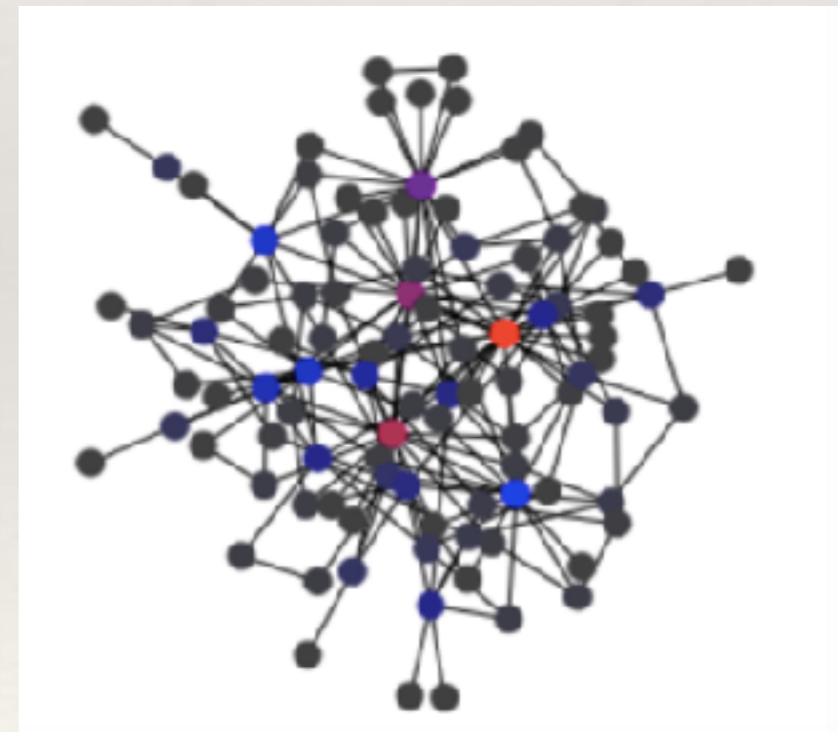
2. A grafikus megjelenítés

- ❖ Az adatok értelmezésének módszerét a vizuális megjelenítésre alapoztuk.
- ❖ Ez a réteg Java nyelven készült és használja a Graphstream csomagot.
- ❖ A gráf formázásához CSS fájlokat használtunk.
- ❖ A 3D eredményt egér segítségével lehet mozgatni, forgatni, kicsinyíteni és nagyítani a jobb átláthatóság érdekében.

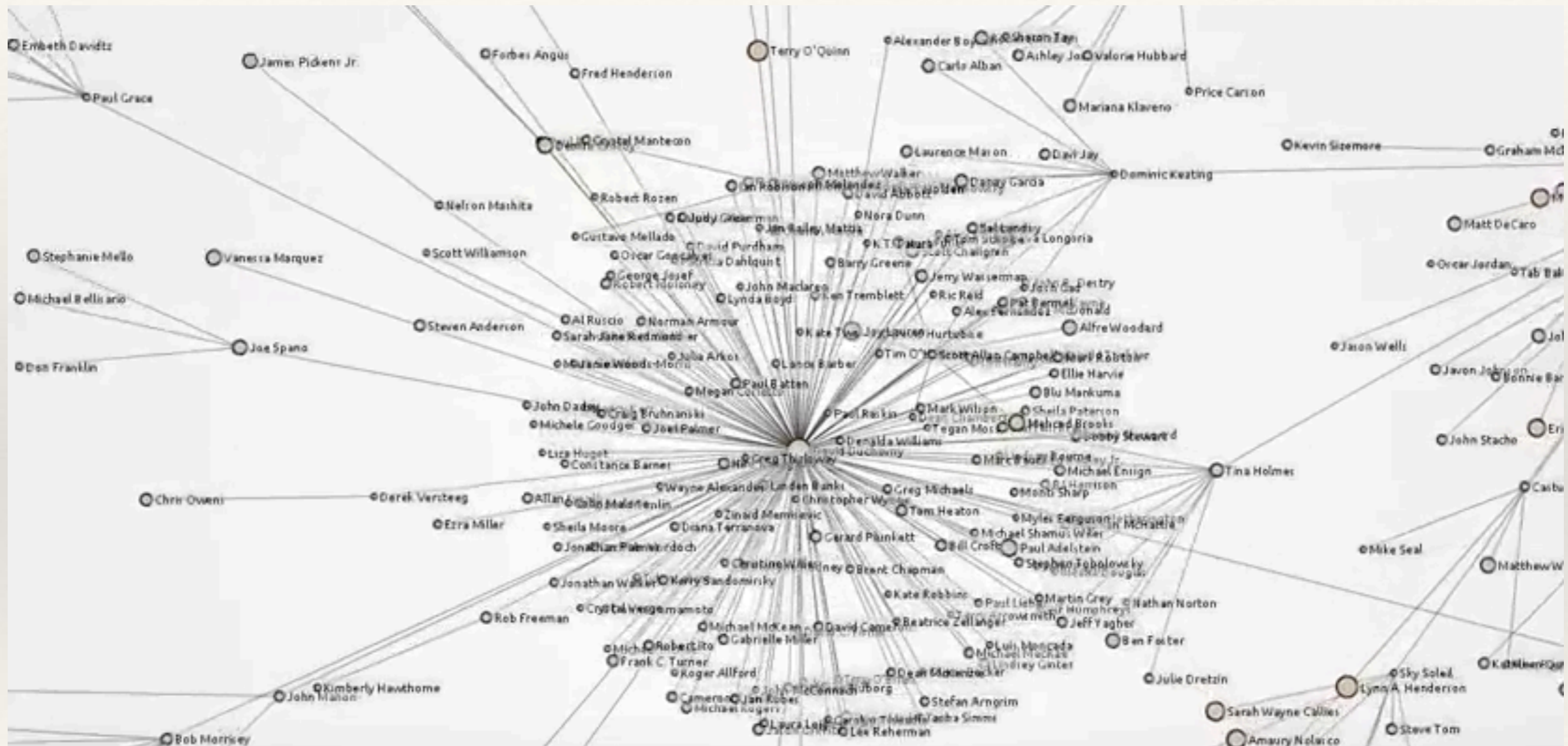


Szűrés, és kapcsolódó gráftípusok

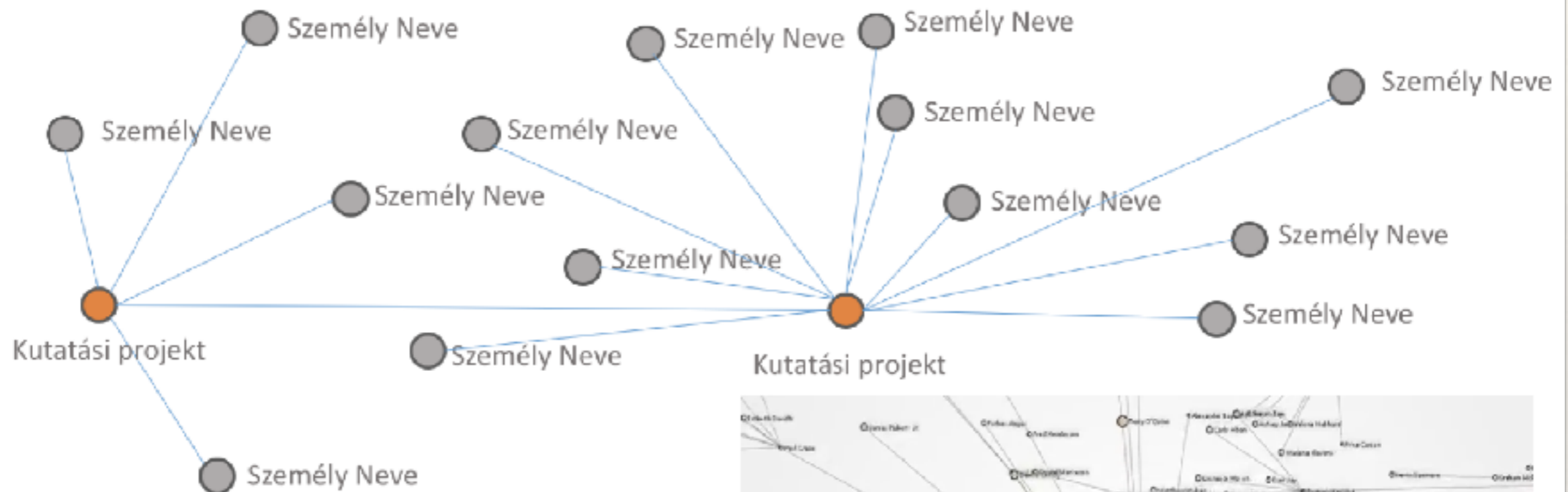
- ❖ Háromféle szűrési lehetőség áll rendelkezésre, szűréstípusokon belül több szinttel:
- ❖ listás szűrések gráf kimenettel,
- ❖ több szintű gráfbejárások,
- ❖ kapcsolati hálók megjelenítése és kifejtése



A grafikus megjelenítés



A grafikus réteg



3. A lekérdező nyelv

- ❖ A nyelv segítségével lehetőségünk van speciális, az adatbázis szintaktikai és szemantikai struktúráját kihasználó lekérdezések futtatására,
- ❖ ezáltal olyan adatok előállítására, amelyek a szemantikus gráf egy-egy részgráfját adják vissza a vizuális gráfmegjelenítő réteg számára.
- ❖ Az így kirajzolt gráfok vizsgálatával olyan információ mennyiséget tudunk átláthatóvá tenni, ami az adatok manuálisan történő, nem vizuális elemző módszerek használatával nem kezelhető.

Példák lekérdezésekre

- ❖ Az első példa lekérdezés segítségével megkapjuk két tetszőleges objektum közötti „közös cikk” típusú kapcsolatokat

```
show relation
```

```
between[object1, object2]
```

```
where relationtype is article
```

A lekérdezés eredménye SQL nyelven

```
SELECT * FROM Relations R INNER JOIN RelationTypes RT ON R.type = RT.id
    INNER JOIN Objects O1 ON O1.id = R.object1
    INNER JOIN (SELECT O.*, OA.*, ATT.*, A.*
FROM Objects O LEFT JOIN ObjectAttributes OA ON O.id = OA.object
    LEFT JOIN AttributeTypes ATT ON ATT.id = OA.attributeType
    LEFT JOIN Attributes A ON A.id = OA.attribute) O1A ON O1.id = O1A.id
    INNER JOIN O2 ON O2.id = R.object2
    INNER JOIN (SELECT O.*, OA.*, ATT.*, A.*
FROM Objects O LEFT JOIN ObjectAttributes OA ON O.id = OA.object
    LEFT JOIN AttributeTypes ATT ON ATT.id = OA.attributeType
    LEFT JOIN Attributes A ON A.id = OA.attribute) O2A ON O2.id = O2A.id
WHERE RT.name = 'article'
    AND (O1.id IN ('objectId1','objectId2') OR O2.id IN ('objectId1','objectId2'))
```


A gráf a lekérdezés alapján

The screenshot displays the NetBeans IDE interface. On the left, the 'Files' and 'Services' panes show the project structure for 'graph1' and 'gsproject'. The 'Source' pane shows a list of files with line numbers. The main window, titled 'GraphStream', displays a graph with three nodes: 'Object₁ neve...', 'Object₂ neve...', and 'Article₁ címe...'. Edges connect 'Object₁ neve...' to 'Article₁ címe...' and 'Article₁ címe...' to 'Object₂ neve...'. At the bottom, a 'show relation' window shows the following query:

```
1 show relation
2   between [object1, object2]
3 where
4   relationtype is article
```

A nyelvben rejlő lehetőségek

- ❖ A kidolgozott nyelv segítségével definiálhatunk objektumokat és rajtuk értelmezett kapcsolatokat is, amelyek az adatbázisba kerülnek, vagy nézeteként a memóriába:

```
define relation
```

```
  between[object1, object2] with attributes
```

```
  [attr1 = objectm,  
   attr2 = 'text1',  
   attrn = (strength, 30)]
```

A nyelvben rejlő lehetőségek

- ❖ Kérhetjük azokat a részgráfokat is, ahol a kapcsolatok erőssége (élkötegek száma) egy előre meghatározott értékkel rendelkezik.

```
show relation  
  between objects  
  with attributes [strength > 30]
```

Összefoglalás

- ❖ A keretrendszer és a nyelv aktuális állapotában nagyrészt lekérdezések futtatására alkalmas,
- ❖ a query-k futtatása lehetséges, de a szintaktikai és szemantikai lekérdezések ellenőrzések nem teljesek (Erlang nyelven implementáljuk ezeket)
- ❖ A prototípus még nem adja vissza az összes lehetőséget, ami a modellben rejlik (70%os a készültsége, de már használható).
- ❖ Jelenleg azon dolgozunk, hogy a nyelv segítségével ne csak lekérdezéseket futtathassunk, hanem új objektumokat is tudjunk definiálni.

Tervek

- ❖ Több objektum között többféle kapcsolatot lehet kialakítani, és az adott objektum maga is lehet kapcsoló elem.
- ❖ A felvitt, és szűrt adatokról pillanatképet lehet menteni.
- ❖ A pillanatképek között lehet válogatni (mint a verziókövető rendszerekben, pl. GIT, vagy Subversion)
- ❖ Nem csak adatok, vagy fájlok lesznek verziózva, hanem maga a szemantikus gráf pillanatnyi állapotai.
- ❖ A tervezhetőség minden lehetőségének a megteremtése.

Köszönöm a megtisztelő figyelmet!

Kérdések?