Introducing the MGRID HL7 Datatypes

Dijkstra | Enting | Havinga

Schiphol Airport, 27 October 2009

Today's agenda

Introduction
- Requirements on Medical IT
- The Datatype Gap

MGRID Solution

Datatype examples
- PQ
- PQ & TS flavours
- IVL<T>
- II
- CV
- CV / SNOMED CT

ORM Example

Questions

Requirements on Medical IT

- Support growing information needs
  - without increasing costs
  - without decreasing performance
- Support interactive queries
  - Support online transaction processing
  - Information is used to guide treatment, archival becomes secondary
- Support different and changing information requirements
  - Storage should not be optimized for one particular question, but for all
  - Use standards for annotation, storage and query

The Datatype Gap

- Database datatypes are far removed from HL7 datatypes
  - fill gap using ORM: HL7 datatypes allow for complicated operations which can easily translate into full tablescans
  - fill gap using custom code: standard is hard to interpret
- Current HL7 specific ORM implementations:
  - seem to consider only limited amount of data
  - have incomplete null handling

The MGRID solution

Native HL7 datatype support in the database

- Creates a query language that is powerful, fast and easy to learn
- Enables a lightweight hibernate interface for JavaSIG interoperability
- Makes terminfo / SNOMED CT implementations easy

Example: PQ

```
create table obs (ptnt int, date ts, dosage pq);
insert into obs values (1, '200910011214', '10 ml');
...
insert into obs values (1, '200910091214', '20 ml');

select ptnt, effectivetime::date,
  convert(sum(dosage),'ml') as "sum",
  convert(avg(dosage),'ml') as "average"
from obs
group by ptnt,effectivetime::date
having sum(dosage) > '100ml' order by ptnt,effectivetime;
 ptnt | effectivetime |  sum   | average
------+---------------+--------+---------
    1 | 2009-10-01    | 120 ml | 10 ml
    1 | 2009-10-02    | 150 ml | 50 ml
    1 | 2009-10-04    | 600 ml | 100 ml
    1 | 2009-10-09    | 180 ml | 20 ml
    2 | 2009-10-02    | 300 ml | 50 ml
    2 | 2009-10-09    | 120 ml | 20 ml
    3 | 2009-10-01    | 150 ml | 10 ml
    3 | 2009-10-02    | 450 ml | 50 ml
    3 | 2009-10-04    | 900 ml | 100 ml
    4 | 2009-10-04    | 300 ml | 100 ml
(10 rows)
```

PQ - Physical Quantity

- PQ supports all ucum units
- PQs with the same dimensions can be converted and compared
- Database can do PQ arithmetic and conversions
- Nullflavors handling considering QTY

Flavours of PQ and TS

```
create domain pq_time as pq
       constraint pq_time_compares_to_s
        check (value /= 's');
CREATE DOMAIN

create domain ts_date as ts
        constraint ts_date_prcs_le_8_no_tz
        check (isnull(toany(timezone(value))) = 'true'::bn
        and    "precision"(value) <= 8
        and    calendar(value)='GREG');
CREATE DOMAIN

create domain ts_date_full as ts
        constraint ts_date_prcs_eq_8_no_tz
        check (isnull(toany(timezone(value))) = 'true'::bn
        and    "precision"(value) = 8
        and    calendar(value)='GREG');
CREATE DOMAIN

select '10 ml'::pq_time;
ERROR: value for domain pq_time violates check constraint "pq_time_compares_to_s"
```

Example: IVL<PQ>

```
select '<8m'::ivl_pq;
         ivl_pq
-------------------------
 ]Nullflavor.NINF m;8 m[
(1 row)

select contains('<8m'::ivl_pq, '20cm'::pq);
 contains
----------
 true
(1 row)

select '2m-8m'::ivl_pq;
  ivl_pq
-----------
 [2 m;8 m]
(1 row)

select '[10ml; 20m3]'::ivl_pq;
      ivl_pq
------------------
 [1e-05 m3;20 m3]
(1 row)
```


Example: IVL<TS>

```
select '2008'::ts, promotion('2008'::ts), demotion(promotion('2008'::ts));
  ts   |  promotion   | demotion
-------+--------------+----------
 2008 | [2008;2009[ | 2008
(1 row)

select '200801 .. 02'::ivl_ts;
     ivl_ts
-----------------
 [200801;200803[
(1 row)

select '200801 .. 02'::ivl_ts <@ '2001 .. 2010'::ivl_ts;
 ?column?
----------
 t
(1 row)

select '<2005'::ivl_ts <@ '2001 .. 2010'::ivl_ts;
 ?column?
----------
 f
(1 row)

select intervalafter('2001 .. 2010'::ivl_ts, '20080102'::ts);
  intervalafter
-----------------
 ]20080102;2011[
(1 row)
```

IVL<T> - Interval

- All 7 IVL forms are supported
- Comparisons between interval forms are hard to program and easy to get wrong
- Additional functions:
  - overlaps, contains, intervalbefore, union

II - Instance Identifier

```
select '2.16.840.1.113883.5.1011:any extension'::ii;
                    ii
-----------------------------------------
 2.16.840.1.113883.5.1011:any extension
(1 row)

select uuid_generate_v1mc()::ii;
          uuid_generate_v1mc
--------------------------------------
 9fb6a05c-c237-11de-ab08-47231b73ad4b
(1 row)

select 'wrong ii'::ii;
ERROR:  could not parse root wrong ii
LINE 1: select 'wrong ii'::ii;
               ^
HINT:  The root must be either an OID or an UUID

create table testii (id ii);
insert into testii values ('2.16.840.1.113883.5.1011:any extension');
insert into testii values ('c478b896-c235-11de-8436-ebddef4f1dbe:another extension');
insert into testii select uuid_generate_v1mc() from generate_series(1,10000);

explain analyze select * from testii where id='c478b896-c235-11de-8436-ebddef4f1dbe';
                              QUERY PLAN
-----------------------------------------------------------------------
 Index Scan using iii on testii  (cost=0.00..8.27 rows=1 width=25)
   Index Cond: (id = 'c478b896-c235-11de-8436-ebddef4f1dbe'::ii)
 Total runtime: 0.179 ms
(3 rows)
```


CV - Coded Value

- CV is a CD with no translations and only a single concept
- CS is a CV with an implicit code system which is clear from the context. This codesystem is not visible in the string representation.

Example: CV

```
select oioid, oiname, oitype, oireleaseid, oistatus from pg_oid where oiname='ActClass';
            oioid              |  oiname   | oitype | oireleaseid | oistatus
-------------------------------+-----------+--------+-------------+----------
 2.16.840.1.113883.5.6         | ActClass |      5 |         145 | complete
 2.16.840.1.113883.1.11.11527 | ActClass |     11 |          -1 | -
(2 rows)

select 'OBS:2.16.840.1.113883.5.6'::cv;
            cv
---------------------------
 OBS:2.16.840.1.113883.5.6
(1 row)

select 'OBS:2.16.840.1.113883.5.7'::cv;
ERROR:  invalid code OBS for codeSystem ActPriority (7)
LINE 1: select 'OBS:2.16.840.1.113883.5.7'::cv;
               ^
```


Example: CV

```
select * from pg_conceptdomain where coname = 'ActClass';
 coid |  coname
------+----------
  362 | ActClass
(1 row)

select 'OBS'::cv('ActClass');
                                                cv
-------------------------------------------------------------------------------------
 OBS:2.16.840.1.113883.5.6@2009-08-30:2.16.840.1.113883.1.11.11527@2009-08-30
(1 row)
```

Example: CV

```
create table testcv (class cv('ActClass'), mood cv('ActMood'));

        Table "public.testcv"
 Column |      Type       | Modifiers
--------+-----------------+-----------
 class  | cv('ActClass') |
 mood   | cv('ActMood')  |

insert into testcv values ('wrong','values');
ERROR:  invalid code wrong for codeSystem ActClass (6)
insert into testcv values ('OBS','EVN.CRT');
insert into testcv values ('PROC','APT');
select class from testcv;
                                    class
-------------------------------------------------------------------------------
 OBS:2.16.840.1.113883.5.6@2009-08-30:2.16.840.1.113883.1.11.11527@2009-08-30
 PROC:2.16.840.1.113883.5.6@2009-08-30:2.16.840.1.113883.1.11.11527@2009-08-30
(2 rows)

select code(class), displayname(class), code(mood), displayname(mood) from testcv;
 code | displayname |  code   |   displayname
------+-------------+---------+-----------------
 OBS  | observation | EVN.CRT | event criterion
 PROC | procedure   | APT     | appointment
(2 rows)
```


Example: SNOMED CT

```
select conceptid::text::cv('SNOMED-CT') into cvgi from
   (select conceptid from snomed.concepts order by random() limit 1000) a;
describe cvgi;
           Table "public.cvgi"
  Column   |      Type       | Modifiers
-----------+-----------------+-----------
 conceptid | cv('SNOMED-CT') |

select code(conceptid),displayname(conceptid),codesystemname(conceptid)
  from cvgi where conceptid << '404684003'::cv('SNOMED-CT') limit 10;
   code    |                displayname                 | codesystemname
-----------+--------------------------------------------+----------------
 5919001   | Rupture of papillary muscle                | SNOMED-CT
 193666005 | Unspecified visual field defect            | SNOMED-CT
 22395006  | Oesophageal body web                       | SNOMED-CT
 254943007 | Benign tumour of choroid plexus            | SNOMED-CT
 293041006 | Sulphaguanidine adverse reaction           | SNOMED-CT
 199617004 | Large-for-dates unspecified                | SNOMED-CT
 168624001 | Plain X-ray of radius/ulna normal          | SNOMED-CT
 24825006  | Central alveolar hypoventilation syndrome  | SNOMED-CT
 57361003  | Anomaly of chromosome pair 5               | SNOMED-CT
 269468002 | Malignant neoplasm of short bones of leg   | SNOMED-CT
(10 rows)

select '71620000|fracture of femur|:
   116676008|associated morphology|=21947006|compression fracture|
   ,363698007|finding site|=29627003|structure of neck of femur|
'::cv('SNOMED-CT');
                                        cv
-------------------------------------------------------------------------------
 71620000|fracture of femur|:
    116676008|associated morphology|=21947006|compression fracture|
    ,363698007|finding site|=29627003|structure of neck of femur|:2.16.840.1.113883.6.96
(1 row)
```

Example: SNOMED CT

In an Act instance where Act.code attribute is a SNOMED CT expression, the expression must represent a type of [363787002|observable entity], [«129125009|procedure with explicit context] or [«272379006|event]

```
create table act (class cv('ActClass'), code cv);

alter table act add constraint act_code_snomedct CHECK (
     (codesystem(code)='2.16.840.1.113883.6.96') ?
       (code << '363787002:2.16.840.1.113883.6.96'::cv or
        code << '129125009:2.16.840.1.113883.6.96'::cv or
        code << '272379006:2.16.840.1.113883.6.96'::cv));

insert into act values ('PROC','224166006'::cv('SNOMED-CT'));
ERROR:  new row for relation "act" violates check constraint "act_code_snomedct"

insert into act values ('ACT','3974003'::cv('SNOMED-CT'));

select code(class),displayname(code) from act;
 class |         displayname
-------+-----------------------------
 ACT   | Contact with sharp leaves
(1 row)
```

ORM MGRID using Hibernate

- Consider for instance storing a table of PQs;
- Differences using 'standard datatypes' versus HL7 datatypes:

versus	**standard**	**HL7 datatypes**
java api	JavaSIG	JavaSIG + MGRID mappings
table definition	2 columns; value unit	1 column; pq
query	full scan, make canonical, filter	partial scan

Hibernate mapping is trivial

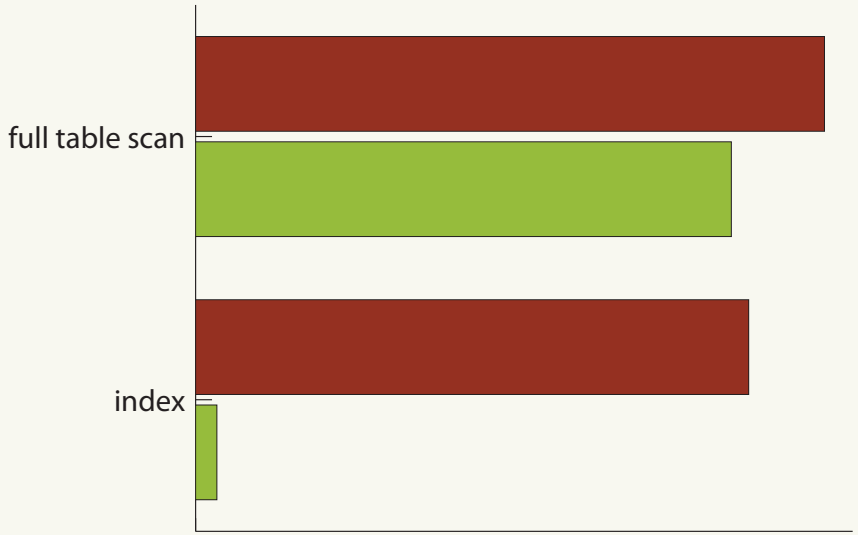
- provide own `nullSafeSet` and `nullSafeGet`

```
public void nullSafeSet(PreparedStatement st, Object rawValue, int index)
    throws HibernateException {
    try {
        if (rawValue == null) {
            st.setNull(index, java.sql.Types.OTHER);
        } else {
            PGobject object = new PGobject();
            PQ pq = (PQ) rawValue;
            object.setType("pq");
            object.setValue("" + pq.value() + " " + pq.unit());
            st.setObject(index, object);
        }
    } catch (SQLException ex) {
        throw new HibernateException(ex);
    }
}
```


Hibernate performance comparison

- Slightly better performance when using full table scans

- Enormous performance boost when using indexes



Questions

Contact

MGRID

ir. Willem Dijkstra
Managing Director

www.mgrid.net

T +31 886 474 302
F +31 886 474 301
M +31 611 144 118
w.p.dijkstra@mgrid.net

Oostenburgervoorstraat 106-114
1018 MR Amsterdam

PO BOX 1287
1000 BG Amsterdam
The Netherlands