

Note that in our description, we use $A.*$ to represent the annotation information. More specifically,

- $MeasType = \{\langle \underline{MeasTypeId}, ObsTypeId, CharType, StdType, ProtType, Precision, Value, isKey \rangle\};$
- $ObservationType = \{\langle \underline{ObsTypeId}, EntTypeId, isUnique \rangle\}$
HP Question: I did not use “AnnotId” in the algorithm, so I remove it here. How shall this be used?
- $ContextType = \{\langle \underline{ObsTypeId}, ContextObsTypeId, RelType, isIdentify \rangle\}$
HP Question: Add the *isIdentify* for the purpose of checking whether we need to use one observation’s context for key, is this ok?
- $Map = \{\langle \underline{MeasTypeId}, ObsTypeId, Cond, Val \rangle\}$

$OBOE.*$, on the other hand, represent the *OUTPUT* data represented in the OBOE model. In detail,

- $Observation = \{\langle \underline{ObsId}, EntId \rangle\}$ keeps all the observation instances materialized from *Dataset*.
HP Question: I did not use “AnnotId” in the algorithm, so I remove it here. How shall this be used?
- $Measurenebt = \{\langle \underline{MeasId}, ObsId, Characteristic, Val \rangle\}$
Changed to:
 $Measurenebt = \{\langle \underline{MeasId}, MeasTypeId, ObsId, Val \rangle\}$
HP note: the other information about the measurement type, e.g., standard, characteristic, can be gotten using *MeasTypeId*. In real application, we can think of “duplicating the measurement type” information. Too detail. Ignore here.
- $Entity = \{\langle \underline{EntId}, EntType \rangle\}$
- $Context = \{\langle \underline{ObsId}, ContextObsId, RelType \rangle\}$
HP Question: I did not use “Relationship” table, do we really need to instantiate this information?

Each row in the input dataset represents the information related to one observation and its contexts.

The algorithm tries to catch the *key*, *unique* and *identifying* constraints in the annotation during the materialization process.

We can run the example in page 6 in the powerpoint file to explain the algorithm. For Row(2007, 1, piru, 35.8)

- Create a measurement $mi_1 = \langle meas1, null, Year, 2007 \rangle$
- Create a measurement $mi_2 = \langle meas2, null, Year, 35.8 \rangle$
- Create a measurement $mi_3 = \langle meas3, null, TaxonomicTypeName, Picea rubens \rangle$
- Create a measurement $mi_4 = \langle meas4, null, EntityName, 1 \rangle$
- Get $MeasSet = \{mi_1, mi_2, mi_3, mi_4\};$
- Step 2: Get $KeyIdx = \{o_1 \rightarrow \{mi_1\}, o_2 \rightarrow \{mi_2, mi_3, mi_4\}\}$
- Step 3-4: for each observation types o_1 and o_2
- for o_1
 - Since m_1 is specified as key, get the $KeyVal = 2007;$

- No entity with this key exists in $EntIdx$, create an entity $ei = \langle ent1, TemporalRange \rangle$; Now, $EntIdx = \{\langle o1, 2007 \rightarrow ent1 \rangle\}$.
- Since o_1 is specified as *distinct*, need to make sure we do not create redundant observations. No entry with this key exists in $ObsIdx$, so, create an observation $oi = \langle obs1, ent1 \rangle$. Now, $ObsIdx = \{\langle o1, 2007 \rangle \rightarrow obs1\}$
- When deal with o_2 ,
 - $KeyVal = 1$.
 - Create an entity $ei = \langle ent2, Tree \rangle$; $EntIdx = \{\langle o1, 2007 \rightarrow ent1 \rangle, \langle o2, 1 \rightarrow ent2 \rangle\}$.
 - Create an observation $\langle obs2, ent2 \rangle$. No need to update $ObsIdx$ because $o2$ is not identified as *unique*.
- Connect mi_1 to $obs1$;
- Connect mi_2 , mi_3 and mi_4 to $obs2$;
- Set the context relationship between $obs1$ and $obs2$;

For Row (2008, 1, piru, 36.2)

- Create a measurement $mi_5 = \langle meas5, null, Year, 2007 \rangle$
- Create a measurement $mi_6 = \langle meas6, null, Year, 35.8 \rangle$
- Create a measurement $mi_7 = \langle meas7, null, TaxonomicTypeName, Picea rubens \rangle$
- Create a measurement $mi_8 = \langle meas8, null, EntityName, 1 \rangle$
- Get $MeasSet = \{mi_5, mi_6, mi_7, mi_8\}$;
- Step 2: Get $KeyIdx = \{o_1 \rightarrow \{mi_5\}, o_2 \rightarrow \{mi_6, mi_7, mi_8\}\}$
- for type o_1
 - $KeyVal = 2008$;
 - Create an entity $\langle ent3, TemporalRange \rangle$; $EntIdx = \{\langle o1, 2007 \rangle \rightarrow ent1, \langle o2, 1 \rangle \rightarrow ent2, \langle o1, 2008 \rangle \rightarrow ent3\}$.
 - Create an observation $\langle obs3, ent3 \rangle$, and $ObsIdx = \{\langle o1, 2007 \rightarrow obs1 \rangle, \langle o2, 1 \rightarrow obs2 \rangle, \langle o1, 2008 \rightarrow obs3 \rangle\}$
- When deal with o_2 ,
 - $KeyVal = 1$.
 - **Some item $\langle o2, 1 \rangle \rightarrow ent2$ is already in $EntIdx$, so get the entity id $ent2$. No need to create an entity.**
 - Since $o2$ is not specified with *unique yes*, we NEED to create an observation $\langle obs4, ent2 \rangle$. No need to update $ObsIdx$.

For ROW (2008, 2, abba, 33.2)

- For $o1$'s measurement 2008, since $\langle o1, 2008 \rangle \rightarrow ent3$ already exists in $EntIdx$, **No need to create a new ENTITY.**
- Since $o1$ is specified with *unique yes*, and since $\langle o1, 2008 \rangle \rightarrow obs3$ already exists in $ObsIdx$, **No need to create a new OBSERVATION** and no need to put the measurement for 2008 into OBOE model.

Algorithm 1 MaterializeDB (*Dataset*, *A*)

```
/* Dataset: [Input] in the form of flat file */
/* A : [Input] Annotations*/

ObsIdx =  $\emptyset$ ; /* for keeping index  $\langle \text{ObsTypeId}, \text{KeyVal} \rangle \rightarrow \text{ObsId}$  */
EntIdx =  $\emptyset$ ; /* for keeping index  $\langle \text{MeasTypeId}, \text{KeyVal} \rangle \rightarrow \text{EntId}$  */
for (each Row  $\langle A1, A2, \dots, An \rangle \in \text{Dataset}$ ) do
    MeasSet =  $\emptyset$ ; /* Keep the set of new measurement instances */

    /* 1. Create new orphan measurement instances */
    for (each m =  $\langle \text{MeasTypeId}, \text{ResAttribute}, \text{Cond}, \text{Val} \rangle \in A.\text{Map}$ ) do
        if (m.ResAttribute! = Row.Ai.Attrname) OR (Row.Ai does not satisfy m.Cond)
            then
                continue;
            end if
        ObsTypeId = GetObsTypeId (A.MeasType, m.MeasTypeId);

        MeasId = GetNewMeasId(OBOE.Measurement);
        if(m.Val! = NULL) MeasVal = m.Val;
        else MeasVal = Row.Ai.Val;
        Create a measurement instance mi =  $\langle \text{MeasId}, \text{MeasTypeId}, \text{null}, \text{MeasVal} \rangle$ ;
        Add mi to MeasSet;
    end for

    /* 2. Get observation types and measurement types with new instances */
    KeyIdx =  $\emptyset$  /* Keep index for  $\text{ObsTypeId} \rightarrow \{mi\}$  */
    for (each mi  $\in \text{MeasSet}$ ) do
        ObsTypeId = GetObsTypeId (A.MeasType, mi.MeasTypeId);
        Update KeyIdx by changing the item  $\text{ObsTypeId} \rightarrow \{mi\}$ ;
    end for

    for (each ObsTypeId  $\in \text{KeyIdx.keys}$ ) do
        /*Get the key value for this observation.
        Case 1: generally, it is the value for the “key” measurement.
        Case 2: several measurement types are marked with “key yes”, the key value is the
        combined value of these several measurement.
        Case 3: this object type is marked with “identifying yes, the key value ” is the
        combined value with its context observation’s key measurement values. */
        KeyVal = GetObsTypeKeys (ObsTypeId, KeyIdx);

        /* 3. Get an existing or create a new entity instance */
        HasKey = false;
        if (ObsTypeId has key measurements) then
            HasKey = true;
        end if
        EntId = MaterializeEntity(ObsTypeId, HasKey, KeyVal, EntIdx, A, OBOE)

        /* 4. Get an existing or create a new observation instance */
        ContextIdx =  $\emptyset$ ; /* keep index  $\text{ObsTypeId} \rightarrow \text{ObsId}$  to materialize context */
        for (each ObsTypeId  $\in \text{KeyIdx.keys}$ ) do
            IsObsUnique = checkIfObsUnique(A.ObservationType, ObsTypeId);
            ObsHasKey = HasKey && IsObsUnique ;
            ObsId = MaterializeObs(ObsTypeId, ObsHasKey, KeyVal, EntId, ObsIdx, OBOE);

            /*Maintain the measurement instances for this observation */
            miSet = GetMeasInst(KeyIdx, ObsTypeId);
            if (ObsId is a new one) then
                Set the obsId to each mi  $\in \text{miSet}$  so that mi-s are not orphans;
            else
                Discard all the mi  $\in \text{miSet}$ ;
            end if
            Put all the mi  $\in \text{miSet}$  to OBOE.Measurement;
        end for
    end for

```

Algorithm 2 MaterializeEntity($ObsTypeId, HasKey, KeyVal, EntIdx, A, OBOE$)

```
EntType = GetObsEntityType ( $A.ObservationType, ObsTypeId$ )
CrtNewEntInst = true;
if ( $HasKey == \text{true}$ ) then
  EntId = GetEntId( $ObsTypeId, KeyVal, EntIdx$ );
  if ( $EntId$  is not Null) then
    CrtNewEntInst = false;
  end if
end if
if ( $CrtNewEntInst == \text{true}$ ) then
  EntId = CrtEntId( $EntType$ );
  Create an entity instance  $ei = \langle EntId, EntType \rangle$ ;
  Put  $ei$  to  $OBOE.Entity$ ;
  if  $HasKey == \text{true}$  then
    /*Only when this is the key measurement, we need to maintain the index*/
     $EntIdx = EntIdx \cup \{ \langle ObsTypeId, KeyVal \rangle \rightarrow EntId \}$ ;
  end if
end if
return  $EntId$ ;
```

Algorithm 3 MaterializeObs($ObsTypeId, HasKey, KeyVal, EntId, ObsIdx, OBOE$)

```
CrtNewObsInst = true;
if ( $HasKey == \text{true}$ ) then
  ObsId = GetObsId( $ObsTypeId, KeyVal, ObsIdx$ );
  if ( $ObsId$  is not Null) then
    CrtNewObsInst = false;
  end if
end if
if ( $CrtNewObsInst == \text{true}$ ) then
  Create an observation instance  $oi = \langle ObsId, EntId \rangle$ 
  Put  $oi$  to  $OBOE.Observation$ ;
  if ( $HasKey == \text{true}$ ) then
    /*Only when it has key measurement and it is identified as unique, we need
    to maintain the index*/
     $ObsIdx = ObsIdx \cup \{ \langle ObsTypeId, KeyVal \rangle \rightarrow ObsId \}$ ;
  end if
end if
Return  $ObsId$ ;
```

Algorithm 4 MaterializeContext($ContextIdx, A, OBOE$)

```
for ( $ObsTypeId \rightarrow ObsId \in ContextIdx$ ) do
   $ContextObsTypeId, Rel = \text{GetContextObsTypeRel}(A.ContextType, ObsTypeId)$ ;
  if ( $ContextObsTypeId$  is not Null) then
     $ContextObsId = \text{GetContextObsId}(ContextIdx, ContextObsTypeId)$ ;
    Create a context  $c = \langle ObsId, ContextObsId, Rel \rangle$ ;
    Put  $c$  to  $OBOE.Context$ ;
  end if
end for
```
