Best Practices for Implementations

Person Attribute vs. Observation

Patient data are normally stored as observations in OpenMRS, but could be stored as person attributes. Person attributes are used to describe something about the person (e.g., place of birth) whereas observations are normally something recorded at a moment in time (e.g., current weight).

If person attributes support having a historical record, they become more sophisticated than the original "virtual column in the person table." For example, you might want to record and keep a history of their civil status since this can change over time. Is it better to record civil status as a person attribute with concept answers OR as an observation with dates? You don't care what day the patient got married, but that when they had encounter data captured, the civil status was Married. By using an observation, these are recorded as snapshots over time, and that's probably the real use case.

These are examples which have been recorded as patient attributes:

- Birthplace -> never changes
- Citizenship -> rarely changes, but should allow multiple values on rare occasions
- Civil status -> changes
- Assigned health center -> program enrollment location
- District of assigned health center -> calculated off the program enrollment location
- Mother's name -> never changes
- Telephone number -> changes; handled by PatientContact entry (OpenMRS 1.9+)
- Telephone type -> changes; handled by PatientContact entry (OpenMRS 1.9+)
- Treatment supporter -> changes; handled by Relationship
- Current school -> changes

Some of these attributes are fixed and will never change - i.e. birthplace and mother's name. The rest of them could potentially change over time, in which case we would want to be able to represent that history. "Voiding" is not a proper solution for this, as we do not want to invalidate the historical data. We want to mark it as something that was true in the past, but is not true in the present.

If using person attributes is the proper way to model this data, we need to add date bounds to PersonAttribute, and to allow a person to have multiple PersonAttributes of the same type (either at the same time or at different times).

If it is more appropriate to use other mechanisms for these (Observations; Patient Programs, which supports Locations; or Relationships, which supports start/end dates), we need best practices documentation and to get some feedback into the community as to how these use cases would better be modeled. Do others have examples of Person Attributes that would need start date and end date bounds to meet their full needs?

Most of these are more properly handled in some other way. Some do change, but have a legitimate claim to be person attributes (ie. civil status and current school). In those cases it depends on the rationale for collecting these things. If you're collecting them for epidemiological/research purposes, they should probably be observations recorded at an intake encounter. If you really want to store the marital history of a patient, or the school someone went to, for an administrative purpose, you might actually need start/end validity dates on person attributes. If it is to be used for administrative purposes, it may be more appropriate as a program enrollment.

Recommended Best Practices

- If a data point is fixed and will never change for all intents and purposes (eg. birthplace) use a PersonAttribute (or VisitAttribute, etc)
- If a data point is not fixed, and may change over time but you might not know the dates of starting and stopping, use an Observation
- If a data point is not fixed, and you want to capture the exact start and end dates:
  - Use a Relationship, if possible
  - Use a PersonContact, if possible
  - Use a PatientProgram, if possible
  - Use an ObsGroup otherwise

Comments

Comments from Roger Friedman

We have similar issues in the HR Module: We want to keep a job history as well as knowing the person's current job. We maintain begin and end dates, and at most one job per employee has a null end date. However, there remains the issue, as yet not resolved, of distinguishing between "edits" (to correct errors) and "changes" (to indicate a job change). The choice seems to be between not having edits, which means voiding and replacing erroneous records; and requiring the user to distinguish consistently between edits and changes.

While putting this capability into the Person Attribute table is no different than having a "preferred" name or address, probably a small percentage of patients have more than one name or address, while potentially many person attributes could have multiple records. Perhaps the best solution performance-wise would be to capture new/changed Person Attribute records as they are written and put them in a separate table. This would be easy and invisible if you can live with creation/modification dates. You could use the separate table to do longitudinal reports by, say, civil status. Also you could implement it by AOP and not have to monkey with the trunk. That's not the case for HR, which is interested in effective dates of personnel actions.