There aren’t a lot of places out there where this has really been well thought through and there isn’t clear understanding with supporting use cases.  The “concern” tracker act permits us to represent the clinical thinking process.  When I doctor treats you, she or he may not know what is wrong.  They group together your problems, symptoms, complaints, other objective findings from the exam and develop a diagnosis or a differential diagnosis (multiple possible diagnoses which then need to be ruled out or confirmed to see which diagnosis is the real problem.)  Over the episode of care, which may involve multiple visits, the concern remains there to track the issue.  Your symptoms may change, complaints may change, findings may change and the diagnosis may change over time, as the doctors sees how your health progresses and how certain interventions play out.  It is a shame that more implementers don’t use this concern structure to its full potential.  When you think of the need to deal with a person’s diabetes in addition to their flu, you can see that having different concerns allows the focus to be aligned with the different issues that may need to be handled separately.  The fever, vomiting and runny nose may all be associated with the flu concern while the dry mouth may be associated with the diabetes.  The feeling of exhaustion may be determined to be associated with the flu rather than the diabetes if the person’s blood sugar has been very well controlled. If there blood sugar had not been well controlled, that symptom may go with the diabetes concern and trigger a medication change. This really gives a good picture of care over time.  I wish more people understood and used its power.

I’m worrying about situations where someone may try to count the number of concerns being tracked.  As we move toward Quality Measures and outcome assessment, people are going to need to use this data to count things. I think we need to be careful not to leave this use case out of our thinking about how best to represent zero problems on the problem list, zero allergies on the allergies list, or zero medications on a person’s med list.  If we need to say there are none, we need to be careful that it won’t look like there are some.  There will be other cases where a problem observation could carry a negationInd of true.  We need to be sure that computers will have a clear way to find places where the patient has zero.  I think we need to accurately represent the situation of no allergies or no problems, as not being concerns.  This will accurately not count these situations as concerns.

We also need to be careful about showing and explaining that the authorship may be asserted at the level of the concern, or at the level of the problem observation. In the examples below, I am showing the author at the concern level.  We really need to work through the use cases where a patient sees a physician’s assistant on the first visit and that person records a concern, then the patient comes back for a follow-up because there symptoms were not clearing up, or had changed a bit, and now sees the physician who records new problem observations and perhaps changes the diagnosis associated with this concern.  We need to work on this further, but I just wanted to be sure that we don’t assume the authorship is only associated with the problem observation, and not the concern.

One of the reasons I have been pushing for us to consider all these needs together, and not do them just one at a time, is that I believe we have inconsistency issues in the current designs of these templates.  These design inconsistencies are adding to the challenge of how to consistently represent zero or “none”.  Part of what I think we need to do is align our template designs so that we could come up with a more consistent solution to meet the need to express zero of some type of entry.

Below I have included a bit of an analysis I have done to compare all the C-CDA R1.1 template designs regarding the vocabulary binding for the code element and for observation templates the code and value elements. The analysis is very telling and shows the inconsistencies in our designs across this family of templates.

I have also attached a set of proposals for how to address the needs to record zero for certain types of entries.  The approach is very consistent, but it requires me to simultaneously recommend a couple of design tweaks to first achieve greater design consistency in the base templates.

For problems, we could add the concept of “none” to the Problem Value set.  Then we could negate the concern act and put the current time in the effectiveTime/low. This would say there is no concern at this time. Then include one problem observation where the code is “Problem” and the value is “none”.

For allergies, it is unfortunate that a consistent strategy was not used for the design of the template. I can’t see why they were not designed to work the same way. I’m guessing the two designs were not considered together when the designs were being created.  It would be ideal for the Allergy/Adverse Event Type to be in the code element and then have a vocabulary binding on the value element to a value set of things the person is allergic to.  (The concept of Adverse Event really does not belong in here, IMO. While it is true that an allergic reaction is an adverse event, not all adverse events are allergic reactions.  A fall, for example could be an adverse event, a wrong dosage administered could be an adverse event, an infection could be an adverse event….this value set currently mixes things up in a way that is not right.) Anyway, if we had more consistency in our template approach, then it would be easier to devise a consistent approach for representing none.  If we adjusted the Allergy-Intolerance Observation to use the same “design pattern” as Problem Observation, then we could include the concept of “none” in the Allergy-Intolerance Substance value set.  Make this change would also help us address another use case that has come up in PCC where we need to record that an allergy or intolerance is to a certain type of mold, but we need a place to say that means we are “allergic to” vinegar, ketchup, barbeque sauce, etc.  Without touching the current modeling for the allergen in the participation, we could use the value element to build the list of things we shouldn’t be eating, or interacting with.  This is a very practical need that we don’t currently have an answer for, but this change would help us meet this other need too.

For procedures, the vocabulary binding is too broad. It isn’t really correct to mention a whole code system in a vocabulary binding.  If a proper value set was created and included in the vocabulary binding, then we would have the right set of terms that might be mentioned for the types of procedures that may or may not have been performed.

(I am copying Lloyd McKenzie on this thread so that we also can make sure that what we come up with is consistent with FHIR, if possible, so that we minimize smooth interoperability across CDA and FHIR.)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Template Name | @code | | | Code | | @code | | Value | | Code Element Conformance | | Code | | Value Element Conformance |  | |
| Allergy - Intolerance Observation | Y | | | A | | Y | | CD | | SHALL contain exactly one [1..1] code (<CONF:15947>).  This code SHALL contain exactly one [1..1] @code="ASSERTION" Assertion (CodeSystem: ActCode 2.16.840.1.113883.5.4 STATIC) (<CONF:15948>). | | Assertion | | This value SHALL contain exactly one [1..1] @code, which SHALL be selected from ValueSet Allergy/Adverse Event Type Value Set 2.16.840.1.113883.3.88.12.3221.6.2 DYNAMIC (<CONF:9139>). |  | |
| **Problem Observation** | N | | | VS | | Y | | VS | | 6. SHALL contain exactly one [1..1] code, which SHOULD be selected from ValueSet Problem Type 2.16.840.1.113883.3.88.12.3221.7.2 STATIC 2012-06-01 (<CONF:9045>). | | Problem Type | | 10. SHALL contain exactly one [1..1] value with @xsi:type="CD", where the @code SHOULD be selected from ValueSet Problem Value Set 2.16.840.1.113883.3.88.12.3221.7.4 DYNAMIC (<CONF:9058>). a. This value MAY contain zero or more [0..\*] translation (<CONF:16749>). i. The translation, if present, MAY contain zero or one [0..1] @code (CodeSystem: ICD10CM 2.16.840.1.113883.6.90 STATIC) (<CONF:16750>). b. This value MAY contain zero or one [0..1] @nullFlavor (<CONF:10141>). i. If the diagnosis is unknown or the SNOMED code is unknown, @nullFlavor SHOULD be “UNK”.  If the code is something other than SNOMED, @nullFlavor SHOULD be “OTH” and the other code SHOULD be placed in the translation element (<CONF:10142>). |  | |
| **Medication Activity** | |  |  | |  | |  | | Conformance for code element is missing. No guidance is provided. | | No guidance | |  | |  |  |
| **Procedure Activity Procedure** | | N | CS | |  | |  | | 5. SHALL contain exactly one [1..1] code (<CONF:7656>). a. This code SHOULD contain zero or one [0..1] originalText (<CONF:19203>). i. The originalText, if present, SHOULD contain zero or one [0..1] reference (<CONF:19204>). 1. The reference, if present, SHOULD contain zero or one [0..1] @value (<CONF:19205>). a. This reference/@value SHALL begin with a '#' and SHALL point to its corresponding narrative (using the approach defined in CDA Release 2, section 4.3.5.1) (<CONF:19206>). b. This code in a procedure activity SHOULD be selected from LOINC (codeSystem 2.16.840.1.113883.6.1) or SNOMED CT (CodeSystem: 2.16.840.1.113883.6.96), and MAY be selected from CPT-4 (CodeSystem: 2.16.840.1.113883.6.12) or ICD10 PCS (CodeSystem: 2.16.840.1.113883.6.4) (<CONF:19207>). | | Code System | |  | |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |