Code

logo

HL7 Domain Analysis Model Coordination

January 2019

HL7 Informative Document

Sponsored by

Clinical Interoperability Council

Co-sponsored by

Public Health

Patient Care

Copyright © 2019 Health Level Seven International ® ALL RIGHTS RESERVED. The reproduction of this material in any form is strictly forbidden without the written permission of the publisher. HL7 and Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. Pat & TM Off.

IP page

Contents

[1. Introduction: Problem statement 5](#_Toc530994138)

[2. Methods 5](#_Toc530994139)

[A. Gap analysis 6](#_Toc530994140)

[B. Analytical decomposition 6](#_Toc530994141)

[C. FHIM support 7](#_Toc530994142)

[3. Results 7](#_Toc530994143)

[A. Model Content 7](#_Toc530994144)

[A. Domain: Provider 8](#_Toc530994145)

[B. Domain: Patient 9](#_Toc530994146)

[C. Domain: Adverse Event 9](#_Toc530994147)

[D. Domain: Substance 10](#_Toc530994148)

[E. Naming conventions 10](#_Toc530994149)

[F. Documentation 11](#_Toc530994150)

[G. States 11](#_Toc530994151)

[H. Data Types 12](#_Toc530994152)

[I. Relationships 12](#_Toc530994153)

[J. Terminology 13](#_Toc530994154)

[K. Harmonization options 14](#_Toc530994155)

[L. Requirements traceability 17](#_Toc530994156)

[M. Requirements currency 17](#_Toc530994157)

[4. Discussion 17](#_Toc530994158)

[5. Conclusions 17](#_Toc530994159)

[6. Appendices 18](#_Toc530994160)

Participants

Galen Mulrooney, Modeling facilitator

Stephen Hufnagel, Modeling facilitator

Jay Lyle, Project coordinator

Steve Wagner, FHIM program manager

# Introduction: Problem statement

Among the many reasons data specifications diverge, causing redundant interface development and degrading the quality of interoperable data, is that they are based on different requirements assets. Non-aligned analysis results in non-aligned specifications.

We propose that maintaining these requirements assets in a framework that supports harmonized analysis modeling will allow requirements specifiers to work more efficiently and will allow specification developers to support cross-platform consistency more easily.

The objectives we see for coordinated DAMs are the benefits of elements that are aligned across domains and specifications:

* Reduce effort to author requirements, as much of the context of a domain can be leveraged from other efforts
* Reduce effort to keep elements consistent across specifications (such as FHIR & C-CDA), as the subtending model framework will maintain asserted semantics over time and across organizations
* Support consistent understanding of data (query, CDS, integration)
* And, if coordination extends to bi-directional traceability, reduce effort to keep requirements up to date, as specifications will be living documents that don’t require wholesale reinvention

The objectives of this study are

* To demonstrate the feasibility of hosting DAMs in a coordinated model
* To Identify issues with this approach and recommend solutions
* To characterize the value that this approach can provide.

# Methods

We proposed to take two DAMs that have domain overlap and put them into a single model. We expected to identify issues as we did so, and to make tactical decisions about how to resolve these issues. One goal of this study is to solicit feedback on these decisions.

The DAMs we chose are Allergy & Intolerance and Immunization. They overlap—i.e., provide semantically redundant content—in four areas:

* Patient
* Practitioner
* Substance
* Reaction

Our expectation is that the results of this work will support future specification development to support, e.g., inference of an Allergy/Intolerance and reaction record from an immunization adverse event.

We found limited value in analysis of use cases. There is clearly overlap between the Allergy *Use Case 3: Adverse Reaction to Medications* and the Immunization *P04: Manage Adverse Event Reporting*, but the different levels of granularity mean that most of the Allergy case occurs in the interstices of the Immunization case. While we expect to investigate how alignment with the EHR system functional model might assist our efforts in the future, at this point, we focus on the information model only.

The first issue we encountered is what it meant to have two DAMs in one model. Where the DAMs differ, would the best approach be to select the best one, filling in gaps as identified by the other, using a third model to connect the respective elements, or simply identify the difference and propose that the sponsoring workgroups determine what to do? If a coordinating model were to be used, should it leverage other existing assets, or should it limit its involvement to the precise semantics of the input models?

Because the answers to these questions are not obvious, we drafted three distinct approaches to the problem. Each approach has benefits and challenges, and we attempt to identify these for evaluation. The first set of feedback we hope to gather is on the relative value of these options.

These efforts – gap analysis, analytical decomposition, and FHIM support—are outlined below.

## Gap analysis

The most straightforward approach is to identify the synonymous parts of the respective models and document their similarity. The results can be used to generate a set of deltas to inform harmonization across work groups.

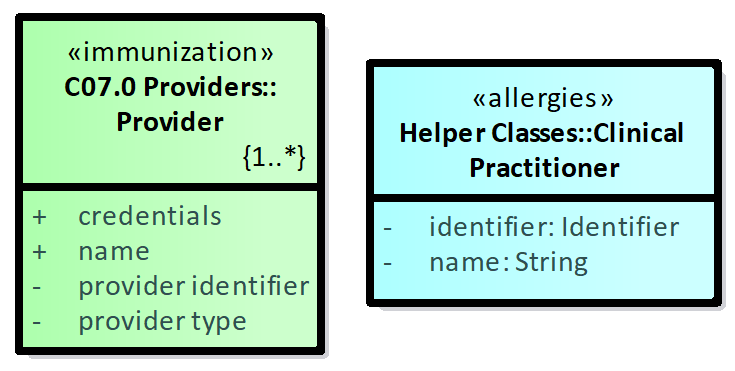


Figure 1: Comparison of Provider and Practitioner

Figure 1, for instance, could be used to request harmonization of the names “Provider” and “Clinical Practitioner,” or of the property names or datatypes.

## Analytical decomposition

A second method is to create a new model that represents both DAMs, but which asserts synonymy by making the DAM elements specializations of more general elements. This approach demonstrates synonymy, both in the underlying common elements and in their manifestations in more specific DAMs. In the figure below, both the substance in the Allergy DAM and the vaccine from the Immunization DAM are modeled as substances, one directly and one via specialization.



Figure 2: Analytical decomposition of Substance, with compositional domain classes

## FHIM support

The third method leverages the FHIM, which already documents DAM requirements, as well as specification requirements. This approach foregrounds the FHIM elements that support the DAM specifications, demonstrating full support. The elements so identified are also traced to a variety of specification requirements, so they can be used to support transforms across specifications as well as to identify downstream effects of modifications to requirements.

Because the FHIM contains information not specified in the DAMs, we adopt a masking approach to reproduce an artifact at the level of granularity preferred by the subject matter experts. In some cases, the FHIM model supports this masking easily. In others, the relationships make the masking difficult, and possibly misleading, so we have left some of the complexity of the FHIM in the DAM presentation. We solicit feedback on this choice.

<<img>>

Figure 3: FHIM view, filtered to show DAM-specific content

# Results

We documented the kinds of issues we faced in this process, and each issue discussion includes perspectives from the respective approaches.

## Model Content

Both DAMs contain dynamic and static content. They demonstrate different processes.

The immunization model begins with a set of real-world scenarios called “Storyboards”—descriptions of concrete cases used to discover and confirm the required scope of functionality. It then names the Actors in these storyboards, defines the Information Model that articulates the knowledge these actors need, details a set of compositional UML Use Cases to show how the needs of various actors are met, and illustrates the flow of the cases in a set of Activity Diagrams.

The allergy model begins with real-world scenarios, but it calls them “Use Cases.” Each case lists actors, but they are not used to define a set of standard actor roles. It then builds Activity Diagrams which seem to represent the functionality implied in the use cases, but which don’t provide explicit tracing to allow a reader to verify this assumption. It provides a state machine for clinical knowledge of sensitivities. And then it provides an information model.

The FHIM models information from specifications, so the class model is the primary artifact. Use cases in the FHIM are packages used to generate implementation artifacts for specific technologies in specific domains, using underlying UML profiles. The FHIM also provides terminology bindings, following the specifications it supports.

A great deal of the analysis in this project was identifying where the two DAMs contain the same information and where they diverge. We make assumptions regarding classes and properties we deem to be synonymous. We do not intend to imply that facets modeled by one DAM and not the other are therefore relevant to the other, only that cases that cross domains may assume that, e.g., a provider in an allergy case may have an organization that can be used when generating an Immunization artifact.

|  |  |  |
| --- | --- | --- |
| **Allergy DAM Classes** | **Immunization DAM Classes** | **FHIM Classes** |
| Clinical Practitioner | provider | HealthcareProvider:: IndividualProvider |
| Patient | person::patient | Person::Patient |
| Substance | vaccine | Substance, Vaccine |
| AdverseReaction | adverse reaction | ReactionObservation |

Table 1: Domains of overlap

## Domain: Provider

The Provider classes seem to align well: both “Clinical Practitioner” and “provider” denote persons licensed to provide healthcare.

Names and identifiers are not specified in detail. Both assume a single name and a single identifier: immunization explicitly and allergy by default UML convention. While EHRs may record multiple values for either of these properties, and may structure them, the assumption that an interoperability artifact might only support a single value for either one seems reasonable.

Immunization identifies a provider “type,” without further specification. This may be appropriate, as different specifications may have different needs, but the use cases don’t seem to imply a requirement, so it’s not clear what purpose the value serves, or when a specification might be considered to have fulfilled the requirement.

Immunization identifies a set of provider credentials, also underspecified. In this case, however, the ambiguity is complicated by the text definition: “provider licensing credentials.” This seems to differentiate the value from the license itself but constrain the credentials to those that support licenses held.

Immunization also defines a deeper class graph for provider, associating the provider with organizations, which may have locations, addresses, and “vital records organization” status.

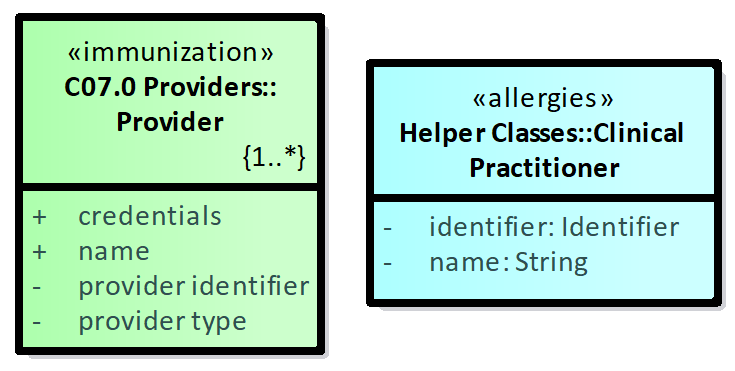


Figure 4: Provider domain comparison

## Domain: Patient

The Allergy DAM represents a patient as a class with a name and an identifier. As with provider, the cardinality is assumed to be 1:1, which may be unrealistic for an EHR but acceptable for an interoperability specification.

The Immunization DAM supports a richer model, making Patient a specialization of Person. Person has a dozen attributes, some with multiple cardinalities, and relationships to collections of structured properties including name, address, and identifier.

The Immunization DAM also uses the aggregation relation to associate a person with a “responsible person,” but the relationship is labeled “is-a” and should probably be a specialization. (A similar pattern is seen in the Organization package.)

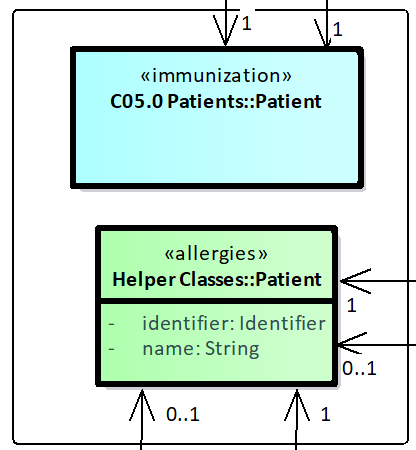


Figure 5: Patient domain comparison

Note that the FHIM makes Patient a role played by a Person, so the patient name is actually Patient.subject.name, a shown in Figure 6.

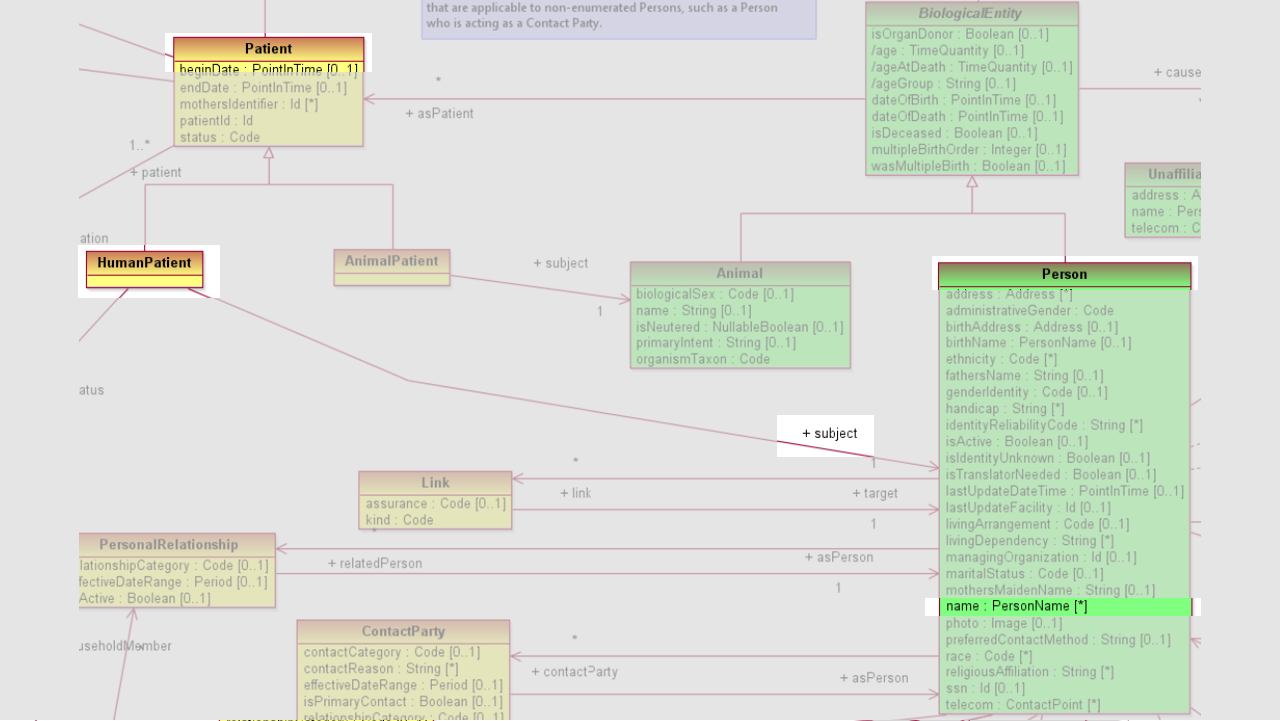


Figure 6: Patient name in FHIM

Figure 6 also demonstrates that, while the FHIM incorporates the requirements identified in the respective DAMs, the detailed view may not be the best tool for engaging subject matter experts.

## Domain: Adverse Event

The Immunization model has a simple model: an immunization encounter may be associated with an AdverseReaction, which has a date, a report date, and an “adverse event” property, presumably the symptom experienced.

The Allergy model is more detailed, distinguishing an AdverseReaction from its potentially multiple signal Manifestations.

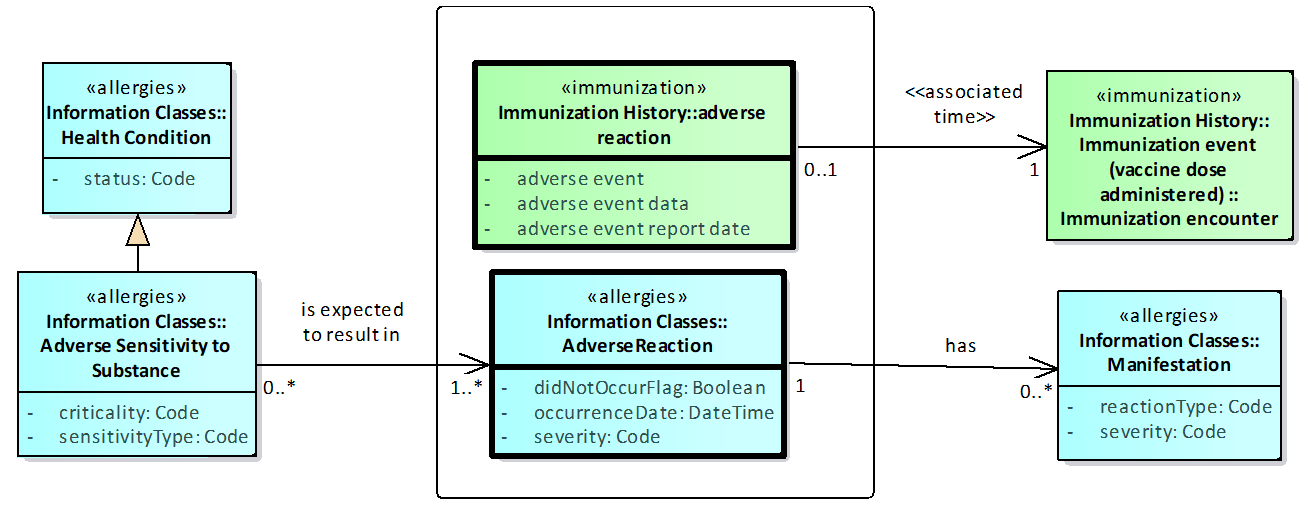


Figure 7: Adverse Event domain comparison

## Domain: Substance

The models have very different substance models. The allergy model is agnostic to the category of substance, relying on the variety of available coding systems to identify the correct kind of substance at the correct granularity. The Immunization model is concerned with a circumscribed set of predictable substances, and is able to specify both inheritance and supporting relationships.

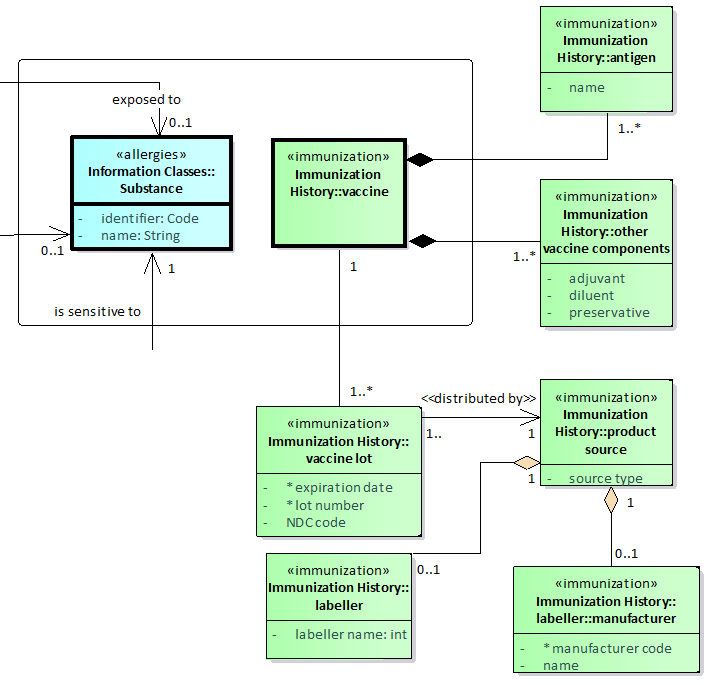


Figure 8: Substance domain comparison

## Naming conventions

Currently published DAMs use different assumptions when constructing class and property names. Some use terms in title case (capitalizing all words), some in lower case, and some in camel case (capitalizing words and removing spaces).

Capitalization provides an indicator of the kind of artifact. Many models use upper case for class names and lower case for property names. Spaces are useful for human legibility, but they can confuse machines, so artifacts that may be used to generate implementable specifications tend to remove spaces, either using CamelCase or inserting underscores between words in a name.

We find that no one design is best for all uses: the summary below suggests that camel case may provide an adequate compromise.

|  |  |  |  |
| --- | --- | --- | --- |
| **Approach** | **Example** | **Human legibility** | **Machine legibility** |
| Lower case | provider organization | ➋ | ➌ |
| Title case | Provider Organization | ➊ | ➌ |
| Sentence case | Provider organization | ➊ | ➌ |
| Camel case | ProviderOrganization | ➋ | ➊ |
| Underscore | Provider\_organization | ➌ | ➊ |

Table 2: Spelling conventions for names

Note that UpperCamelCase is often used for classes, lowerCamelCase for properties.

The Immunization DAM uses sentence case; Allergy uses a mix of title case and camel case. These differences complicate efforts to use common names, but they don’t present obstacles to comprehension. The FHIM uses upper camel case for classes and lower camel case for properties and relationships.

All three models use singular nouns for classes and properties. Both DAMs use both finite verbs and participles for relationships; the FHIM uses nouns.

## Documentation

Analysis models vary widely on the extent and formality of the defining assets they provide.

Both DAMs provide brief textual descriptions of the meaning of element names, but neither provides similar support for classes or relationships. The only definition is what is implied by class names and associated properties – two (and three relationships) for Allergy’s “Clinical Practitioner”, four (and four relationships) for Immunization’s “Provider.” The FHIM defines classes as well as properties.

None of the models provides a rationalized description of how information artifacts support use cases. Support can be inferred; e.g., Immunization use cases define tasks such as “request immunization history,” which may be fulfilled by the “Immunization history” class graph, but the equivalence may not be intended to be precise. The “report” capabilities are clearly more diverse, and they have no corresponding class graph.

None of the models consistently provides references to clinical documentation supporting the definitions or other modeling decisions, though some of the allergy use cases include references. Neither DAM specifies semantic definitions—e.g., model bindings to terminologies or value sets to illustrate valid values for property instances; the FHIM provides value bindings.

Neither DAM provides tracing from requirement to implementation specifications. This is normal in a waterfall process, where any traceability would be documented from the specification back to the prior source requirement. The FHIM, as a repository of specifications, keeps all this in one place.

The FHIM provides documentation of class and attribute definitions from its constituent requirement sources, and it documents the sources of these definitions. As CIMI models are added to the FHIM, CIMI documentation will follow, including model bindings (and implicit semantic definitions) and, according to the metadata strategy, clinical references.

## States

A common feature of analysis models is a “state machine” that describes how key objects change over time. It’s even more common to imply the existence of such state machines by including properties named “status,” without formally defining the allowed states.

Status properties can be ambiguous, because the concept of state tends to be use-case-specific. A condition, for instance, may have a physiological state (latent, febrile, remission), a concern state (of concern, no longer of concern), and a state of clinical knowledge (suspected, confirmed, refuted), among other possibilities. The best names are as specific as possible about what the state measures; less specific names tend to resurface as arguments in specification discussions.

The immunization model does not make such a model explicit.

The allergy model includes a state machine. The Adverse Sensitivity class inherits the “status” property from the Condition class, but the state machine specifies values for Adverse Sensitivity, the specialization. The values seem to be specific to sensitivities, as stated by the text, but the property is assigned to the more general Condition.

The FHIM state machines are all implicit in the bound status value sets, taken from standard specifications.

## Data Types

DAMs use a host of different assumptions about data types. Some use existing types (e.g., V3 types), though this brings a significant set of technical assumptions into the analysis space. Some don’t assign types at all, as too technically specific for an analysis model. And some concoct their own types in search of a reasonable level of technical abstraction for an analysis model.

The Immunization DAM falls in the second group, using no types, and leaving it to the reader to determine what kind of information is implied by a property name. The Allergy DAM falls in the latter group, using a short list of five unspecified analysis types: Identifier, String, DateTime, Code, and Boolean. (It also uses a TimeStamp, but it’s not clear whether this implies a requirement distinct from DateTime or not.)

The FHIM types were originally a simplified version of the V3 types, but have been refined to closely support requirements for FHIR types.

The simplest harmonization would be to apply the Allergy types to the Immunization DAM. Another option would be to use the types already assigned in the FHIM.

The decomposition approach uses a small set of analysis types based on the Allergy types. The FHIM uses its defined types. Refinement of these types to support target specification types is managed by UML profiles.

## Relationships

Documentation of relationships may differ in many ways. They may simply have different names, UML stereotypes, or directionality. Much of this technical layer of specification is immaterial to the domain level of modeling.

Relationship names may indicate domain-specific ways of thinking about similar things. A vaccine is a kind of substance, and an immunization is a kind of exposure, and while it’s not terribly colloquial, it’s fair to say that “uses” (in this context) is a kind of “exposed to.”

* Immunization: Immunization event “uses” a Vaccine
* Allergy: Exposure has an “exposed to” relationship to Substance.

While there is clear value in using the same class for “patient” in both models, the value of making “immunization event” a specialization of “exposure” is less clear. It is clear that there is a limit to the value of generalization; such a correspondence may be more usefully manifest in a use case that identifies the immunization as an exposure.

Some relationships require a greater degree of license in inferring relationships not present in the topology of one DAM. In Immunization, adverse reaction has an “associate in time” relationship to an encounter, which is related to Patient via an Immunization History. Allergy does not even model an encounter; there, AdverseReaction has a “preceded by” relationship to an Exposure and an “expected to result in” relationship from a Sensitivity, which is a specialization of Condition, “recorded for” a Patient. It doesn’t deny that there is an encounter, but a general model can’t simply harmonize; it needs to support divergent networks of data outlining independent paths from reaction to patient. We propose that the encounter can be understood as implicit in the Allergy model relationship, possibly via Exposure.

Another difficulty with relationships is direction. In the Immunization model a Patient “has” a Condition; in Allergy, a Condition is “recorded for” a Patient. In the analytical view, it makes no difference which way the relationship points, so a harmonized analysis model can portray either perspective. This flexibility does mean that, when taking analysis into the design layer, it’s important to understand the technical use cases the model is intended to support. A reaction might be the primary subject of an adverse event report, having a reference to a patient, but a patient care coordination record should also refer to allergies and reactions.

A third issue is the class boundary. Both DAMs specify a person property “name,” typed as a string in Allergy and structured as a class with five properties in Immunization. Any implementable specification will require that the name be structured to support at least first and last names. The FHIM defines a PersonName type with seven properties that performs this work in a way that supports generation of structured specifications. In addition to parts and a type, the FHIM supports a valid date range to support FHIR specifications.

## Terminology

Most DAMs intentionally defer terminology definition to the specification phase. The one exception is that the Immunization model specifies that a vaccine lot has an NDC code. This is the FDA’s system for identifying specific packaged products. It might be more appropriate to describe the property as a package class identifier to avoid realm-specific requirements; in an analysis model, the use of “NDC” as a shorthand has limited ill effects.

We feel that it is appropriate to defer terminology *binding* the specification stage, but the *semantic* specificity that terminologies provide can be usefully defined at the analysis level.

While the DAMs do not provide this dimension, the FHIM provides bindings to terminologies specified by component specifications. Whether these bindings can or should influence “upstream” to the DAM is for discussion, but it can be useful to identify where specifications diverge unnecessarily.

## Harmonization options

Our first question was the meaning of “harmonization.” The HL7 process refers to the work of managing change to the RIM; other SDOs adopt the more general meaning of identifying divergent designs and converging on designs that support all stakeholder requirements. However, without an agreed process, we wanted to demonstrate the various stages that might be of value. We include three stages: the initial illustration of overlap, a design that supports exactly the requirements identified in the respective DAMs, and a view of a model that already supports these requirements as well as other supporting assets.

The first view is illustrated above, and in Appendix A: it suggests synonymy to support discussion and agreement; it does not assert a design. The last view is a filtered view of the FHIM; it requires no design work other than the technical support of the view.

The middle option faced the question of what a DAM-scoped intermediate representation should look like. We identified three options. First, we could select one DAM as the baseline, and enhance it as necessary with new elements from the other. We identified no clear criterion for making such a selection, though making an arbitrary selection would be an option. Second, we could build a new DAM that represents super-classes of the content from the source DAMs. This seemed to be a path to creating additional opportunities for divergence, the very problem we were trying to solve. Or, third, we could build a foundational model that provides elements to two distinct views, each representing the content of one source DAM. This path is illustrated here.

It was not clear whether to decompose every property or only the shared ones. In figure 4, every property of the respective adverse reaction models is decomposed into a “detailed clinical model” pattern of representing all properties as classes.



Figure 9: Completely decomposed adverse reaction properties

They are recomposed into their DAM forms in figure 5.



Figure 10: DAM reactions composed of atomic elements

In this case, we deem the immunization properties AdverseReaction.adverseEvent and AdverseReaction.adverseEventDate to be synonymous with Allergy DAM properties Manifestation.reactionType and AdverseReaction.occurrenceDate. The alignment is questionable, as the “adverse event” of Immunization may have broader semantics than the coded property from the Allergy DAM. We anticipate that the proposal may generate constructive feedback one way or the other: whether they are the same, how they differ, and how this difference might affect specifications used in overlapping use cases.

We observe that properties not common to both DAMs don’t need to be harmonized in this way. The decomposed model might only contain the overlapping elements, leaving those elements that are specific to one DAM alone, as we illustrate in figure 6.



Figure 11: Immunization reaction using partially decomposed model

In this figure, the ImmunizationAdverseReaction class uses two common properties drawn from the decomposed model, ReactionType and OccurrenceDate, but the third property is unique to the Immunization DAM and not present in the Allergy DAM: it is modeled directly in the ImmunizationAdverseReaction class. This approach reduces the amount of work involved in migrating model content.

Two issues complicate this approach. First, in addition to the common elements, the decomposition must model their relationships in a way that supports both models. As we note above, the respective assumptions about relationships and class graph paths may be commensurable even if they differ in their explicit diagrams; the decomposed model must define this common view. Hence, the decomposed side of figure 6 shows four classes, not two.

Second, we anticipate that as other DAMs are brought into scope, more of the “unique” elements may find analogs in other models; event report date, for instance, is sure to have an analog in the Adverse Event Reporting model.

All of these questions become moot if the harmonization vehicle is a model that already defines all of the properties and their relationships. The FHIM traces requirements not only from implementable specifications but also from DAMs. The primary difficulties the FHIM presents are procedural rather than logical. In addition to questions around organizational stewardship, a key challenge is the task of configuring FHIM diagrams to show domains of interest, without distracting details from other domains or technical layers.

## Requirements traceability & currency

In a waterfall development methodology, requirements are defined, and design artifacts refer back to them. This way, a report can demonstrate an artifact’s success at covering expected requirements. However, if other requirements are discovered during design or development, they typically don’t find their way back into the requirement specification, which lapses further and further into obsolescence.

In an iterative methodology, requirements are expected to be uncovered at design, development, and even testing stages, so they must be documented in a way that supports two-way tracking. Keeping these references up-to-date requires a mutable requirements document. Keeping it current is further complicated when the requirements support multiple design specifications.

Following the HDF, neither DAM even attempts forward tracing. And while the derivative specifications in CDA templates and FHIR resources may incorporate insight from the DAMs, neither incorporates recognition of support for specific requirements, so it’s hard to tell when divergences are based on error, error correction, technology constraints, or new knowledge.

The FHIM captures forward tracing to C-CDA, NIEM, V2, and FHIR. It does not at this point support an SDO-grade process for establishing consensus on the requirements.

# Discussion

Modeling style differences complicated this analysis. We believe that more consistent guidance on analysis modeling would be beneficial. Names may differ irrespective of such guidance: different teams will bring different experiences and assumptions to the task. But consistent expectations around definitions, data types, and semantics would seem to help all consumers of the DAMs.

We also identified several issues with adopting the HL7 web-based Enterprise Architect tool. These are mostly orthogonal to our purpose, but they affect the ease with which the work can be done. They are cataloged in Appendix D.

The two DAMs model some content that is clearly different, and some that seems manifestly the same, but there are some cases that seem to be arguable. In these cases, we find it is usually the case that one model is simply giving more detail than the other, whether in terms of number of properties of a class, relationships of interest, or granularity of decomposition.

The fact that the cases support processes that any general practitioner might perform suggests that apparent synonymy is likely to be correct. I.e., while it may not be practical to make immunization a literal specialization of exposure, it is true that both events happen between a GP and a patient, and that the immunization, in such a case, is in fact an exposure. The equivalences we propose are subject to review, especially where we assert that a general structure in one model corresponds to a more specific structure in the other. We solicit feedback on these assertions wherever they may be found inadequate.

We observe that many elements do exhibit synonymy, and we conclude that the specifications built on these DAMs may miss opportunities for reuse. The vaccine underlying an adverse event, for instance, is documented in more detail than the corresponding substance recorded for an allergy record, but there is no reason that the substance shouldn’t be inferred, or at least suggested, based on the vaccine of record.

On the whole, these domains seem semantically quite commensurable, and we expect that a consolidated repository offers value in the recognition of common properties—for reusing those that are sufficient for both, and for testing and refining those that may not be, and for supporting the definition of implementable specifications that leverage these synergies.

We further observe that the value of the DAM is compromised by the timeframe of the standards development process. The DAM can provide guidance to specification teams, but those teams typically identify requirements not anticipated at the analysis stage. These requirements do not find their way back to the DAM, and they must be re-discovered for future specification efforts—in which they will likely be re-discovered slightly differently. FHIR and C-CDA, for instance, identify semantically identical requirements for allergy criticality, but they define value sets from different systems.

One advantage the FHIM offers is tracing to specifications. Because the FHIM is built on both DAM and specification inputs, it provides two-way tracing. If an analysis requirement is modified or refined, it is immediately clear what design artifacts should be reviewed. Furthermore, if a design specification is modified or refined, the relevant requirement can be identified, and other specifications referencing that requirement can also be reviewed. Such a cross-reference across domains and across development stages supports the possibility of maintaining a repository that reflects requirements and is also current.

Work at detailed levels can often seem too technically restrictive at the analysis phase; analysts should not have to choose among technical paradigms. However, understanding what specifications a data element supports provides insight to the subject matter experts that may help them establish designs that better support all use cases, especially the cross-domain ones. Knowing, for instance, that a coded element implemented in CDA may have a translation, or that an identifier may have no only an assigner but a use, may elicit more detailed requirements than those found without this sort of prompting. Existing specifications are useful and germane tools in the task of requirements elicitation.

# Conclusions

We propose that it would be of value to model classes common to multiple domains in a way that supports a common understanding. Such an understanding would have several benefits.

1. Common elements would need to be reviewed but not reinvented when new domains join the effort, or when new requirements prompt reassessment of existing designs.
2. Prior work will enrich new work not only by reducing the required effort but by suggesting facets of analysis that may not have been the primary concern of the available subject matter experts.
3. Specifications that support multiple domains will be able to use data requirements that already support the domains in question, rather than having to design new integration rules.

In addition, we propose that the model support traceability of requirements to and from specifications.

1. Traceability will ensure that, when requirements are discovered or modified in steps 1 & 2, their potential impact on specifications can be easily understood, and that when specification efforts identify new requirements, they can be more easily integrated into the requirements specification for comparison to other use cases.

The need to keep the analysis model in a mutable state will present challenges to the standards development methodology but may provide the ability to manage assets to be both consensus-based and current.

Finally, given the scenario of a cross-domain repository of coordinated and current data element requirements, we propose that such a model be used to generate specifications that are harmonized by definition or, where extended, are extended in an environment where the necessity is clear and the new extension is available to other domains.

Given these three proposals, we assert the ability to support cross-domain use cases with harmonized specifications. Adverse event specification designers should not be confused by input DAMs using different domain terminologies, ad hoc choices of boundaries, or use-case-driven granularities. A consolidated DAM will provide these authors of new content with harmonized elements. These will support the ability to develop implementable specifications that support current data elements from the domain of interest and from neighboring domains as well.

In order to illustrate this value, we plan to design a use case for demonstration that builds upon this work by demonstrating how data captured in specifications designed to support one domain may support the generation of data to support additional domains. This case will involve a reaction to an immunization resulting in an allergy record.

We solicit feedback on how to

* Address issues with the US ‘federal’ origins of the FHIM
* Address concerns with weighing the cost of analysis against the cost of divergent specifications and identifying criteria for deciding which path to take
* Balance standards consensus with DAM currency & develop a responsive methodology for managing requirements
* Share FHIM to make it easy to do the right thing, both in requirements analysis and in specification development

# Appendix A: Gap analysis

Image2

Image2

Image2

Image2

# Appendix B: Analytical Decomposition

# Appendix C: Filtered FHIM

F

# Appendix D: Tooling issues

* 1. **CM Issue**: difficulty finding legacy Sparks \*.EAP for the 2012 PHER Immunization DAM.
     1. **Discussion**: The PC Allergies and Adverse Events Sparx EAP file was kept with the HL7 published DAM. The PH 2008 IZ EAP file was found by the co-chairs; but the 2012 version had to be redone to allow analysis of the published 2012 IZ DAM.
     2. **Conclusion and recommendation**: At a minimum the source content of a standard should be put in GitHub for safekeeping and configuration management. Better yet, the EAP files should be configuration managed within the Sparx WebEA server for easy access.
  2. **Modelling Issue:** Inconsistent modelling style and format:
     1. **Discussion:** The two DAMs were developed by separate workgroups at different times. It is no surprise that the look and feel vary substantially.
     2. **Conclusion and recommendations:** A DAM style guide is needed.
  3. **Tool Issue:** Sparx EA management of multiple models can be challenging:
     1. **Discussion:** the import of multiple models into a single instance of EA requires properly exported XMI files, which were specified to be imported as a root node. There are many internet resources on UML and the basic button-ology of Sparx tools.
     2. **Conclusion and recommendations:** A HL7 WebEA administrators guide needs to be maintained to make the management of multiple Sparx EA and Sparx WebEA models easier and safe for novice users. Frequent backup is advised to ensure inadvertent model damage is easily recoverable.
  4. **Presentation Issue**: Distinguishing class heritage on diagrams
     1. **Discussion:** Having multiple independently developed models in the same tool resulted in it being confusing on a diagram to know a class’s origin.
     2. **Conclusion and recommendations:** UML Stereotypes, keywords and color be used to indicate the origin of classes and on diagrams, showing the stereotypes and colors within the classes, as appropriate.
  5. **Data consistency Issue**: Data Type Inconsistencies
     1. **Discussion:** The Immunization DAM used no data types,
        1. The Allergy DAM uses Boolean, Code, DateTime, Identifier, String, Timestamp
        2. UML primitive data types are Boolean, Integer, UnlimitedNatural, String and Real
        3. HL7 FHIR, C-CDA and V2 each have their own data types
     2. **Conclusion and recommendation:** Workgroups might start ny limiting themselves to the Primitive UML data types. If the target implementation context and paradigm is known, they might use the appropriate data types and value sets for that environment, e.g., FHIR>
  6. **Use of UML “Alias”**
     1. **Discussion:** Initially, the use of alias appears to reconcile seemingly similar classes with different names. The problem is that each class can only have one alias, which cannot be changed based on a keyword or stereotype.
     2. **Conclusion and recommendation:** UML class name alias is not a suitable across multiple healthcare domains, e.g., provides, clinician, medic, practitioner, etc.
  7. **Use of UML Stereotype**
     1. **Discussion:** The use of stereotypes can identify and filter different classes and attributes across different domain models. The problem is that each class and/or attribute can only have one stereotype, which limits the use of an attribute in let’s say 2 out of 3 domains.
     2. **Conclusion and recommendation:** UML class name stereotype is helpful to distinguish classes across multiple healthcare domains, e.g., provides, clinician, medic, practitioner, etc.
  8. **Use of UML keywords**
     1. **Discussion:** Classes can have multiple keywords.The use of keywords can identify and filter different classes across different domain models. Attributes cannot have associated keywords.
     2. **Conclusion and recommendation:** UML class keywords provides a good way to be able to filter classes in a consolidated model. If attributes are modeled as associated classes, then keywords can work well for all situations. I believe this has been referred to as the Michael van der Zel style of modelling, by Galen.
  9. **Flattened inherited attributes and operations**
     1. **Discussion:** Sparx EA has the capability to flatten inheritance hierarchies on drawings. By right clicking on a class, select the compartment visibility option. The attributes and/or inherited attributes plus operations and/or inherited operations can be made visible or hidden. Inheritance makes information consistent, but may make diagrams complex to understand, especially when the inheritance is shown on a different diagram.
     2. **Conclusion and recommendation:** Flat structures are used in FHIR, C-CDA, etc. for simplicity. Flattened classes in the published DAM diagrams may make them simpler to understand, as shown in Diagram 10a FHIM Patient versus Diagram 10b FHIM Patient (inherited attributes).
  10. **Associated Role and subtype modelling style duality**:
      1. **Discussion:** Diagram 7 illustrates how a HealthcareProvider can have a subtype of an IndividualProvider with a role of a Person. The result is that an IndivisualProvider attributes can be specified multiple ways, as illustrated in this appendix.
      2. **Conclusion and recommendation:** Associations, subtypes and data types are equivalent ways to express differences among classes (classifiers). When modeling the structure of your system there are basically two ways to express a structural relationship between two Classifiers. You could use an Association between the two Classifiers, or you could create an Attribute owned by one Classifier with it’s type set to the other Classifier. A data type is a special kind of classifier, like a class. It differs from a class in that instances of a data type are identified only by their value. In recommendation the artform of modelling is to select the simplest structure to tell a viewpoint’s story to its intended audience.