



NIEM High Level Version Architecture

NIEM Technical Architecture Committee

**July 31 2008
Version 1.0**

Record of Changes

No.	Date	Reference: All, Page, Table, Figure, Paragraph	A = Add. M = Mod. D = Del.	Revised By	Change Description
1.0	July 31 2008	All	A	NTAC	Initial version

Contents

1	Introduction	1
1.1	Scope	1
1.2	Background.....	1
2	Objectives	3
2.1	Not Objectives	3
3	Overview	5
3.1	Classes of Information.....	5
3.2	Data Stores.....	6
3.3	Activities.....	6
3.4	A Walkthrough of Versioning	7
4	Activities.....	10
4.1	Domain Reconciliation	10
4.1.1	NBAC Decisions.....	11
4.1.2	Coherence.....	12
4.2	NIEM Micro Releases	13
4.3	Cross-Domain Harmonization.....	14
4.4	Core Synchronization	14
5	NIEM Releases	15
5.1	Properties of a NIEM Release	15
5.2	Status of a NIEM Release.....	15
5.3	Version Identifiers for NIEM releases.....	15
6	Data Stores.....	17
6.1	The Release Area.....	17
6.2	The Collaboration Area	17
6.3	The Publication Area	17
6.3.1	Partitions of the Publication Area	18
6.3.2	Use of the Publication Area	18
6.3.3	Namespaces for Content of the Publication Area	19
6.3.4	Additional Requirements of the Publication Area	19
6.4	The Issue Tracking Area.....	19
7	Additional Artifacts	21
7.1	Release Manifest.....	21
7.2	Change Log.....	21
7.3	Namespace URI Directive	22
8	Summary.....	23
	Appendix A: Index	A-1

Figures

Figure 1: Activity flow for the NIEM versioning processes	7
Figure 2: Schema set that is not coherent	12
Figure 3: Coherent schema set.....	13

1 Introduction

This document describes the NIEM version architecture at a high level. It describes the approach and the major artifacts involved, without specifying them at a detailed level. This document is intended to frame further discussion and specification efforts, leading to a precise specification in the future.

The version architecture discusses issues that have inherently conflicting tradeoffs. It seeks a balance of concepts drawn from previous NTAC discussions and ideas. Major tradeoffs include:

- Timeliness versus Stability
- Autonomy versus Interoperability
- Ease for IEPD developer versus ease for Domain developer

1.1 Scope

The high-level version architecture describes how NIEM governance bodies update the data components and schemas that make up NIEM. This document is a high-level, abstract form of the eventual specification. It describes, at a high level, topics that will eventually be specified by a complete version architecture.

This document assumes that the reader is familiar with NIEM and its artifacts. Readers needing additional background on NIEM should refer to documents on the NIEM website at <http://niem.gov/>.

1.2 Background

The National Information Exchange Model (NIEM) is represented as a set of XML Schema documents (schemas) that define data components. Implementers of information exchanges use these data components to build information exchange packages. The goal of NIEM is to define common data components that are highly reusable, to improve the economy of implementing information exchanges, to increase comprehension of those exchanges, and to reduce errors in exchanges.

The NIEM Technical Architecture Committee (NTAC) and the NIEM Business Architecture Committee (NBAC) govern NIEM. The NIEM Program Management Office (PMO) is the executive body overseeing the development and maintenance of NIEM. The NBAC and NTAC may appoint standing subcommittees and transitory tiger teams who take responsibility for specific issues. Together, these bodies are referred to as the NIEM governance bodies.

A NIEM domain is an organization or body that works with NIEM governance to manage data component definitions for some community of interest or agency. A domain has an organization independent of NIEM, and has its own rules, procedures, schedule, and priorities.

The data components that make up NIEM are published as a set of NIEM releases. Each NIEM release is composed of a set of schemas. Within a NIEM release, the NIEM Core schema contains the content of the release that is common or universal. Additional schemas within

NIEM releases include domain schemas and code table schemas. Each domain or code table schema has a body responsible for its content, with each group having its own timeline for update and publication. For this document, the term "domain" applies to domain working groups and to code table working groups.

For more information on what is included in the NIEM Core, what domains and code table schemas have been defined, and what is included in these domain and code table schemas, see the NIEM website at <http://niem.gov/>

2 Objectives

The architecture described by this document should achieve to the following objectives:

- Each domain may publish updates on its own timeline. This publication is not delayed by waiting for the NBAC review.
- Domain updates are quickly available for use by IEPDs (Information Exchange Package Documentation). A domain may accommodate IEPDs for its domain as well as cross-domain IEPDs. Updates may be used by IEPDs without being delayed by an update/synchronization/harmonization process.
- Domain updates are incorporated into the next NIEM release.
- IEPD developers are provided with a NIEM release, which is an updated schema set that is coherent, for increased usability.
- IEPDs have the flexibility to use NIEM components as required to satisfy their business requirements. An IEPD may reference components from one or more NIEM releases, as well as other published content.
- There is a specific, concrete path for domains to provide input to the NBAC's update/synchronization/harmonization processes, for inclusion in future NIEM releases.
- There is a reliable schedule for (1) domain reconciliation activities, (2) the minor releases produced by domain reconciliation, (3) Core synchronization activities, (4) the NIEM major releases produced by Core synchronization. Providing schedules ensures that participants in the activities know when their input is due and timelines for required work. It also ensures that users of the releases can plan for release dates.
- All changes are visible: Each namespace URI is used for exactly one version of a schema, so any change results in a new namespace URI. In addition, change logs support descriptions of changes made to namespaces.

2.1 Not Objectives

The following were not objectives of the NIEM High Level Version Architecture:

- Reuse of a single schema target namespace Uniform Resource Identifier (URI) for multiple versions of a schema. The methods of the HLVA create new schemas with new URIs for changed, modified, and otherwise updated components.
- Definition of backwards compatibility or forwards compatibility between XML Schema documents. The concept of compatibility between schema definitions is difficult to maintain without sacrificing precise schema validation. The HLVA selected methods that allow for easy migration between schema versions, while maintaining precise schema validation based on specific schema definitions.
- Description of specific criteria for harmonization and best practices for design of the contents of schemas. These will be addressed in other NIEM documents.

- Description of candidate releases and prereleases. The NIEM HLVA is concerned only with schemas and supporting artifacts that are durable, and for which there must be long-term support. The use of candidate releases and prereleases will be addressed by recommendations on domain governance and release strategy, which will be the subject of other NIEM documents. Additional methods and functionality within the collaboration area to support prereleases will be described by the final version architecture specification.

3 Overview

The high level version architecture provides a framework for changes to be performed, in a systematic way, on a set of artifacts. These are outlined in subparts of this section.

The activities that lead to changes in the schemas and other artifacts are driven by a desire to improve the model. Improvements include correcting errors, meeting previously unforeseen requirements, and adapting to new needs and new uses of the model.

One concern is how we define *improvement* to the model. A common term for improving systems is harmonization, a term which we adopt for this effort:

Harmonization (of a schema or set of schemas) is the process of modifying a schema set in an incremental fashion for the purpose of improving the quality of the schema set with respect to some criteria. For example, a schema set may be harmonized to reduce semantic overlap of its components. Or, a schema set may be harmonized to ensure uniformity of the vocabulary used in definitions of its components.

Harmonization may be conducted within a single schema, or may be conducted across multiple schemas of a schema set.

The version architecture does not anticipate that NIEM will ever be completely **harmonized** (a theoretical perfect state for a schema set, in which all components perfectly meet all criteria). In practice, a fully harmonized state is never reached, and the goal of harmonization is incremental improvement, rather than perfection.

This section outlines the basic structure of the architecture. Many details are in later sections.

3.1 Classes of Information

There are several classes of information that are employed by the version architecture. These include:

1. NIEM releases. A NIEM release is a set of schemas that is published by NIEM. Each of these schemas defines data components for use in information exchanges. A release should be easy to use, and should define each concept in a coherent way. Each release is independent of other releases, although a schema may occur in two releases. A release is of high quality, and has been vetted by NIEM governance bodies.
2. Published updates. A published update is a schema or set of schemas that is issued by a domain or by a NIEM subcommittee or tiger team. An update may be new content or may be an update of content that was previously included in a NIEM release. A published update may define new versions of content from NIEM releases or other published content. The issuing body vets each update before its publication, but the update is not subject to review by other NIEM bodies. A published update must be conformant, but otherwise it has fewer constraints on quality than does a NIEM release.
3. Issues. Users of NIEM schemas and components will find problems that should be addressed. These will include erroneous syntax or definitions, inconsistencies between

different parts of the model, and new requirements that are not yet satisfied. An issue is a record of one of these problems. An issue's record may include discussion of the problem and possible solutions, as well as changes made to the model to solve the problem.

3.2 Data Stores

There are several data stores in the architecture. General terms are used here, to avoid introducing any requirements or restrictions associated with an existing tool, such as a registry, repository, or specific web software such as Bugzilla and Sharepoint. This description seeks only to describe requirements and functionality for these data stores.

1. Release area: The release area stores NIEM releases, and makes them publicly available.
2. Publication area: The publication area stores and makes publicly available domain updates and artifacts published by tiger teams.
3. Issue tracking area: The issue tracking area supports registration, modification, and discussion of issues pertaining to NIEM artifacts. This may be or include the NCCT (NIEM Configuration and Control Tool).
4. Collaboration area: The collaboration area supports cooperative editing of artifacts, as well as discussion by domains and other NIEM working groups. Each domain or working group receives its own partition within the collaboration area. These partitions are private, as access to each partition is restricted to the members of the appropriate working group.

3.3 Activities

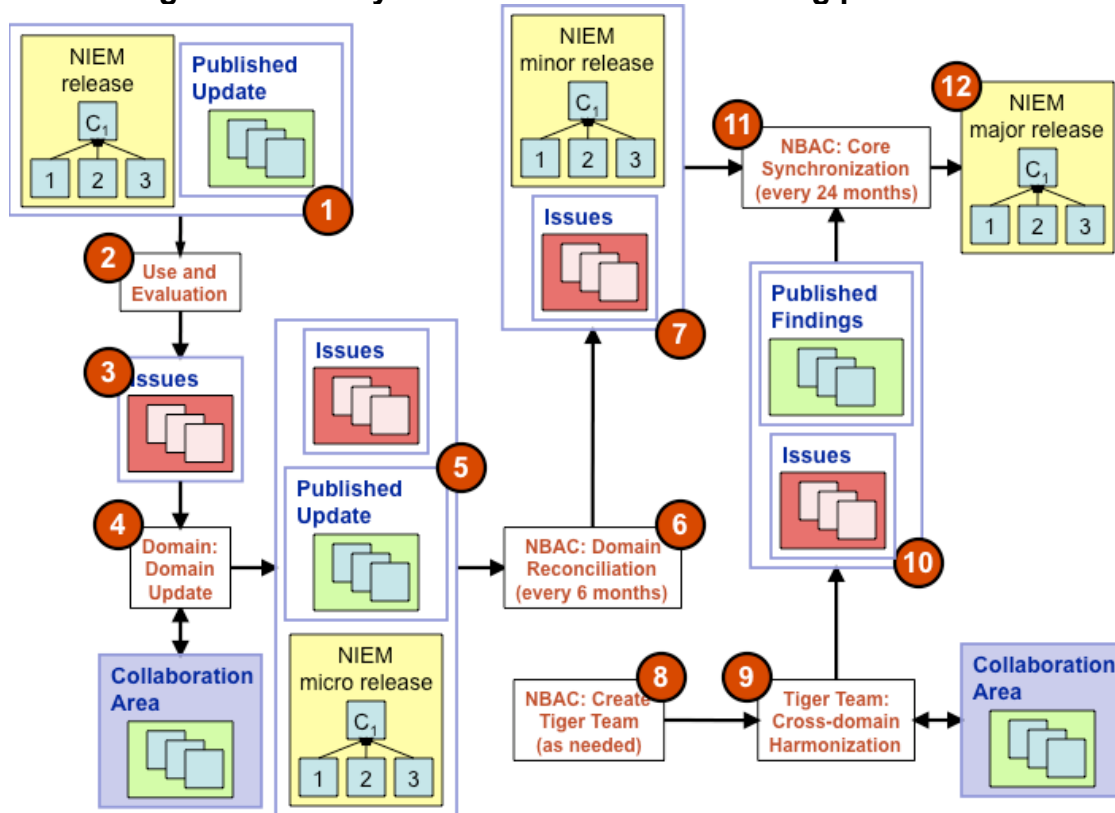
There are 4 major activities involved in managing versions of NIEM. These are the processes in which the domain managers and NBAC and NTAC chairs are engaged, as they improve and extend NIEM.

1. Domain update: A domain may publish updates to its schemas. A domain update is published to the publication area. A domain update proposes updates to future NIEM releases.
2. Domain reconciliation: The NBAC will resolve conflicts between domain updates, through a process called domain reconciliation. This process results in a reconciled, coherent schema set. This schema set is published as a minor NIEM release.
3. Cross-domain harmonization: The NBAC will initiate tiger teams and working groups to resolve inconsistencies, overlaps, and other semantic issues between the domains, and between the domains and Core. The results of these efforts will go into the publication area, to be included in the next major or minor NIEM release, as appropriate.
4. Core synchronization: The results of cross-domain harmonization are merged into a new release of the NIEM Core namespace, with domains synchronized to the new NIEM Core. Together, these form a major release.

3.4 A Walkthrough of Versioning

The NIEM versioning process contains several activities, performed by various parties. This section walks through these activities, and highlights the inputs and outputs of each. Each activity and artifact is discussed in more detail in later sections.

Figure 1: Activity flow for the NIEM versioning processes



Here is a brief high-level walkthrough of versioning, describing the processes:

1. NIEM releases and domain updates are available for use in IEPDs and other use.
2. Through use and analysis of NIEM releases and published content, domain users will identify problems and new requirements for domain content and Core content. Users include developers of IEPDs, as well as implementers and users of exchanges.
3. These problems and requirements will be entered and tracked as **issues** in the **issue tracking area**.
4. NIEM domains use issues as the basis for incremental improvements, extensions, and proposed changes to NIEM releases. This process is called **domain update**. Domains may work together via a partition of the **collaboration area**
5. Domains publish their domain updates in three ways: (1) Issues are updated and new issues are entered in the issue tracking area. (2) Domain schemas are updated in the publication area. (3) Optionally, domains may release coherent schema changes via a

NIEM micro release, published in the **release area**. A minor release is often domain specific, and has a 3-digit version ID, such as 3.10.2.

6. Periodically, the NBAC reconciles domain updates into a new NIEM minor release. The goals of this process are (1) to resolve the conflicts and inconsistencies that are created by new and modified domain content, (2) to produce a coherent set of reference schemas for the domains, and (3) to perform some incremental harmonization on that schema set.

This process is called **domain reconciliation**, and is repeated approximately every 6 months. In this process, the NBAC examines issues, published updates, and releases, and reconciles changes to domains into a single coherent domain schema set. The NBAC ensures that all domains reference each other in a consistent way.

7. The result of this reconciliation process is published as the next NIEM minor release. This release is coherent. This release has a version ID. This release is called a NIEM minor release, and it contains no modification to the NIEM Core namespace. It may contain some additional Core content (in a new namespace), but will not modify the NIEM Core namespace. This is published approximately every 6 months, as the result of domain reconciliation. A NIEM minor release has a 2-digit version ID that does not end in "0", such as "4.2".

For those issues that are not resolved via domain reconciliation, issues in the issue tracking area are updated, created, and closed, as appropriate.

8. At any time, asynchronously with other processes, NBAC or NTAC may initiate a tiger team to conduct cross-domain harmonization. A **tiger team** is a small working group of subject matter experts (SMEs) formed to solve a specific set of problems. A tiger team exists only for a short time; it is disbanded once it has served its purpose.
9. A tiger team that is formed by the NBAC or NTAC will solve some set of problems with the NIEM model. For example, it may resolve a set of major harmonization issues between the domains and Core. Or, it may address a new requirement that requires thought and experience. Or, it may identify, within the domains, components with semantic overlap, and will pull them up into Core for common definition and use. This process is called **cross-domain harmonization**.
10. The results of tiger teams will be made available in a partition of the publication area. This partition may hold schemas that contain additions to and extensions of the NIEM Core namespace. Each schema is published with a unique namespace, distinct from the NIEM Core namespace. The domains may use these schemas in domain updates, and IEPDs may use these schemas, as well. The contents of these partitions of the publication area will be considered during the domain reconciliation process (step 6, above), and a supporting schema containing additional Core content may be published with the periodic minor release.
11. Approximately every 18-24 months, the results of cross-domain harmonization and domain reconciliation are brought together into a new NIEM major release. This process is called **Core synchronization**. The NIEM Core namespace is updated with help from issues in the issue tracking area, as well as the published updates from the domains and

tiger teams. Domain definitions that were dependent on the previous NIEM Core are harmonized with the updated NIEM Core namespace.

12. Core synchronization results in a new NIEM major release, with an updated NIEM Core namespace. A NIEM major release has a 2-digit version ID that ends in "0", such as "4.0".

4 Activities

This section contains further discussion and details of the major activities of the NIEM version architecture.

4.1 Domain Reconciliation

Once a NIEM release is published, updates and requests for changes accumulate. Users post issues (in the issue tracking area) describing problems with and unmet requirements of NIEM schemas.

In response to these issues, domains publish updates to their namespaces. They identify and define new content that they require. They define and update code tables that they require. These updates are published in the domain partitions of the publication area.

Approximately every 6 months, the NIEM Business Architecture Committee (NBAC) reconciles these requests with the latest published release, in order to produce a coherent NIEM schema set. This reconciliation will result in a schema set that is easily used by IEPD developers.

The NBAC makes decisions on how to accommodate changes. In the publication areas, domains publish updates prior to NBAC approval, ensuring autonomy of the domains. The NBAC determines how best to migrate these updates into the next NIEM minor release.

The NBAC conducts domain reconciliation approximately every 6 months.

This reconciliation process will consist of the following stages:

1. Gather and prepare input to the reconciliation process.

This input will come from open issues in the issue tracking area, and from domain partitions of the publication area. The content of the publication area will be evaluated for impacts. Sample reports that will be prepared may include:

- Reports describing the impacts of proposed changes on other domain namespaces.
- Reports assembling the comments that other domains have made about proposed changes.
- Reports containing technical metrics that quantify the (1) consistency of harmonization on the input schema set, and (2) the impacts of proposed changes.
- NTAC recommendations on technical methods and techniques for handling domain requests.
- Conformance and quality checks (Reference NIEM NDR and NIEM Quality Assurance Strategy and Plan). Conformance and quality checks are performed prior to publication of schemas to the publication area, but additional reports on quality and conformance may be made available to the NBAC for reconciliation.

2. Reconcile proposed domain updates.

Starting with the previous NIEM release, the NBAC will review and apply the changes that were requested by the domains. For each proposed domain update, approve the update, or modify it as required to maintain high quality across all NIEM schemas.

When a change is applied, make modifications in other domains, as required, to keep the schema set coherent. Where domain updates introduce integrity issues, modify the schema set to eliminate those problems. Propagate domain updates through impacted namespaces, as appropriate.

Note that there is no requirement or implication that any harmonization will be done by tools, or automatically in any way. The goal of this architecture is to ensure that the responsible party has the authority, schedule, goal, and ability to do what is needed to ensure that domain changes result in a coherent set of domain schemas that can easily be used in IEPDs.

4.1.1 NBAC Decisions

Through the process of domain reconciliation, the NBAC will resolve many issues of cross-domain conflict. This may include changes to one domain that affect another domain, or overlaps of content between domains. Through the reconciliation process, the NBAC will make adjustments as-needed to ensure that domain updates do not harm the quality of the NIEM release as a whole.

During its periodic domain reconciliation, the NBAC may conduct additional harmonization. Issues addressed may be "low hanging fruit", such as particularly easy or notable problems. Or, the NBAC may try to resolve one or more aspects of harmonization across the NIEM model, such as vocabulary consistency. During reconciliation, it is likely that time will be short, and the level of effort will be limited. So, during reconciliation, extremely broad changes would be avoided, as would changes that required significant consideration by subject matter experts. Such issues would be handled by tiger teams and small groups in a separate process, cross-domain harmonization.

The NBAC, as the designated body for harmonization, may make any decision that they deem prudent to integrate domain updates with the previous NIEM release. This may include the modification of dependent namespaces to accommodate domain changes, the creation of new elements for indirect (substitution) methods, and the modification of definitions and component names.

The NBAC may also choose to reject, in whole or in part, any domain update. Of course, the NBAC should have valid reasons for such action. However, the NBAC is given the responsibility of acting in the best interest of the NIEM community, rather than in the interest of a single domain. Note that no change the NBAC makes will affect any content that is published by the domains in the publication area, nor will any content from the release area be affected. That content stands as it is, without change.

Barring a valid objection against a domain update, the NBAC would adopt the update for the next version of the NIEM domain schema, and would propagate that change into dependent schemas, as required. This may require that the NBAC exercise judgment.

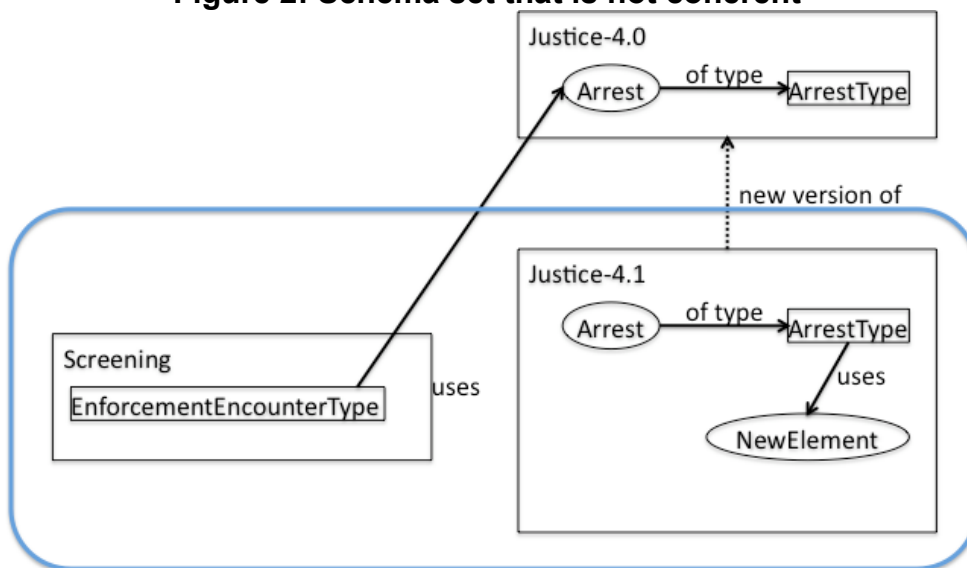
4.1.2 Coherence

This document uses the term coherence to describe whether components of a schema set refer to each other in a consistent manner:

A schema set is **coherent** when it has the properties that (1) the schema set does not refer to a schema outside the set, and (2) the set does not include two different versions of the same component in an incompatible way.

The coherence of a NIEM distribution is best described by showing a schema set that is not coherent. In this schema set, a Justice 4.1 version is based on Justice 4.0, with a new element ("NewElement") added to ArrestType. The latest schemas are outlined in the rounded boxes below. The schema set that is composed of these schemas is not coherent.

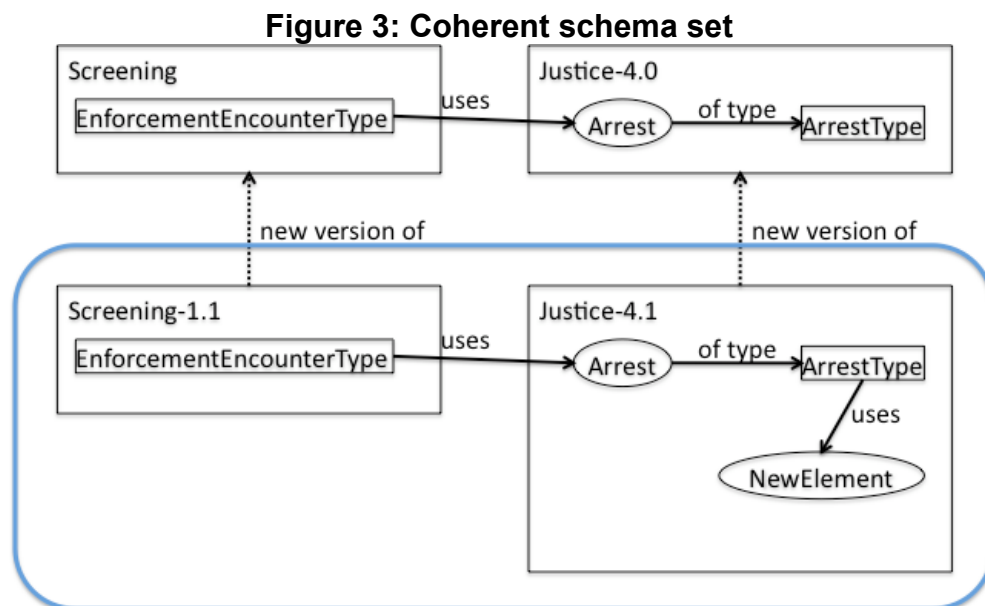
Figure 2: Schema set that is not coherent



Why is this not coherent? An IEPD would not be able to use the newest schemas (Screening and Justice-4.1) together as designed. Instead, the IEPD would require a substantial amount of work to make these two schemas work together.

For example, the developer of an IEPD might duplicate NewElement within its extension namespace. Or, it might create an extension of the 4.0 ArrestType that uses the 4.1 NewElement. Such changes would be required for each incompatibility between versions. In an environment where many domains are updating frequently, this presents IEPD developers with a substantial problem.

However, the NBAC could reconcile the request from the domain (adding "NewElement") with other requests from other domains. It could then produce the reconciled schema set, which would look like the following:



This schema set could be easily used by an IEPD developer. It would not require IEPD developers to develop custom extensions to fix problems in the schema set. The new Screening namespace (1.1) is integrated with the new Justice namespace.

The final version architecture specification will address the structure of version identifiers in the domain schemas. Specifically, it may be desirable for version identifiers for "bumped" namespaces to be easily distinguished from version identifiers for domain-initiated changes. The specific rules that apply to version identifiers of schemas will be defined by the version architecture specification.

4.2 NIEM Micro Releases

When a domain has created its updated content, it may determine that the content should be released as a coherent NIEM release. This will be verified by tools that check conformance and coherence of the resulting schema set. Once these checks are complete, a domain may elect to publish their update directly as a NIEM micro release in the release area, rather than as a published update in the publication area.

Schemas published in this manner may be more durable than schemas published to the publication area. That is, schemas so published may work better with more versions of NIEM. NIEM releases will not refer to schemas in the publication area; schemas from the publication area are reconciled and modified before their release, and they get new target namespace URIs. Schemas published directly to releases avoid reconciliation. They also will not have to be modified as they go from the publication area to the release area, as they avoid the first step.

A prime candidate, and simple example, of this method is a code table. By defining a code element such that it is substitutable for an element from a NIEM release, multiple versions of a

schema set may be usable at the same time. A new version of such a code table may be released as part of a NIEM micro release without requiring rework of other schemas.

NIEM micro releases require tools for validation and verification. They also require a commitment to frequent NIEM releases, with limited vetting.

4.3 Cross-Domain Harmonization

Larger and deeper issues of semantic consistency and modeling will be addressed by tiger teams and small groups of subject matter experts, as determined by the NBAC. The tasks undertaken by these teams will, in many cases, not require the entire NBAC to weigh in. The use of such teams will enable parallel harmonization efforts. This will result in greater scalability, as the number of domains increase.

The products of cross-domain harmonization will include (1) recommendations on domain updates in future minor releases, and (2) additional Core content provided via extensions and augmentations in new schemas within the publication area. The additions to NIEM Core that appear within the publication area may be immediately used by IEPDs, and domains may base published updates on those schemas.

4.4 Core Synchronization

Periodically, these products will be synchronized with domain content, to produce a new major release. A major release will contain a new NIEM Core namespace. A major release provides an opportunity to execute destructive changes, such as component deletions. At the time of a major release, other major changes to the NIEM release schema set may be considered by the NBAC, such as changes to structures and appinfo, which are schemas that are part of the infrastructure of NIEM. For more information on structures and appinfo, see the NIEM Naming and Design Rules and other documents on the NIEM website.

NIEM Core synchronizations will occur approximately every 18-24 months. However, a NIEM Core synchronization may be held back over a longer period if changes to be released do not meet a certain threshold, such as a certain number of changes, or changes of a certain severity. The NBAC will have to balance the need for regular scheduled releases with the set of components that must be harmonized across domains for inclusion into the Core.

In between updates to NIEM Core, the NBAC may add Core content via additional components in additional schemas in minor releases of NIEM, without updating the NIEM Core schema. These would be additive changes only, and would not modify the original NIEM Core schema in any way.

5 NIEM Releases

The NBAC domain reconciliation activity results in a NIEM minor release.

5.1 Properties of a NIEM Release

A NIEM release has the following properties:

- The release is coherent. That is, it contains only a single version of each domain namespace.
- The release integrates the latest input from the domains, with accommodations made by the NBAC to make them work together.
- The release contains new versions of only those namespaces that require update. Schemas that require no update will be part of the release without modification.
- If required, a namespace will be updated to accommodate changes in other namespaces. This is determined by the NBAC during the reconciliation process.

5.2 Status of a NIEM Release

Two types of harmonization/reconciliation: (1) between the domains (domain reconciliation), and (2) with NIEM Core and the domains (Core synchronization). This gives the opportunity to distinguish between the subsequent releases. Reconciliation of domains results in a minor release, while reconciliation of NIEM Core results in a major release.

5.3 Version Identifiers for NIEM releases

Domain-synchronized NIEM releases are given minor release version numbers. NIEM Core-synchronized NIEM releases are given major release version numbers. Direct releases due to domain updates are given micro release numbers.

When Core synchronization occurs, the results are published as a NIEM major release. These releases will be given IDs like 2.0, 3.0, and 4.0. These releases will yield a NIEM Core namespace that has the same version ID as the NIEM release.

When domain reconciliation occurs, without update of the NIEM Core namespace, the results are published as a NIEM minor release. These releases have IDs like 2.1, 2.2, 2.3, 3.1, 3.2, and 3.3.

When domain update occurs, without reconciliation, and the domain causes to be published a new release that incorporates that update, the results are published as a NIEM micro release. These releases have IDs like 2.1.1, 2.1.2, 2.2.1, and 2.2.2.

For NIEM release identifiers, major releases have two numbers (like 2.0). The major digit (2), and the minor digit (.0). The minor release following 2.0 is 2.1. The micro release following 2.1 is 2.1.1. There is no version 2.1.0.

For NIEM release identifiers, the value following 2.9 is 2.10. Each number (major, minor, or micro) is a field storing an integer; it is not a single digit.

The version IDs for released domain schemas (in the release area) and published schemas (in the publication area) are not described by this high-level version architecture. Constraints on these identifiers have not yet been determined, and will be specified fully in the final version architecture specification.

6 Data Stores

The version architecture defines several areas in which artifacts of NIEM are stored. Each may be implemented using new or existing tools. The goal of this document is to describe the areas, what information they contain, and what functionality is needed to support activities that use these areas.

The specific software and tools that make up these data stores is not specified by this document. Such specification will be made by later documents.

6.1 The Release Area

All releases of NIEM are published to the release area.

All schemas published to the release are persistent, and are available for immediate use by IEPD developers and other users.

A single schema may be a part of multiple releases. For example, a single release of the NIEM Core namespace will be a part of several minor releases. The release area will provide metadata to make clear exactly which schemas are associated with which releases. This metadata is referred to as the **release manifest**. The format of this metadata is yet to be determined.

This release area may be the current `niem.gov` website, but this is yet to be determined.

6.2 The Collaboration Area

The collaboration area supports cooperative editing of artifacts, as well as discussion by domains and other NIEM working groups. Each domain or working group receives its own partition within the collaboration area.

These partitions are private, as access to each partition is restricted to the members of the appropriate working group.

The precise mechanics and content of the collaboration area is to be determined.

6.3 The Publication Area

The publication area is a well-known location for domains and other working groups to publish schemas and other artifacts for NIEM. This has two major purposes:

1. It allows published content to be immediately used.
2. It provides a concrete method for changes to be expressed for domain reconciliation and Core synchronization.

All content of the publication area is *persistent*. That is, it never goes away once it is published. It also is never modified. Revised versions of published content may also be published, however, no artifact is removed or revised. This ensures that all content of the publication area may be freely used by any development effort, without fear of it changing or disappearing.

All content of the publication area is available for use by IEPD developers, as well as developers of other schemas and applications.

6.3.1 Partitions of the Publication Area

Each domain and working group controls a partition of the publication area. This partition is the publication site for products of the working group. The domain may place any number of schemas into its publication area.

6.3.2 Use of the Publication Area

Schemas within the publication area may provide additional content, or describe other changes to a NIEM release. Here is how the publication area is used:

- After a NIEM release is published, it becomes available for use by IEPDs. Through use and analysis of that NIEM release, domain users will identify problems and new requirements for domain content and Core content. These will be conveyed via the issue tracking area.
- The domain, under its own governance, determines what changes it believes should be made to the latest NIEM distribution. It expresses these changes by publishing schemas and supporting metadata in its partition of the publication area. Requests by the domain for changes to other domains and Core will be made through the issue tracking area and the NCCT.
- Schemas published to the publication area are immediately available for use by IEPDs. Both domain IEPDs and cross-domain IEPDs may use any schemas in the publication areas, as they require.
- The schemas, supporting metadata, and issues (in the issue tracking area) describe changes the domain makes to the domain and requests for changes to Core content. The requested changes are considered by the NBAC, and are reconciled with the last release, and with changes requested by other domains.
- The result of the NBAC reconciliation is published as the next NIEM release.

Some details on how the publication area works:

- Schemas in the domain partition of the publication area may extend components within NIEM releases, instead of superseding namespaces within NIEM releases.

The schemas in NIEM 2.0 contained schemas that superseded the schemas in NIEM 1.0. Schemas in the domain publication areas are not required to supersede older schemas in this manner. Instead, schemas in the publication area may extend schemas in previous NIEM distributions. This may make domain updates easier to use by IEPDs.

For example, suppose domain "JXDM" wishes to add an element "ArrestSubject" to the type "ArrestType" that occurred in "JXDM-4.0". It may publish a schema in its publication area that extends the JXDM 4.0 ArrestType. This extension may then add ArrestSubject to the ArrestType. A substitutable element may be made to provide for the new ArrestType to be used in IEPDs.

Extensions made in this manner may be easier for IEPDs to use than replacement components. When a domain publishes an update as an extension of the latest release, that update may be easier for an IEPD to use than if it was published as a replacement of the latest release.

- The specific method used in a domain update will be left up to the domain to determine. Some domains may choose to replace components, while other domains may extend components. It is expected that best practices will be determined over time. Best practices may vary from domain to domain, as some domains may prefer to use common extensions, while others may prefer that each IEPD do its own integration work.

6.3.3 Namespaces for Content of the Publication Area

Each schema that is posted to the publication area will have a namespace that meets several criteria. (1) It is unique to the posted schema. There will not be two schemas posted to the publication area that have the same namespace URI. (2) It clearly identifies that it is a namespace from the publication area. (3) It clearly identifies the domain or working group that is responsible for it. (4) It clearly identifies a version of the schema namespace.

An example is "<http://niem.gov/pub/domains/jxdm/4.2>", where the namespace is as in NIEM releases, but with "/pub" instead of "/niem".

6.3.4 Additional Requirements of the Publication Area

Each schema in the publication area has a target namespace that is distinct from that of all other schemas. There won't be two different schemas in the publication area with the same namespace, nor will there be a schema in the publication area with the same namespace as a schema in a NIEM distribution.

Each schema in the publication area is persistent. Like the schemas in the NIEM distributions, once published, the schemas will not go away. These schemas will remain available for use by IEPDs.

Schemas in the publication area will be accompanied by change logs, which give comprehensible descriptions of the changes that were made to schemas. The change log is discussed in Section 7.2, Change Log.

6.4 The Issue Tracking Area

NIEM currently conducts issue tracking using the NIEM Configuration Control Tool (NCCT). This is an issue tracking system with forums for the NTAC, NBAC, and domains. This document uses the more general terminology "issue tracking area" rather than referring directly to the NCCT, as whether the version architecture will use the NCCT in its current form is yet to be determined.

The purpose of the issue tracking area is to capture and track requirements and problems from end users, developers, domains, and other people and bodies associated with NIEM.

Although issues come from users, whether or not users will directly interact with the issue tracking area is yet to be determined. Issues may be entered by a helpdesk, in order to ensure

that duplicates are not entered, and that issues are specified fully and clearly, so that additional work is not created downstream.

The issue tracking area may include a conventional issue tracking system as well as forums/bulletin boards for discussing issues.

Each domain and working group may be given a partition of the issue tracking area, as needed. These groups will have control over their issues, and may dispatch them and resolve them as desired.

To be determined is whether a domain may also have issue tracking systems and methods outside the NIEM-provided issue tracking area. This will reduce transparency, and may impair cooperation between working groups. It may also increase error when moving issues from one body to another.

One group may delegate an issue to another group, when the alternate group is better able to address the issue.

7 Additional Artifacts

7.1 Release Manifest

The release area contains many schemas, each of which may be a part of multiple releases. The release manifest expresses (1) what are the releases in the release area, and (2) what schemas are parts of each release.

The mechanics and format of the release manifest are to be determined.

There may be a single manifest for all contents of the release area. There may also be an individual manifest for each release.

7.2 Change Log

The publication area will publish domain updates. These updates are provided as XML Schema documents, and will be accompanied by a change log.

The change log is able express a single big change that is composed of a set of small changes. For example, the renaming of an element may include changes to many types that used that element.

Without a change log, changes made between versions of a schema would not include an expression of what the changes to the schema might mean. Although the revised schema may be fully formed and well defined, it may be unclear what impact the change might have on other schemas.

For example, upon renaming an element (from an old name to a new name), the change is represented in XML schema by the original element (with the old name) not appearing in the revised schema, and a new element (with the new name) appearing in the revised schema.

A type in another domain may be impacted by this change. If the type used the original element with the old name, it would be unclear what changes should be made to the type to accommodate the renaming.

In order to describe changes to schemas with enough detail to enable reconciliation, schema definitions are accompanied by metadata that describes changes. This metadata will include a list of changes, without restating the details that occur in the schemas that are in the publication area.

The mechanics and format of the change log are to be determined. The format will be in XML, and may include XPath expressions for relating change log entries to updated schemas.

The change log XML file may describe the following about each change made by a domain:

- A description of the change, which says what was done and, perhaps, why it was done.
- A list of the components that were added as a result of the change. It may include the location of each new component, via XPath locations. It may also describe the location

(in the sequence of the original component) where an element may be inserted into a type.

- A list of the components that were deleted as a result of the change. (Note that no change actually deletes anything as all schemas are persistent, but deleted components will not exist in future schemas)
- A list of modifications, each modification giving the location of the original definition and the location of the new definition.

Users will be able to audit change logs against schema updates to look for (1) changes and updates that are inadequately described by the change log, and (2) changes in the change log that is not correlated by updated content. Such a change log may be audited when uploaded, before publication of the schemas and change log to the publication area.

Descriptions of the changes made are provided to the NBAC during the reconciliation process, and would help them to determine the contents of the next NIEM release.

7.3 Namespace URI Directive

Through the NBAC's domain reconciliation process, it may be determined that some namespaces must be adjusted to accommodate changes to other namespaces. That is, although the domain has no new updates for its namespace, the NBAC must change the domain namespace to accommodate the changes in *other* domains. In this situation, we say that the domain namespace was **bumped**.

The version architecture does not allow for changes to be made to a published namespace. A changed schema must be given a new target namespace URI. So, when a domain namespace is bumped, it must be given a new namespace URI. The **Namespace URI Directive** file will specify that URI, the target namespace URI of the new schema.

The exact mechanics of how a namespace URI directive is published is to be determined. One option is for it to be an XML file that is published in the publication area, within the domain's partition.

There are several goals for the namespace URI directive: (1) provide clear direction to those building NIEM releases regarding how to construct namespaces for domain schemas, (2) enable domains to maintain control of their namespace URIs, and the progression of those URIs from version to version. It is expected that a domain may wish to exert strong control over the version identifiers for its schemas.

The namespace URI directive allows the domain to express the structure of its target namespace URIs, and the sequence of version identifiers. The domain will be able to specify how namespace URIs are incremented when the namespace is updated or bumped.

8 Summary

Under the NIEM version architecture, domains are free to publish updates on their own timelines, without being required to wait for other bodies or participants to act. The architecture ensures that all such updates are available for use in IEPDs or other uses of NIEM schemas. The architecture ensures that NIEM releases will be made available to developers at regularly scheduled intervals, and that those releases will be easy to use, and free of structural inconsistency.

The version architecture also ensures that domain changes are made available in a concrete form that may be incorporated into the next NIEM release. It also ensures that NIEM releases are scheduled in a way that they may incorporate recent domain changes. NIEM bodies have authority to make required modifications in order to ensure consistency between the domains and Core.

This document described the NIEM version architecture at a high level. The high level description will be expanded and refined into a specification. The specification will detail the processes, activities and artifacts that are required to maintain high-quality and transparent change management for NIEM.

Appendix A: Index

- activities, 6
- appinfo, 15
- bumped, 23
- change log, 22
- coherence, 13
- coherent, 9, 11
 - definition, 13
- collaboration area, 6, 8, 18
- Core synchronization, 7, 10, 15
- cross-domain harmonization, 6, 9, 15
- data stores, 6, 18
- domain, 1
- domain reconciliation, 6, 9, 11
- domain update, 6, 8
- harmonization, 9, 15
 - definition, 5
- harmonized
 - definition, 5
- IEPD, 8
 - definition, 3
- issue, 6, 8
- issue tracking area, 6, 8, 9, 20
- major release, 7
- namespace, 20
- namespace URI directive, 23
- NBAC, 11, 12
 - definition, 1
- NCCT
 - definition, 6
- NIEM domain, 1
- NIEM major release, 10
- NIEM micro release, 9, 14
- NIEM minor release, 6, 9
- NIEM release, 2, 5
- NTAC
 - definition, 1
- partition, 19
- PMO
 - definition, 1
- publication area, 6, 9, 11, 18
- published update, 5
- release area, 6, 9, 18
- release manifest, 18, 22
- reports, 11
- scalability, 15
- semantic consistency, 15
- semantic overlap, 9
- SME
 - definition, 9
- structures, 15
- tiger team, 6, 15
 - definition, 9
- tradeoffs, 1
- URI
 - definition, 3
- version identifiers, 16
- walkthrough, 8