Integrating the Healthcare Enterprise

IHE IT Infrastructure

XDS Patient Identity Management White Paper

Date: March 4, 2011

Author: José Mussi (IHE Canada)
Nathan Domeij (IHE Canada)
Karen Wiiting (ITI Planning Committee)
Charles Parisot (ITI Tech Committee)

Email: iti@ihe.net

TABLE OF CONTENTS

1 Introduction

This white paper discusses the current state of the art and gaps with the integration of patient identity management services into a Health Information Exchange infrastructure using the IHE Cross-Enterprise Document Sharing (XDS) profile. The IHE XDS profile does not give definitive guidance regarding management of patient identifiers but does provide some general approaches and requirements of its use. This allows for a variety of models for managing patient identifiers. This white paper reviews the most common models, points out a function that is not currently supported by an IHE profile, and suggests several approaches to providing the missing function.

2 XDS and Patient Identity Management

2.1 XDS Background

The XDS profile supports the exchange of a patient's longitudinal health record across multiple enterprises. Its basic premise is that clinical information will be shared using a "clinical documents model", where a collection of clinical information pertaining to the same patient is grouped into one or more "documents" and published to a shared infrastructure. This shared infrastructure is composed of one or more document repositories and a common document registry containing metadata and pointers to all shared documents across these repositories.

Systems that publish documents are known as "document sources" while those who retrieve them are called "document consumers". The same system can be (and often will be) both a source and consumer of XDS documents. These document source and consumer actors interact with XDS Document Repository(ies) and a Document Registry Actor. All these "XDS actors" are contained within a single "XDS Affinity Domain", which establishes a set of conventions about what type of clinical documents, security constraints and other applicable policies must be used by all organizations (i.e., enterprises) that have come together to exchange documents.

2.2 XDS Affinity Domain Patient Identification

Crucial to the ability to share documents reliably is the need to uniquely and correctly identify the person (i.e., patient or client) to whom the information belongs. This is a non-trivial problem, as each clinical system that participates in the XDS Affinity Domain may (or more likely will) use different identification means for its patients. The challenge is to find a common, reliable identification scheme that can be used across the entire XDS Affinity Domain.

The XDS specifications do not attempt to resolve the identification problem, rather it assumes that the XDS Affinity Domain will have some common means to create a unique patient identifier for persons involved in the domain and allow document sources to find the appropriate patient identifier prior to publishing documents to the XDS infrastructure. This identifier is called the XDS Affinity Domain Patient Identifier (XAD-PID).

There are several approaches recommended by IHE to manage the XAD-PID. The simplest approach is when there is a shared patient identification scheme, such as a regional or national patient identifier, in place among all XDS Affinity Domain actors. , In other situations IHE recommends the use of the PIX or PDQ profiles to manage the correlation of identifiers across the XDS Affinity Domain. A PIX Manager actor or a PDQ Supplier actor provides each document source (and later document consumers as well) a match between the patient's local identifier or identity (i.e., that which is known to the Point of Service (POS) application) and the common, XAD-PID.

With this information in hand, the document source system can reliably publish documents, create submission sets and organize information into XDS folders, all assigned to the same patient identifier.

Document consumers must also query the PIX manager to determine the corresponding XAD-PID before submitting any queries to the document registry. As mentioned previously, the XAD-PID is the only identifier that is recognized by the XDS infrastructure.

2.2.1 Impact to the XDS Document Repository

The XDS Document Repository, which first receives the information from the document sources, is mostly agnostic to the XAD-PID contained in each Provide and Register Transaction. The job of the XDS Document Repository is basically to receive the documents from the source, store them within its persistence layer, calculate the hash and size of each document, and assign an unique identifier to each object and pass all this information to the document registry. It does not validate or change the patient identifier contained within the Provide and Register Transaction.

2.2.2 Impact to the XDS Document Registry

On the other hand, the XAD-PID is a key document attribute for the XDS Document Registry. Its role is to organize and group documents that belong to the same person. The registry must ensure that a proper XAD-PID is provided with each document registration. To accomplish this, and before documents are allowed to be registered for any patient, the document registry receives patient identity feeds from the XDS Affinity Domain patient identity source containing information about valid identities for the XDS Affinity Domain. With this information in hand, the document registry can ensure that:

- XAD-PID contained in the registration is valid and active for the XDS Affinity Domain
- documents within the registration belong to the same patient (i.e., the document entry metadata contains the same XAD-PID)
- all documents (document entry metadata) within the same XDS folder (folder metadata) also belong to the same patient

The key point about this approach is that the XDS Affinity Domain patient identifier is the authoritative means for identifying patients and grouping documents. Although the local patient identifier can also be provided with each document, it is not considered authoritative, is not used for grouping and cannot be specified as a query parameter.

2.3 Changes to Patient ID in Published Documents

Once a document has been published to the XDS Document Registry, there are several ways to change the assigned XAD-PID:

1. By the original document source – the original document source (or any other authorized document source) can replace the original registration with the same document with a different XAD-PID. In response, the document registry will deprecate the prior entry and add the new information as provided by the document source. This process is integral to the document lifecycle management defined by the XDS specifications and the XDS Document Registry is not aware of the reasons for the changed XAD-PID.

2. By the Patient Identity Source –the XDS specification currently allows for receipt of a patient identity merge message but does not specify how it should be handled by the Document Registry. A supplement was developed to address the processing of merge requests but that supplement, the XDS Patient Identity Merge Supplement, was based on XDS.a transactions and has recently been deprecated. It is expected that the activities following the approval of this white paper will also include a review of this matter and new specification for merge events. For this reason we give an overview of the approach as outlined in the deprecated supplement:

   In this scenario the Patient Identity Source has identified that two XDS Affinity Domain patient identifiers belong to the same person and one of the two identifiers is chosen to be subsumed (i.e., no longer in use) by the other (i.e., the survivor). A merge notification is sent to the XDS Document Registry by the Patient Identity Source Actor and from that moment on:

   - All documents that were published through an XDS Provide and Register transaction with the subsumed patient identifier are now joined with documents belonging to the surviving ID.
   - Any further submission sets communicated through an XDS Provide and Register transaction and referencing a subsumed ID will be rejected by the document registry with an "*XDSUnknownPatientId* "error.
   - All XDS Stored queries transactions referencing a subsumed patient identifier return no content.
   - All XDS Stored queries transactions referencing a surviving identifier return the entire recorded merge set of documents and return appropriate metadata.

   The XDS Patient Identity Merge Supplement does not specify changes to the internal state of the document registry. Instead it specifies required future behaviors on the part of the transactions listed above. Also, the patient identity merge notification is not propagated to the document repositories, which are not expected in the design of XDS to persist such XAD-PID. One should note that there are no IHE transactions to the XDS

Document Repository that make use of such identifiers (Document are always retrieved by Document UniqueID).

2.3.1 Limitations of the current support

As will be described in the next sections, there is no current explanation in the XDS specification on how to handle link/unlink events triggered by the XDS Affinity Domain patient identity source.

3 Cross-Enterprise Patient Identity Models

This section will focus on two of the various models used to manage the XAD-PID. It is important to understand the differences and similarities among these two models in order to understand the impact of patient identity management events to the XDS Actors. There are basically two classes or models of identity management: central matching or local matching. Both models assume that there is a single source for assigning and maintaining the XAD-PIDs. The difference between the two lies where the matching between local identifiers and XAD-PIDs occurs.

3.1 Decentralizing Matching

In this model, a central patient identity source (e.g., a PDQ Supplier) is used as the patient identity management authority (i.e., Patient Identity Source Actor) for the XDS Affinity Domain. But this actor is not responsible for determining which XAD-PID should be used in any particular transaction. It is the local system (XDS Document Source or Document Consumer) who takes on the task to properly match their local patient identifier or other patient identity information with those of the shared XDS Affinity Domain (see figure below for XDS use combined with PDQ).

Note that a similar sequence of events will occur with Document Consumer systems when querying the XDS Document Registry.

In this approach two key design elements have to be supported:

1. The mapping or matching between the locally assigned patient identity and the shared one is to be performed before documents are being published or queried and retrieved by the local system.
2. Local systems will use different approaches to determine which XAD-PID should be applied, but it usually requires queries to the PDQ Supplier to find a match from a list of candidates or validate a given public (i.e., business) identifier, such as when using health or insurance card numbers as the common identifier for the XDS Affinity Domain.

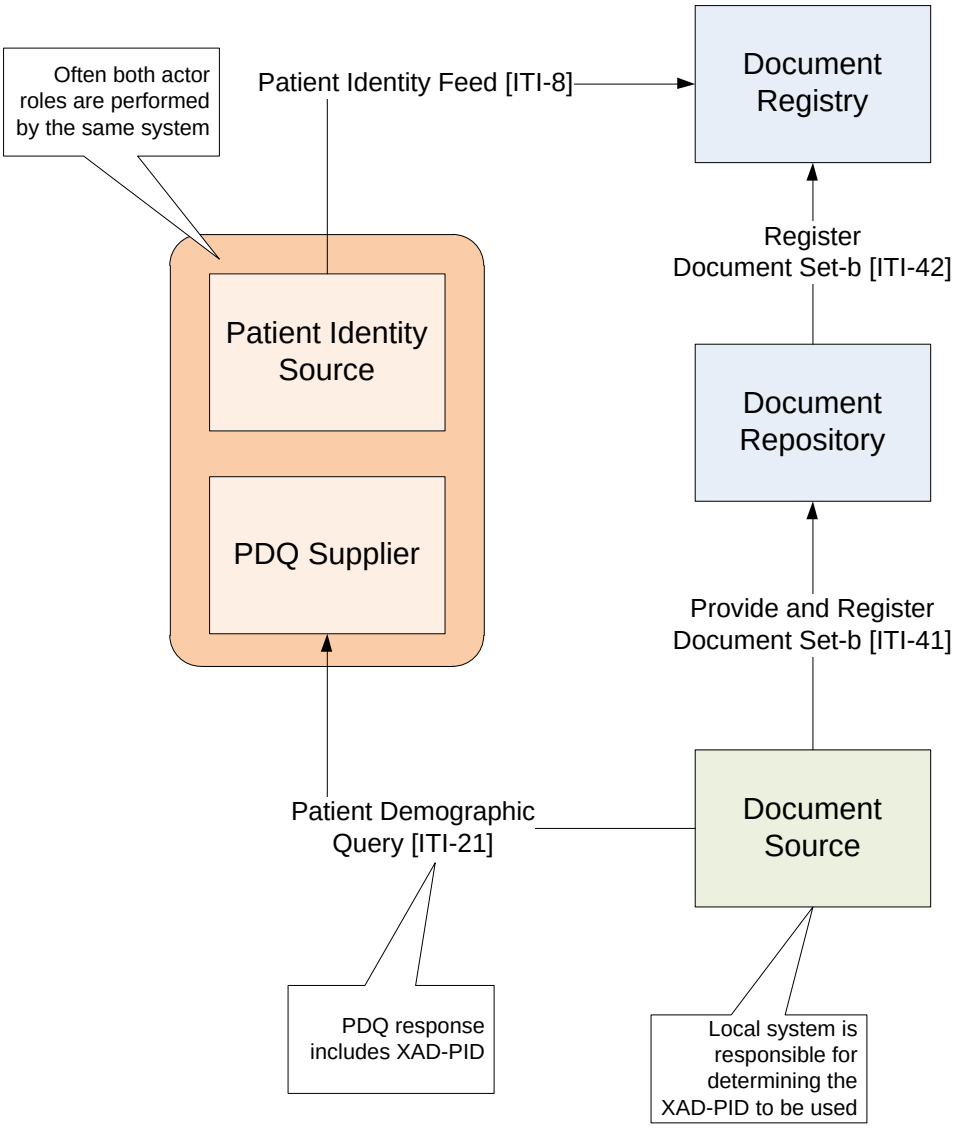

Figure 3.1-1 Decentralized XAD-PID Matching using PDQ

The XDS Affinity Domain must define the process (automated, and /or manual) for the administration of these common identifiers. Such identity management processes often piggyback on other business processes, such as those put in place to obtain patient consent when opt-in is used.

Consequently, this model requires that most changes to the XAD-PID associated to any particular document must be submitted by the original source system (or another authorized source). The only exception is that XAD-PID merge events could be managed and triggered by the central patient identity source and notified directly to the XDS Document Registry through use of the mechanisms described in the deprecated XDS Patient Identity Merge Supplement.

3.2 Centralized Matching

In this other model, the source of XAD-PIDs is assumed by a central service, which also has the responsibility of matching local identifiers with the XAD-PID. This requires the use of a PIX Manager (or similar service) to create and manage the linkage sets between all known identifiers (i.e., all local patient identifiers + XAD-PIDs) as shown below:

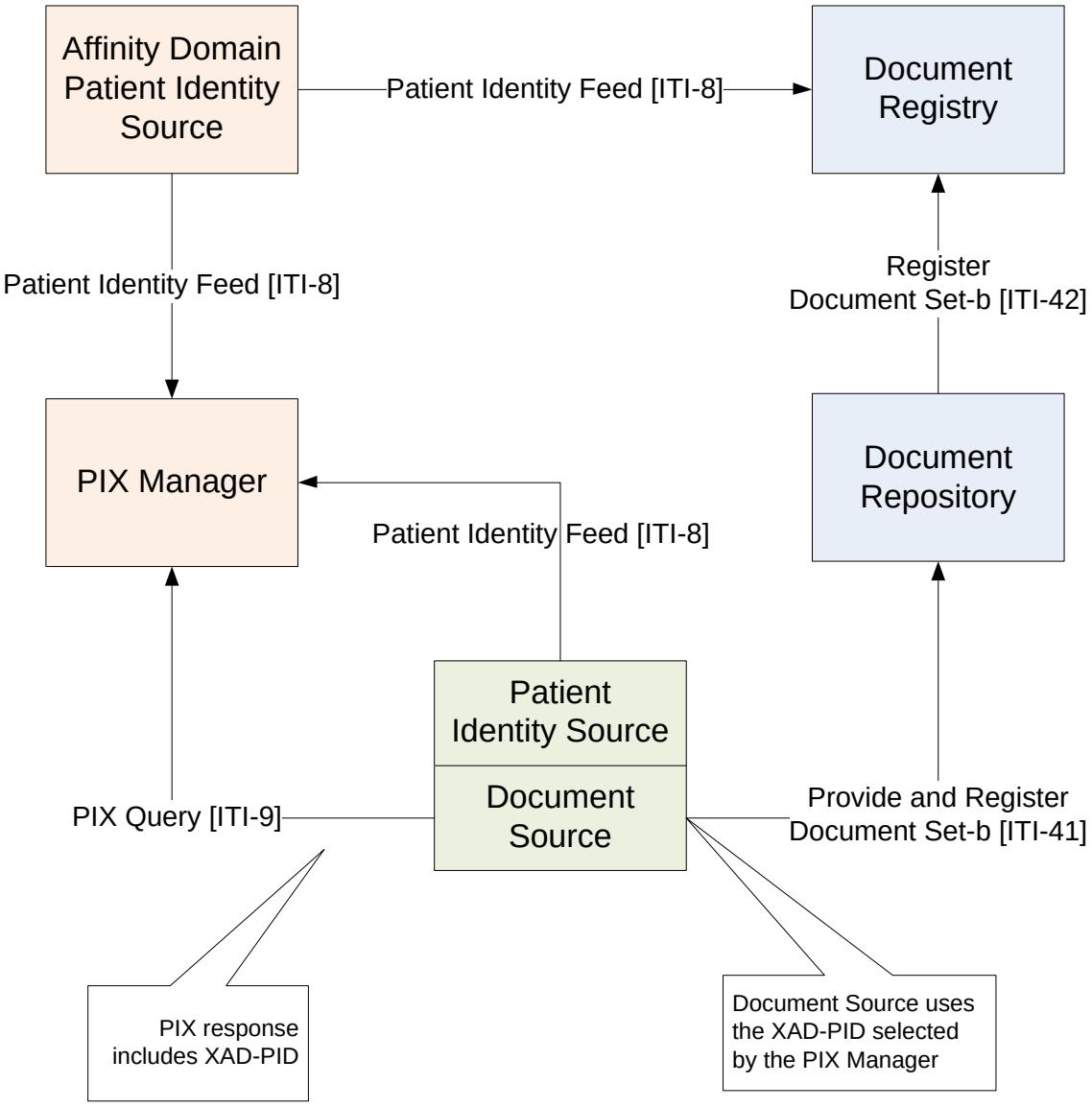

Figure 3.2-1 Centralized XAD-PID Matching

The following conditions must be met for this model to work:

- The Document Source system must also be a PIX Patient Identity Source to the PIX Manager
- The XDS Affinity Domain will establish the business rules for assigning and managing the valid set of XAD-PIDs (could be provided by an external source or by the PIX Manager, if able to dynamically create XAD-PIDs)
- The PIX Manager will implement matching rules and algorithms (deterministic, probabilistic or both) that meets the minimum data quality standards established for the XDS Affinity Domain. Manual processes may be required to correct any questionable matches.

With the conditions shown above in place, the Document Source (or Document Consumer) needs only query the PIX Manager using its own local identifier and retrieve the matching XAD-PID. The local system will trust that the correct match has occurred and use that common identifier in all transactions with the XDS infrastructure.

The fact that the local system has no knowledge of how the XAD-PID was determined or that the relationship sets that define the matches between local identifiers can change means that a new level of communication is required between the PIX Manager and the XDS Document Registry.

As seen previously, XAD-PID merge events can still occur and will be handled as described previously. However, in addition, we now must consider that link/unlink events can also occur within the PIX Manager and these are currently not addressed by the XDS specifications. This is one of the issues to be discussed in the next section.

4 Patient Identity Management Gaps in XDS Specifications

The description of the XAD-PID management schemes in the previous section highlights that there are gaps in the XDS specifications that need to be addressed. This section will describe these gaps and offer options for their resolution

4.1 XAD Level Link/Unlink Events

A recent white paper from Canada[1] describes a main consequence of the Client Registry to XDS integration, which is, the need to support changes to the XAD-PID linkage sets. If the XAD-PID assigned to a local ID never changed, than there would be no impact to the XDS Actors, but since this is not the case, how can the XDS infrastructure deal with the dynamic nature of the XAD-PID linkage sets?

The white paper provides a very good description of the use cases involved in EHR patient identification as implemented in Alberta, which is based on the centralized matching model described in Section 3.2. It also provides three suggestions that resolve the problem of how to notify link/unlink events to the document registry. In all cases, the solution requires that the document source system provide in the document metadata the local patient identifier that was used to execute the client resolution (i.e., the PIX query). This is a significant change from the base XDS specification where the local patient identifier (currently stored in *SourcePatientID* metadata field) does not have any important role in the behaviour of the XDS document registry. It is merely another attribute associated with the document.

To illustrate the scenario in discussion, let's assume that a patient presents to a service location in a given XDS Affinity Domain for the first time and that a set of documents from that encounter are published to the XDS infrastructure:

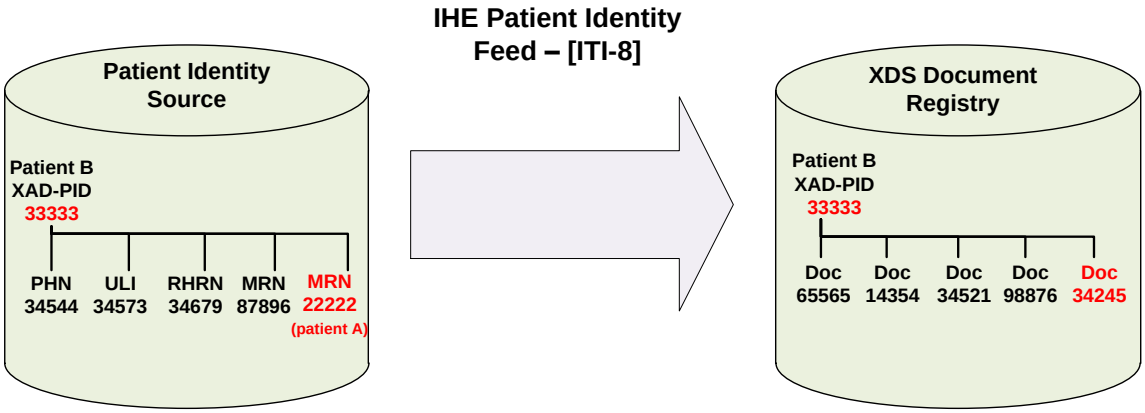

Figure 4.1-1 Initial State of New Document

[1] "Link/Unlink Analysis: pHIE – XDSi", Alberta NetCare Health Information Exchange, Feb/2010

The local patient ID (MRN 22222) is mapped (i.e., linked) by the PIX manager to an existing XAD-PID (XAD-PID 33333). One or more documents are published to XDS using that common identifier.

However, at some later time, it is discovered that Patient A should not have been linked to that XAD-PID in the first place and that in fact, it should have been linked to another identifier as shown below:

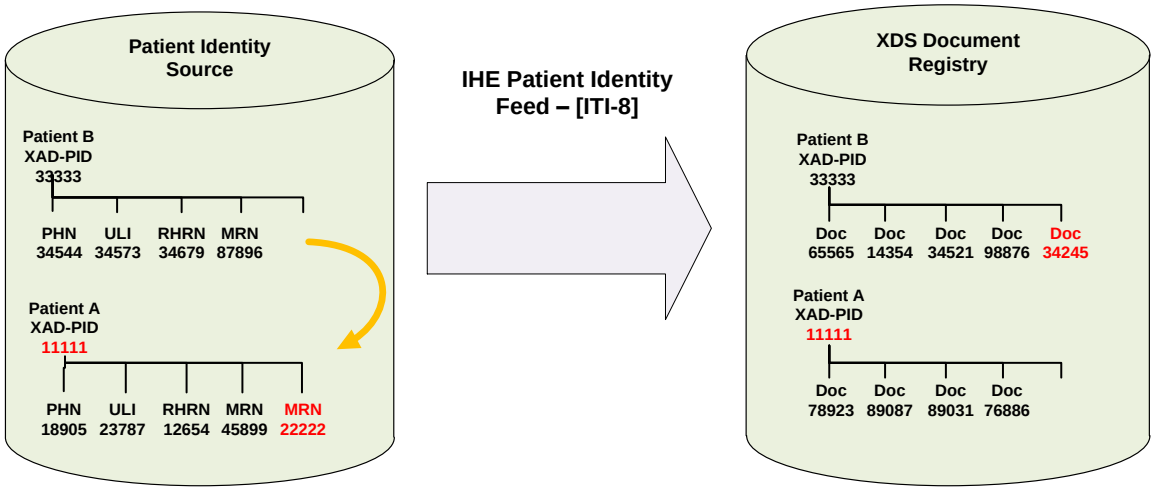

Figure 4.1-2 Link/Unlink MRN

In this case, we see that the correct XAD-PID is 11111 and the change occurs within the XDS Affinity Domain patient ID source. However, the previously published document (DOC 34245) needs to be corrected and reflect this change. Given that the original document source system may not be aware of the link/unlink event, it cannot be expected to deprecate and re-publish the document itself.

4.2 Solution Analysis

The essence of the link/unlink issue is ensuring that the XDS Document Registry is corrected to reflect any and all changes to the XAD-PID linkage sets. The finding the best solution will involve answering two questions:

1. How are linkage set changes inside the PIX manager propagated?
2. How will document entries in the XDS Document Registry be fixed?

The following sections address these two questions and offer some alternatives on how they can be implemented.

4.2.1 Notification of New XAD-PID Link

One approach on how to handle notifications from link/unlink events focuses on the local identifier's relationship to a XAD-PID. As described previously, even before any document is

published to the XDS Document Repository, the source system will have informed the PIX manager about its local identifier (MRN 22222) and the PIX manager will have determined, through its own internal matching algorithms and possibly with human assistance, which XAD-PID the local identifier needed to be linked to. This common identifier (XAD-PID 33333) is used when the patient documents' metadata are sent to the XDS Document Registry.

As the error is discovered and the PIX manager changes the local identifier to a new linkage set (XAD-PID 11111) it would send out a notification that basically would be saying:

"The correct XAD-PID for MRN 22222 is 11111"

The most appropriate message seems to be ADT^A43 (Move Patient Information) as it was designed specifically for this purpose. For the example given, this notification would look like:

```
MSH|^~\&|EMPI||EHRI|CH|20100112113930||ADT^A43|20100126000022537083|D|2.4
EVN|A43|20100126113925
PID|11111^^^^XADPID^EMPI|22222^^^^MRN^HOSP_2|||||||||||||||||||||||||||N
MRG|22222^^^^MRN^HOSP_2|||33333^^^^XADPID^EMPI
```

Figure 4.2.1-1 - Sample ADT^A43

As is usually the case with HL7 v2, there are other message types that could achieve the same goal. For example, ADT^A47 (Change Patient Identification List) could be used instead. A more complex approach would be to use two separate messages, ADT^A24 (Link Patient Information) and ADT^37 (Unlink Patient Information) to convey the event. Although all these options could be use, ADT^A43 seems to be the simpler solution.

With this notification in hand, it is fairly straight forward how to fix the registry:

"For every document where SourcePatientID="MRN 22222", assign *PatientID* to 11111"

This change would ideally be done directly by the XDS Document Registry, as it would have the ability to perform the database update efficiently and reliably. An alternative would be to have another system determine which documents need to be changed and issue a series of metadata update request to the XDS registry. This will be discussed in a later section.

4.2.2 Notification of Linkage Set Updates

Another approach would be to have the PIX Manager send out notifications with the entire linkage set anytime one changes. In this example, this would mean two notifications would be sent out: one for XAD-PID 33333 and another for 11111. The idea here is that receivers of this notification (e.g. the XDS Document Registry) would maintain their own internal copy of the linkage sets for every valid XAD-PID and would use these sets to maintain the correct relationship between documents.

This notification can be sent using ADT^A31 (Update Person Information) message. This approach requires that a first message be sent to unlink the local ID from the current XAD-PID and a second message to re-link the same ID to a new XAD-PID.

For the example given in Section 4.1, the first ADT^A31 message would contain the new information for Patient A (i.e. patient that has "added" a new local identifier to its list):

```
MSH|^~\&|EMPI||EHRI|CH|20100112113930||ADT^A31|20100126000022537083|D|2.4
EVN|A31|20100126113925
PID|11111^^^^XADPID^EMPI|18905^^^^PHN^ABH~23787^^^^ULI^ABH~12654^^^^RHRN^
CHA~45899^^^^MRN^HOSP_1~22222^^^^MRN^HOSP_2||PATIENT_A|||||||||||||||||
||||||N
PV1|N
```

Figure 4.2.2-1 - Sample ADT^A31 – Patient A

Note that the message includes the XAD-PID for Patient A and the five related local identifiers. The XDS document registry upon receipt of this message would have to transverse the document tree for each local identifier in the PID list and ensure it is assigned the correct XAD-PID (11111).

The second message would inform the registry about the new set of identifiers for Patient B:

```
MSH|^~\&|EMPI||EHRI|CH|20100112113930||ADT^A31|20100126000022537083|D|2.4
EVN|A31|20100126113925
PID|33333^^^^XADPID^EMPI|34544^^^^PHN^ABH~34573^^^^ULI^ABH~34679^^^^RHRN^
CHA~87896^^^^MRN^HOSP_1||PATIENT_B|||||||||||||||||||||N
PV1|N
```

Figure 4.2.2-2 - Sample ADT^A31 – Patient B

Using two asynchronous messages to correct a single XAD-PID assignment is a concern since it could leave the registry in an incorrect state if one fails. It would also require the document registry to perform a differential comparison with the information it previously had about that patient and identify any changes to the local patient IDs assigned to the corresponding XAD-PID.

Fixing the XDS registry would be somewhat more complex, since either the XDS registry or an external actor would have to figure out what has changed (the messages are not explicit, rather just reflect the current state of the linkage set) and then performed the necessary database updates.

Overall, the approach described in Section 4.2.1 appears to be a better solution overall.

4.2.3 Use of XDS Metadata Update

A recent option, part of the 2010 work items, that was considered is the use of the XDS Metadata Update[2] which has provisions for updating the *PatientID* information in folder and document entry objects. The use of the transaction Update Document Set [ITI-57] to resolve link/unlink

[2] IHE IT Infrastructure Technical Framework Supplement, XDS Metadata Update – Trial Implementation (August 10, 2010)

events, although feasible, would result in a much more complex interaction model between the PIX Manager and the XDS Document Registry. Since Update Document Set must address each individual object impacted by the new association between local identifier and XAD-PID, it would require that the PIX Manager perform a Registry Stored Query [ITI-18] (as is the case with the Document Administrator) and create a (possibly) large submission set with all objects (folders and document entries) that would need to be changed. This type of capability is non-trivial and likely well beyond the current capabilities of systems that implement the PIX Manager role. The use of existing HL7 v2 ADT messages, such as ADT^A43, seems like a much simpler approach, although it does transfer the complexity to the XDS Document Registry actor.

However, as discussed in the next section, the impact to the XDS Document Registry of patient ID changes is well documented in the XDS Metadata Update Supplement and should be considered when finalizing the decision on the link/unlink support.

4.3 Additional Considerations

However, the white paper does not address some key impacts to the XDS implementation that will occur regardless of which messaging approach is adopted. Although it does note the need for document sources to include the local identifier with each submission set, there are three other issues that need further clarification:

4.3.1 Impact to Submission Sets and Folders

Link/Unlink events can cause inconsistencies in submission sets or folders, where documents from two (now) different patients are grouped together. The issue here is that when folders and submission sets are first created, they are validated to ensure that all documents they contain belong to the same patient (i.e., they have the same XDS Affinity Domain patient ID). After a link/unlink event, it is possible that this condition is no longer valid, as one or more documents that used to belong to a particular XAD-PID is now changed to a different identifier.

One solution would be to include a new (optional) restriction where documents within submission sets and folders must have the same local patient ID. This would ensure that regardless of what happens in a link/unlink scenario, that integrity of the objects in respect to patient identity is preserved. However, this may be too restrictive in regards to folders, where some implementations are using them to combine documents from various sources.

A more pragmatic option would be to maintain the current rules (i.e., documents from the same patient, regardless of source) but add a new behaviour to the registry. If, in consequence of a link/unlink event, one or more documents within a given folder no longer have the same XDS Affinity Domain patient ID than these must be logically removed from the folder. In addition, the XDS document registry must generate an exception event (possibly through a report) to notify that such action has taken place and allow for subsequent corrective measures by the document source.

This is similar to what is recommended in XDS MU (page 17):

"An update that changes the Patient ID attribute of a Folder or DocumentEntry is more complicated. The rules for consistency of Patient ID between Folder and member DocumentEntries require that if a DocumentEntry gets a new Patient ID then it must be removed from the Folder (assuming the Folder does not change). The association propagation rules implemented by the Document Registry actor do not handle this type of update. The Document Administrator must calculate and submit all of the detail changes required."

The key difference is that unlike what is recommended above, the burden of removing the affected documents from folders when patient ID surface would fall upon the Document Registry and not the system that triggered the link/unlink event (e.g., PIX Manager)

This change should not create an issue with submission sets, since documents from the same document source would likely have the same local ID. If this is not the case, than the XDS Document Registry will need to handle the situation.

However, the same may not be true in regards to XDS folders. These can very likely hold documents with different local IDs and a link/unlink event could result in a folder containing entries that now belong to two different XAD-PID references. This would represent an invalid state for the folder that will need to be resolved. Since folders are not widely used, it is difficult to assess if this restriction would create implementation barriers.

4.3.2 Impact to Document Repositories

Link/Unlink event notifications are not propagated to document repositories. The issue here is that document repositories may also contain a copy of the metadata published with each submission set. If this is the case, than existing record would no longer be correct after the link/unlink event occurs.

This situation is similar to the patient demographic information kept in the metadata record. It is not guaranteed to be correct over the lifetime of a document, and only reflects the original state of the record. If this is acceptable, and document consumers are aware of possible discrepancies, than no further changes are required. Otherwise, the same message used by the document registry to handle link/unlink events will have to be passed on to all document repositories in a XDS Affinity Domain.

Note: Many of the current document repository implementations do not persist the document metadata, in which case this issue is not applicable.

4.3.3 Impact of Local Merge Events

Merge/Unmerge of local patient Identifiers need to be propagated.

Here is the scenario: two local identifiers (Lid-A and Lid-B) have been in use for some time but have been determined to belong to the same patient (i.e., source system duplicates). A merge event occurs in the local system and is notified to the PIX Manager, where Lid-A is subsumed by Lid-B. Within the PIX Manager, Lid-B is the only surviving ID but the XDS infrastructure still has one or more documents attached to Lid-A. Since (presumably) Lid-A and Lid-B were linked

to the same XAD-PID (let's say XAD-PID-123), this situation is not a problem as all queries to the registry would use that same XAD-PID number. However, if in the (not very likely) situation where Lid-B is now determined to be linked to the wrong XAD-PID (i.e., it should be linked to XAD-PID-987), we would have a problem. The link/unlink solutions here described would see a notification that all documents belonging to Lid-B should be now assigned to XAD-PID-987. Nothing would be said about Lid-A since it has been for all purposes subsumed. At this point, the document registry would be left in an incorrect state, where Lid-A documents remain attached to XAD-PID-123 while Lid-B documents are moved to XAD-PID-987.

There are different ways to address this problem; most require the propagation of local merge/unmerge events to the XDS infrastructure. This will add a new level of requirements to the XDS integration model and may only be required in very rare instances. Alternatively, one may define a solution where the client identity source needs to include all merged identifiers (i.e., all subsumed local Ids) when creating a notification of a link/unlink event.

5 Recommendations

This white paper has discussed the various implementation models that are used for patient identification management and how they impact the XDS Actors. In particular, it has addressed the scenario where changes to the link between a local identifier and the XDS Affinity Domain patient identifier (XAD-PID) need to be propagated to the XDS Document Registry. This has been identified as an issue in several XDS implementations in Canada.

Considering the various options presented, the option that seems to address the problem with the least impact to the existing XDS Actors is Alternative 3, that recommends to introduce a new optional message (ADT^43) either in an existing Transaction (e.g., Patient Identity Feed [ITI-8] transaction) or as a new Transaction.

This message would inform that a local ID (LID) that was previously linked to XAD-PID(a) is now linked to a different XAD-PID(b). The receipt of this message by the document registry would result in a global change within the registry of all objects that contain the {XAD-PID(a), LID} should be changed to{XAD-PID(b), LID}.

The introduction on new requirements on the use of the local patient identifier could introduce significant changes to existing XDS products and should be analyzed with care. This is especially true when considering adding the support for merge/unmerge events. Since the supplement that dealt with these events has been deprecated, it is necessary to revisit the issue in alignment with whatever solution may be chosen to handle the link/unlink needs.

In addition, this white paper further recommends additional discussion in respect to how to handle the impact of link/unlinks on folders and submission sets previously published to the XDS infrastructure.

Appendix A: PIX/PDQ Integration Models with XDS

A.1 Canadian Interoperable EHR Blueprint

In the Canadian interoperable EHR architecture, the responsibility for managing and cross referencing client (i.e., patient) identities often falls with a central EMPI services known as the Client Registry (CR). The CR collects registration information from various patient identity domains in its jurisdiction and groups these records together through a combination of automatic algorithms and manual linking events. These sets of linked Ids are given a unique identifier, known as the Enterprise Client Identifier (ECID), which is analogous to the IHE XDS Affinity Domain patient identifier (XAD-PID).

The ECID can either be a new ID created by the CR itself (i.e., shadow ECID) or can be assigned (through jurisdictional policy) by one of the patient identity domain (i.e., provincial health card number). Regardless of the model, the CR is the only authoritative source of ECIDs for the jurisdictional EHR infostructure. In most cases, when documents are published to the XDS the actual resolution of local ID to ECID is not performed by the document source, but rather by other common services (i.e., the HIAL) in the XDS Affinity Domain.

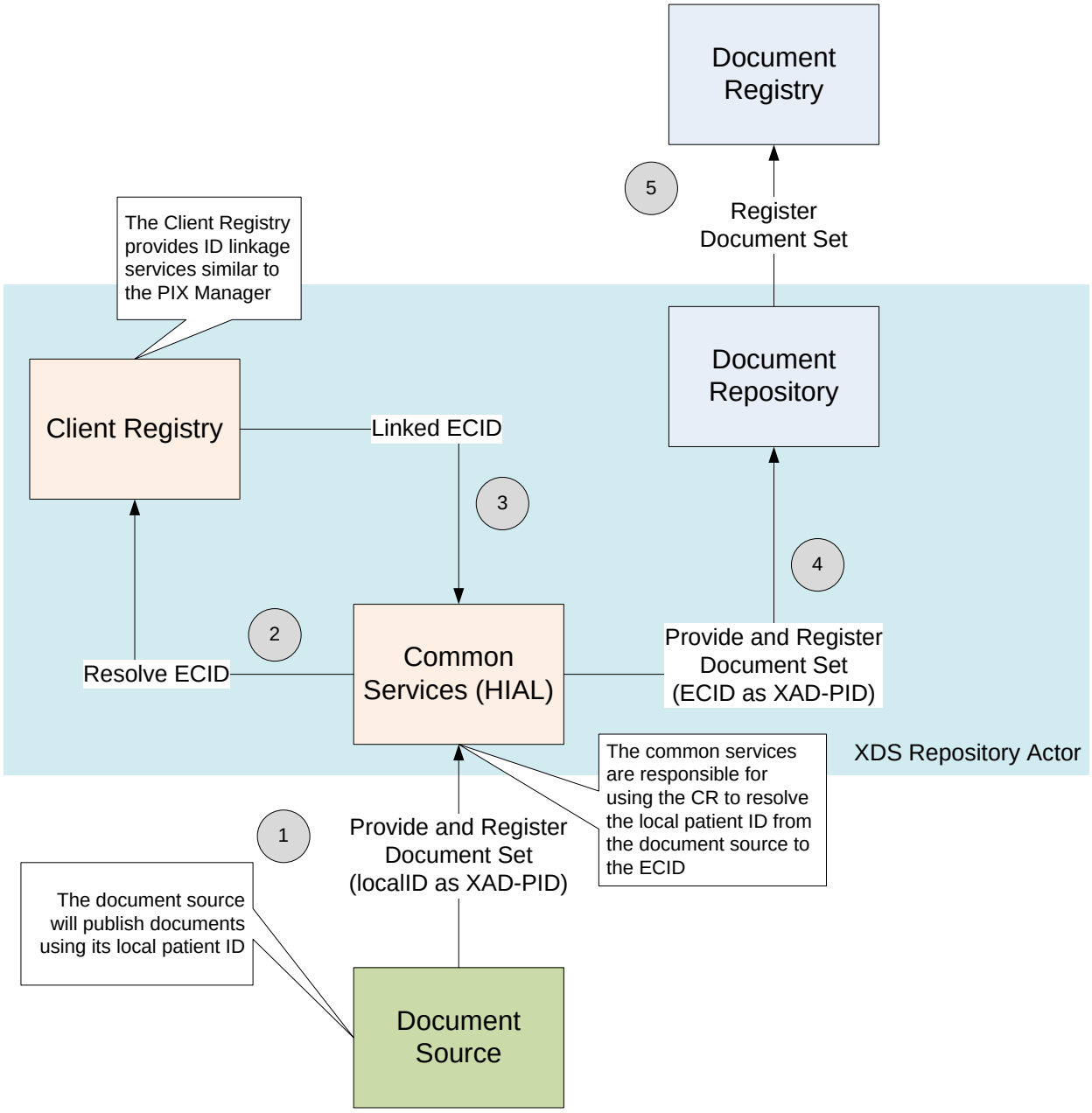

Figure A.1-1 Centralized XAD-PID Matching

For the most part, the jurisdictional CR role is very close to the services provided by a combined PIX/PDQ manager. The most significant difference is that the relationship local ID → ECID is not guaranteed to be permanent. In fact, as described in the Alberta Link/Unlink white paper, it can very often change because a new local patient ID will be given an initial ECID when it is first published to the CR. It may or may not remain the same, depending on the result of the matching algorithm and possibly manual linking events. If a match is found, the local ID is

moved from the original ECID to another, pre-existing ECID. In this case, the original ECID is deprecated and no longer in use[3]. There are also other scenarios where, for different reasons, the original match of a local ID may be wrong and a new ECID assignment is required.

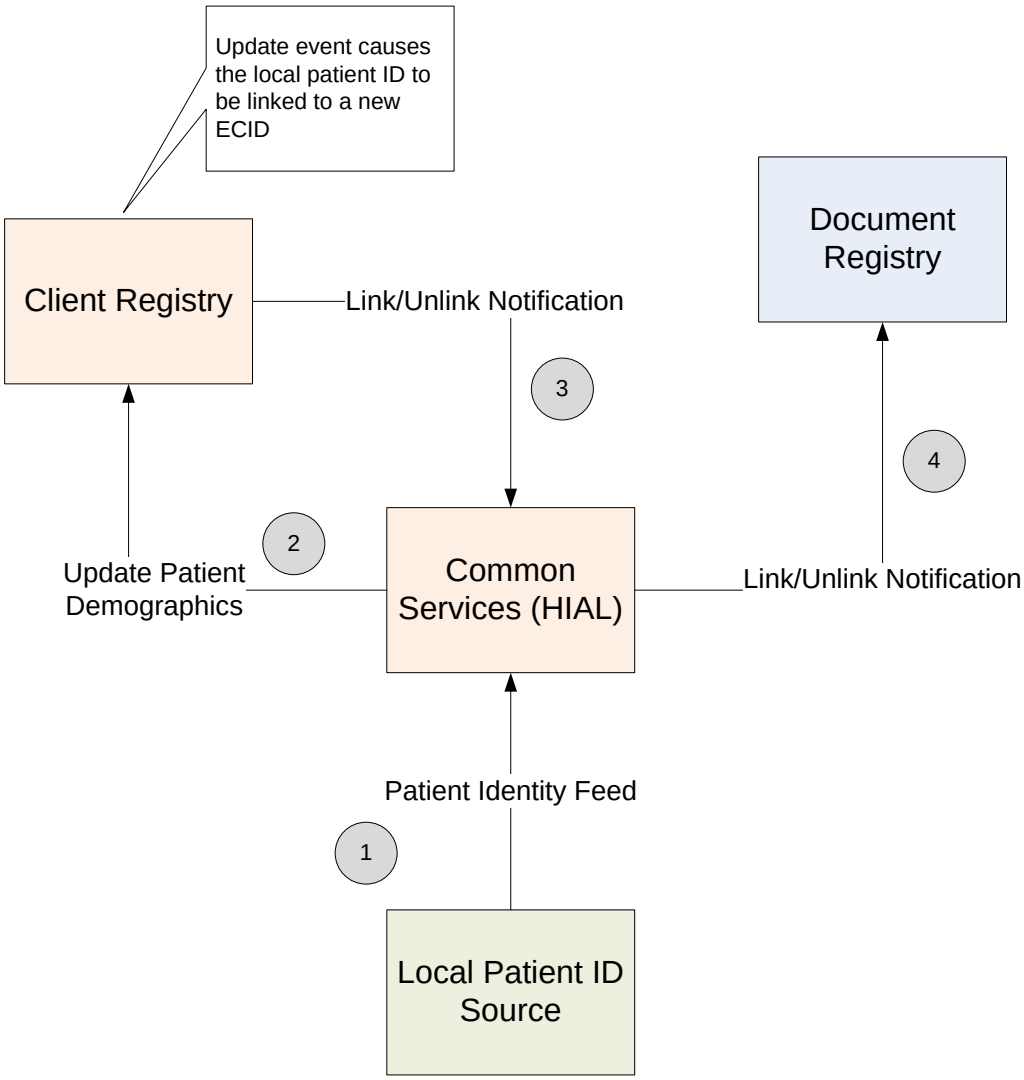

Figure A.1-2 Link/Unlink Events

In all these cases, the impact to the XDS services occurs if there is one or more published documents belonging to the local patient ID in question. Since this patient identifier is now linked (i.e., assigned) to a different ECID, the XDS services needs to be informed about the event and its information corrected.

[3] This is similar to a patient identity merge described previously since the original ECID was a singleton, that is, it only contained one linked local ID.

When these change events occur, the CR will trigger notification messages to any system that relies on the ECID. Typically, these notifications will be forwarded to another jurisdictional service, called the Health Information Access Layer (HIAL), responsible for routing such messages to all appropriate "listeners". The XDS document registry would need to be one of the recipients of the link/unlink notifications.

These examples highlight the key nature of the issue: in the CR model, the authoritative patient identifier for clinical data published to the EHR (and this would include XDS documents) is the local patient ID from the POS source system; the ECID can be seen as just an attribute, externally assigned by the Client Registry that can change!

Note: There are other different EHR client identification models applied in Canada:

1. In some cases (i.e., the province of Quebec), the ECID is a public identifier that must be resolved by each EHR source or consumer system, much as is described in the XDS interaction model. This approach does not require the support of link/unlink and can easily adopt XDS solutions as specified.

2. In other places (i.e., province of Manitoba), the CR manages the linked list as described but the ECID is not used by EHR repositories. Records are created in the EHR just with the local identifiers and linked together, via the CR linked set, dynamically at query time. Link/unlink events are handled exclusively by the CR and repositories need not be notified. However, this model does not fit well with the XDS specifications which require a single patient identification domain. It is likely that the ECID would be used for XDS documents, bringing us back to the model in discussion.

A.2 None Unique XAD-PID Model

Finally, to complete this whitepaper we thought it would be useful to describe a model that may be adopted in some settings. Imagine that there is no XAD-PID available for a particular XDS Affinity Domain. Is it still possible to use XDS and what would be the impact to the solution?

As has been highlighted in this document, the XDS assumption that there would be a XAD-PID available has enabled its adoption in many different scenarios, notwithstanding some pending issues as discussed. However, this may not be always the case. There are a couple of scenarios that fit this model:

1. *Single XDS Affinity Domain, patient ID matching provided by a PIX Manager (or equivalent), but no single, common XAD-PID is created for each patient:*

   In this scenario, document source systems provide the PIX Manager with their local patient IDs but no XAD-PID is available. These systems will publish documents to the XDS Document Repository placing their local ID in the *patientID* metadata field. The XDS Document Registry would know about all document entries, but a complete view of the patient's information is not readily available through the use of a single identifier.

2. *Multiple XDS Affinity Domains, each managing their own XAD-PID, and all sharing a single XDS Document Registry:*

   Here the issue is not the availability of a XAD-PID, but rather the fact that there are possibly many XAD-PIDs created for a single person. Although the setting is different from the first scenario, it results in the same problem: documents for the same patient cannot be directly linked by the XDS registry through a single identifier, thus compromising its ability to provide a valid longitudinal view of the data.

In both scenarios there needs to be a way for the document consumers to obtain the longitudinal view they seek. This is only possible if there is some way to link the various patient IDs contained in the XDS Document Registry. In scenario 1, this would likely be the PIX Manager for the domain, where linkage sets would exist for all the patient IDs from the document source systems. For scenario 2, this would need to be a PIX Manager that can create and manage these linkage sets for the XAD-PID domains involved in the exchange.

Having the appropriate PIX Manager in place, the next problem is how to make use of it to resolve the business scenarios addressed by XDS.

The first approach is to have the XDS document consumer query the PIX Manager first, obtain the set of corresponding patient IDs for the person in question and issue separate queries to the XDS registry for each ID. Although this does not require any changes to the XDS Document Registry, it will likely impose a performance penalty on logitudinal views as each query/response will represent a separate interaction between consumer and registry.

The other approach would be to have the XDS registry, upon receipt of a longitudinal query, reach out to the PIX Manager, obtain the linkage set and peform the complete query itself.

570 This last approach would require a change to the XDS specification.

