Entry	Type	Description	How Implemented
Remove LDAP Implementation	Feature Request	The LDAP implementation of the LexGrid model is no longer being used. The LDAP specific elements in the LexGrid model should be removed, as they add a degree of complexity and confusion that is no longer justified.	Removed LDAP Package
			Removed NumericOID type
			Removed all LDAP annotation on the individual entities
			Removed the "dc" type on aggregation columns
			Removed the constraint that all nodes had to have a single identifier that was unique in the context of the parent
Model Clarification	Enhancement Request	There are several issues that have made the model difficult to explain, implement, and use. These issues include: 1) The inconsistent use of names - some core data types begin with "ts" and others don't. 2) Naming confusion - "URN" is used in several places where the data type should be a URI, labels say "id" when they mean "code", etc. 3) Inconsistent typing - localId is used as a type throughout much of the model instead of specifying the particular domain (e.g. source, language, namespace name, etc) 4) Inconsistent use of the "any" type - it has a misleading label in the builtins section (tsCaseSensitiveDirectoryString), and then isn't used consistently through the model. 5) While we aren't yet resorting to relaxNG, we would like the XML validation to catch more obvious errors. Weak typing prevents some of this validation, but setting minLength to "1" on fields that are expected to have content will reduce the number of XML documents that validate but don't load correctly	
All text type fields need data types	Enhancement Request	We need the ability to add a data type (format) to the target of associations as well as the target of any property. Right now, it only applies to property	Changed builtins name to "tsAnType". Added an optional dataType attribute to the field. Changed the types of "text" and "entityDescription to "tsAnyType". Changed property and associationData to have a reference to an entity of type "text" rather than being a kind of "text"
Flexible Entity types	Enhancement Request	The 2008/01 model supports a fixed set of entity types - concept, relation and instance. While this aligns with OWL/DL, it doesn't account for (a) terminologies that haven't differentiated classes from individuals, (b) OWL Full, where an entity can be both a class *and* an individual and (c) other type systems, such as that supported by KTMI	Made "entity" a first class class, that included all properties and characteristics. Added an optional, repeating entityType field which allowed the entity to be classified, added supportedEntityType in the mappings to allow types to be customized, removed the fixed enumeration of types and added constraints that define "concept", "instance" and "relation" by the entityType field

Incremental Revisions	Enhancement Request	LexGrid updates need to be communicated as sets of changes rather than complete sets of contents. The history mechanism needs to be extended so that a collection of changes can be communicated that will allow an existing system to be updated incrementally	Changed the definition of versionable, added entityState and revision model elements and changed the definition of systemRelease to carry a collection of revisions. Note: This change is closely coupled with the Refined History Model change request.
Refined History Model	Enhancement Request	LexGrid needs the ability to version and activate changes on the property, entity, association instance, pick list, pick list entry level in addition to the concept level. Each of these entities need to support a status, state (active or inactive) the date/time when the change is to become active, the data/time when it is to become inactive, and additional metadata about who, how and why the change should be applied.	Revised versionable to support the activation state, status, effective and expiration dates
			Provided an optional link from versionable to an entryState record that carried the type of change and the revision that this change was included in. Created a model of state changes (changeType), and created machanisms for traversing revisions to determine what changed, when, where, etc.
Namespaces Aren't CodingSchemes	Bug Fix	The namespace used to qualify the URI of a coded entity isn't necessarily the namespace of the coding scheme making an assertion about the entity. These two elements are convoluted in the current LexGrid model.	Added a new localId type, "NamespaceName" and a new mapping, "supportedNamespace". Changed the codingSchemeId attribute of "entity" to entityCodeNamespace, which references a supportedNamespace. If entityCodeNamespace is present, it references a supportedNamespace, which, in turn, has an attribute, "equivalentCodingScheme", which has the local name of the codingScheme that corresponds to the entityCodeNamespace
Revise Value Domains / Pick Lists	Enhancement Request	The value domain model needs to be extended to support the definitions represented in the IHC model. In addition, the model needs to support (a) HL7's value domain *definition* model and the sort of definitions that can be created through the DTS editor. The model of pick lists need to be extended accordingly to meet multiple GE/IHC requirements	Completely replaced the ValueDomains package to carry the new model.
Dual Type Properties	Enhancement Request	RDF based loaders transform triples into a combination of (a) first class attributes (e.g. entityDescription, copyRight, presentation, definition, ...) (b) generic lexical properties or (c) relations. Properties and relations preserve the original RDF type, but the first class attributes lose the information about where the resource was derived from. In addition, there is no way to assign status and provenance information to the first class attributes	Model philosophy was changed to have first class attributes represented in two forms: as an attribute and as a property that is identified as being an attribute. To do this, we added new attributes to property and propertyQualifier that, if non-blank, stated the first class entity that this property (or propertyQualifier) represented. As an example, the copyRight entity would also be represented as a property with a propertyType of "copyRight"

		(see: Refined History Model)	
Backwards Compatibility	Feature Request	The LexBIG API history API goes directly against the NCI Cumulative History content and renders its output in terms of codingSchemeVersion, version and entity version. These elements need to be preserved in the new LexGrid model as deprecated elements that exist for backwards compatibility with the LexBIG API.	Preserved Version, Version Reference, representsVersion, entityVersion and codingSchemeVersion, but marked them as "deprecated"
Common Mappings Type	Feature Request	The 2008/01 model has two different "mappings" entities, one for codingScheme and one for valueDomain. Each has a different collection of supported<type>, and the order of the entries are confusing and arbitrary. With the addition of another "mappings" entry for pick list, we suggest that the three mappings be consolidated into one type, and the contents be alphabetized. This will make code management and authoring easier. (second entry) It should be possible to enter a codingScheme, valueDomain or pickList without having *any* mappings entries and have the loader fill out the information of all of the localId's and, where possible, the URI's that they map to. This should not be the function of an editor or type transformation package	Created a new "mappings" entry in Naming package, removed the existing entries from codingScheme and valueDomain and pointed them at the new entry. Alphabetized the entries and made them all optional.
Agent Role on supportedSource	Bug Fix	supportedSource has an agentRole field, but role is a property of the association of the source with the entity (e.g. the source may be author on one field, editor on another).	Removed agentRole from supportedSource
Assertions can be inferred, entities cannot	Bug Fix	isInferred is listed as a property of a concept. DL can infer additional axioms *about* a concept, but they cannot infer the existence of a concept that isn't already specified.	Moved isInferred from concept to associatiableElement
PropertyLink.propertyLink is confusing	Bug Fix	the link attribute in the propertyLink element was renamed to "propertyLink". This is confusing	(not fixed yet)
isTranslationAssociation is not a property of the association, but how it is used.	Bug Fix	Association was made a first class entity in the 2008/01 model, meaning that all of the characteristics had to be properties of the association itself, not how it used in a particular relations collection.	Removed isTranslationAssociation from the model. No alternative available at the moment

		isTranslationAssociation is a property of use, yet is listed as a property of the association itself	
targetCodingScheme is not a property of an association, but how it is used.	Bug Fix	Association was made a first class entity in the 2008/01 model, meaning that all of the characteristics had to be properties of the association itself, not how it used in a particular relations collection. targetCodingScheme is a function of how it is used, not the association itself	Removed targetCodingScheme from the model, meaning that mappings across coding schemes will always have to provide the namespace id for the target element.
associationName is a localId, not an entityCode	Bug Fix	Association was made a first class entity in the 2008/01 model, and associationName was removed anticipating that the entityCode and associationName would always be the same thing. This may not be the case, however, as an ontology may use, say "isA" as the name of an association, but define it as being the same as an entirely different concept in a different namespace.	Reintroduced associationName as a localId with the supportedAssociation mapping entries
Type is an attribute of entity, not usage. An entity can have multiple types	Bug Fix	sourceEntityType and targetEntityType are incorrect in the associations package, as they assume that the type is part of an entity's identity (i.e. you can have two entities with the same URI, one of which is a class and one an individual).	sourceEntityType and targetEntityType were removed from the model
Need to select associations by context	Enhancement	IHC needs to be able to select associations based on a passed context	added usageContext attribute to associatableElement
Need instance identifiers on associations	Bug Fix	You can assign an identifier to a DataProperty type association, but not to ObjectProperty type association. Both IHC and SNOMED-CT maintain unique identifiers on associations	Removed dataId from the associationData and created associationInstanceId in the associatableElement class. This field is optional, as dataId originally existed for LDAP compatibility.
Need to know whether an association participates in the definition of a concept	Enhancement	While OWL doesn't currently support this, it is useful to understand whether a assertion is considered to be part of the definition of an entity or simply an additional fact that is known about that entity.	Added isDefining attribute to the associatableElement class