FHIR Extension Modifications for UTG

Ted/GG: Make the history stuff work with provenance instead of extensions, and use contained provenance - GG to document how that's done.

Ted/GG: there exists some FHIR guidance and policy for deprecated code systems and value sets (these are still active, but deprecated which is essentially a usage advisory). Needs discussion so we can determine how best to do this (instead of an extension as listed lower down) for UTG.

Existing Extensions to be Changed

Existing Extension	New Extension	Comment	Definition
codesystem-history	resource-history	In addition to Code Systems, Concept Domains and Value Sets also have history items in the MIF. Extension needs to be generalized across vocabulary resources (and more generally for any FHIR resources that persist and need history tracked)	TK: We will use the Provenance resource to track history in a consistent and unified way for UTG objects (CodeSystem, ValueSet, NameSpace, ConceptMap) instead of using this existing extension (and changing it) Dave: No action
valueset-steward	resource-steward		Group claiming stewardship of this resource; usually an HL7 workgroup for HL7-defined and curated value sets. Mandatory element in the HL7 VSD. TK: we will not have steward separately for code systems, or the manifests (List resources) or NamingSystem in UTG so this extension is fine to use as it is;

			no change needed. Dave: Point to correct uri for value-set stewards DT: Neither extension is currently in use

New Extensions Required

New Extension	Comments	Definition	Applies To
concept-history	ConceptDomain HistoryItem is represented in UTG as an XML concept property to codesystem.concept	Date, Id and Text description of the history item	Concept TK: we will use Provenance for this. Open question: since Provenance references version-specific instances, not sure how to use it to provide a trail of history for changes to concepts. Dave: No action OR remove if temporary placeholder exists DT: Not in use
codesystem-contributor	Originating from MIF*: codeSystem.header.contributor* Optional codesystem attribute. Container only for sub-elements.	Contains information about the contributing organization and their role.	Codesystem TK: if this must be persisted then Ted and Lloyd will do a scan on the coremif source and decide where the material should go.. However it is likely never used and unimportant. Deal with later if we have to. Dave: Comment out if it exists ✔
codesystem-contributor.name	Originating from MIF*: codeSystem.header*	Contributing individual or organization, not	Codesystem TK: for all the HL7 objects, this should

	.contributor.name Required if contributor is present. Historically created in MIF and older entries need to be cleaned up.	always aligned with harmonization requestor. May or may not also be the Steward (intended to be an HL7 work group).	go in Publisher. For those handful that say HL7 is this but the actually it is an external organization, fix manually later as a content source flaw. Dave: Comment out if it exists✔
codesystem-contributor.role	Required if contributor is present. Currently all but one instance has value = sponsor. The only exception is the NANDA code system which currently has a value of “Publisher”. Exists as a string (not a code) Originates from the MIF: *codeSystem.header.contributor.role*	Role of contributing individual or organization.	Codesystem GG Note: Sponsor = commercial interest. TK; the only one we have in the coremif is “sponsor”. Do not import; we can fix any that need adjustment for the external NamingSystem entries later. Dave: Comment out if it exists✔
codesystem-contributor.notes	Originates from the MIF: *codeSystem.header.contributor.notes* Optional child of codesystem-contributor	Additional free text information describing the code system contributor. Should not contain the name or role.	Codesystem: TK: the only two we have in the coremif are ‘hsloc’ and ‘nhsn’. Do not import; we can fix if needed later inr the external NamingSystem entries. Dave: Comment out if it exists✔
codesystem-description	NOT required, code is already populating the correct		

	codesystem.description from codesystem.annotations.documentation.description.text		
codesystem-MIFNotation	Originates from the MIF: *codeSystem.header.legalese.notation* In UTG, Legalese uses notation sub-element as it's value. licenseTerms is used for codeSystem.copyright.	Legal notation information nested within MIF legalese information block. Typically misc. prose.	Codesystem TK: no new extension; this information populated here is ONLY in or not in UMLS, which we should NOT be putting in our codesystems regardless Dave: Comment out DT: Not in use
concept-openIssue	Covers concept level openIssue	Documents any open issues on a concept	Concept TK: do not import to UTG code system concepts. These will be extracted to an as yet-to-be-designed content flaw list. Dave: Do no import, remove if we have it ✔
resource-openIssue	Covers resource level openIssue (CodeSystem, ValueSet)	Documents any open issues on a resource instance	codesystem or valueset TK: do not import to UTG objects. These will be extracted to an as yet-to-be-designed content flaw list. Dave: Do no import, remove if we have it ✔
codesystem-relationshipKind		Describes types of relationships supported by the	Codesystem GG/Ted: Move these

Create properties, one for each of the MIF supported concept relationship type, for these. Each one that is used in the appropriate code system gets added as a code system property in the property definition for the UTG code system. The property types will be 'coding', descriptions propagate to property.description, all of the property attributes will be extensions on codesystem.property So we need one new extension for each. Lloyd is creating a change request for these to be added. Ted will define the properties in UTG concept properties This will get done the week of March 16.		CodeSystem.	to Jira vocab QA ticketing system. Do not reproduce in UTG. These will be documented in the as yet-to-be-designed content flaw list. TK/LM: we need to speak of this again with GG as for some code systems such as AddressPart (and some others) this is both important and should be surfaced (ie rendered for browsing of the terminology however we decide to represent it). Dave: Do no import, remove if we have it ✔ TK/GG discuss 3/15: "get Lloyd to hard code it for the MIF if only one or two code systems" Final decision as per group call:
codesystem-subsumes	Not a required extension, multiple parentage is handled with the following guidance: http://hl7.org/fhir/2018Dec/codesystem.html#hierarchy		
codesystem-hl7-approved-indicator	Boolean originating from MIF: codesystem.released	Indicates if the released version has been approved by	

	Version.hl7ApprovedIndicator Is **NOT** required as an extension. This indicator is TRUE for all internally maintained HL7 code systems. It is FALSE for the external ones. Identification within UTG is done in the manifest of externals.	HL7	
concept-history.backwardCompatibl		Indicates the change being documented in the concept-history extension is substantive	Concept TK: will be done as part of Provenance which will fully document each change, including this characteristic. Do not import. Dave: Do no import, remove if we have it ✔
concept-history.substantive		Indicates the change being documented in the concept-history extension is substantive	Concept TK: will be done as part of Provenance which will fully document each change, including this characteristic. Do not import. Dave: Do no import, remove if we have it ✔
resource-history.backwardCompatible	Boolean originating from MIF: *[codeSystem|valueSet].historyItem.isBackwardCompatibleChange*	Indicates the change being documented in the resource-history extension is substantive	codesystem or valueset TK: will be done as part of Provenance which will fully document each change, including this characteristic. Do not import.

			Dave: Do no import, remove if we have it ✔
resource-history.substantive	Boolean originating from MIF: *[codeSystem|valueSet].historyItem.isSubstantiveChange*	Indicates the change being documented in the resource-history extension is substantive	codesystem or valueset TK: will be done as part of Provenance which will fully document each change, including this characteristic. Do not import. Dave: Do no import, remove if we have it ✔
resource-versionDeprecated	Originates from MIF: *[codeSystem|valueSet].appInfo.deprecationInfo.deprecationEffectiveVersion*	Indicates that the use of the resource was deprecated as of the version indicated.	codesystem or valueset TK: will be done as part of Provenance which will fully document each change, including this characteristic. Do not import. Dave: Do no import, remove if we have it ✔
resource-versioningPolicy	Originates from MIF: *[codeSystem|valueSet].header.legalese.versioningPolicy*	Versioning Policy, if any, for the resource.	codesystem or valueset TK: for now, append this (if it exists in the MIF) to the CodeSystem.copyright string Dave: append string ✔
resource-hl7versionIntro	For all V2 resources, this will initially be set to 2.9 as this is the starting version for all V2 UTG code	The HL7 defined version in which the content was introduced	codesystem or valueset TK: This will be a new property for the concepts in v2tables.xml; for

	systems and value sets. For V3 code systems, this should be populated with the `<releasedVersion releaseDate="2019-12-15" publisherVersionId="29"` Values for the date released and the HxID version number if present in the coremif.		now all set to “2.9”. This will be used when a table was introduced in the past with no suggested values, then in a later Standard version values (value set and permaps code system) were added. Not used now; do this for release 2. Dave: see above (v2-version-introduced property) ✔
list.UTG-releaseID		Single string that is used in the publish-release instance manifests for all release types	TK: with the new design of the UTG publishing process, this is no longer needed. It will be part of the package list for a UTG release package. Dave: No action
resource-mif-extended-properties	Optional; present only when isMandatory, defaultValue, or defaultHandlingCode exist.	Documents additional attributes of codeSystem/supportedConceptProperty	codesystem or valueset TK/LM: these should be 3 separate extensions, not one with 3 parts Dave: Do no import, remove if we have it ✔
resource-mif-extended-properties.isMandatory	Optional Boolean		codesystem or valueset TK: requires an extension on property Dave: Do no import, remove if we have it ✔

resource-mif-extended-properties.defaultValue	Optional String		codesystem or valueset TK: requires an extension on property Dave: Do no import, remove if we have it ✔
resource-mif-extended-properties.defaultHandlingCode	Optional Code		codesystem or valueset TK: requires an extension on property Dave: Do no import, remove if we have it ✔
valueset-hl7-assocConceptProp		Sourced from valueSet/version/associatedConceptProperty	Valueset TK: add to UTG concept properties - critical for formal naming and RoseTree This is on code system properties, not valueset. Separate properties see below <code value="rim-assoc-conc-property"/> Dave: Import if not being imported (see concept properties) ✔
valueset-hl7-assocConceptProp.name	String Required		Valueset add to UTG concept properties - critical for formal naming and RoseTree This is on code system properties, not valueset TK: Done.

			<code value="rim-assoc-conc-propname"/> Dave: Import if not being imported (see concept properties) ✔
valueset-hl7-assocConceptProp.value	String Optional		Valueset add to UTG concept properties - critical for formal naming and RoseTree. This is on code system properties, not valueset <code value="rim-assoc-conc-propvalue"/> Dave: Import if not being imported (see concept properties) ✔
resource-deprecationInfo	String Sourced from MIF where resource definition contains a “Deprecation Comment” tag with a value.		codesystem or valueset TK: LM advises chat with GG as there is a FHIR policy on deprecation of resources; otherwise some additional code in C:PublicationStatus to indicate ‘active but deprecated’ would be needed. Not an extension. Dave: Do not import, remove if it exists ✔
resource-concept-binding-strength	Code sourced from codingStrength element of context bindings	Only used for context binding properties on concept domains	conceptDomain TK: additional property for conceptDomain

			concepts TK: four properties added for the 4 realms. Dave: Add strengths back ✔
codesystem-concept-comments	Sourced from v2 mdb in HL7TableValues.comments		
valueset-concept-comments	Sourced from v2 mdb in HL7TableValues.comments		
codesystem-concept-usageNotes	Sourced from v2 mdb in HL7TableValues.usage notes		
valueset-concept-usageNotes	Sourced from v2 mdb in HL7TableValues.usage notes		
codesystem-concept-commentsAsPub	Sourced from v2 mdb in HL7TableValues.comment as pub		
valueset-concept-commentsAsPub	Sourced from v2 mdb in HL7TableValues.comment as pub		
v2-concComment	Sourced from v2 mdb in HL7TableValues.comments		Concept TK: new property for the V2-sourced code systems. Dave: Import if not being imported (see concept properties)
v2-concCommentAsPub	Sourced from v2 mdb in HL7TableValues.co		Concep TK: new property for the V2-sourced

	mment as pub		code systems. Dave: Import if not being imported (see concept properties)
v2-usageNotes	Sourced from v2 mdb in HL7TableValues.usage notes		Concept TK: new property for the V2-sourced code systems. Dave: Import if not being imported (see concept properties)