Hungarian National Library Platform

Data model: Quintuplet, MARC21, BIBFRAME and Dublin Core

Miklós Lendvay
Hungarian National Széchényi Library
Dublin Core Metadata Initiative virtual 2021, 5. October
The Wiki Universe and the Library Domain
Entity based flexible data model

Distributed system real multitenancy

Flexible workflows
Parameter- and context-driven

Modularity; based on microservices
DATA-RELATED REQUIREMENTS

QUALITY LEVEL

MULTIPLE DATA EXCHANGE FORMATS

VALIDITY OF DATA FOR CERTAIN PERIOD OF TIME

FLEXIBLE WORKFLOWS FOR MANY TYPES OF AGENTS; PARAMETER AND CONTEXT-DRIVEN

VARIATIONS OF DATA AND COMPETING DATA

SOURCE OF INFORMATION

TRUSTWORTHINESS

VARIATIONS OF DATA AND COMPETING DATA

SOURCE OF INFORMATION

TRUSTWORTHINESS
- When creating a triplet, the predicate is not stored as the quality of the relationship between the records, instead the predicate is built into the relationship chain as a record.

- The common point of relationships is the statement that is able to make a piece of elementary statement about a given subject.

- The object of the statement may be another object, literal value, ‘itemized’ literal value.
The Structure of General Statements

- The “triplet” is used to define elementary statements
- To add more specific data, statements must be made about a statement
- All statements are equally true until we make a “false” statement about that statement
- The statement “tree” can be branched to infinity
- The framework does not provide guidance on how to deal with competing statements

Source / Copyright: HerMészSoft
Anatomy of Statements: The Structure of a Quintuplet

**SUBJECT:** The Subject is the Document to Which the Statement Applies

**SELECTOR:** The Position of the Statement Alongside the Dimensions of the Document Type of the Subject

**PREDICATE:** The Predicate is a Vocabulary Element Tipifying the Statement, with an Extendable Value Set

**OBJECT:** The Object is the Body of the Statement That Can Store a Literal Value, Point to an Entity Available in Another System

**LIFECYCLE:**
The Life Cycle of a Statement Carries, Among Other Things, the Time of Creation, the Creating Agent, as Well as the Beginning and the End of the Validity Period of the Statement, and the "Certainty" Classification of the Statement.
IIIF – localisation of abstract statements

- The framework specializes in displaying / visualising metadata
- The statements are placed on a virtual canvas
- At the visualisation of an image the given annotations, metadata can be placed in the viewer in an exact manner
- The abstraction formulated in the framework can be extended to all types of media content, by defining the appropriate coordinate system

Source / Copyright: HerMészSoft
Normalized Life Cycle Management of Statements

- An illustration of a hierarchy of conflicting statements
- Easy to select statements currently accepted
- Preserving the history of statements
- Statement protection: "Immutable" data
Cataloguing Module – Local Namespace

**ENTITY TYPE:**
The definition of the possible representations of the entities managed in the system.

**AVAILABLE PROPERTY:**
The definition of namespace elements created for types.

**ENTITY:**
Entities and records managed in the system.

**PROPERTY:**
Statements managed in the system.

**EVENT:**
Record of the changes in the system.

Source / Copyright: HerMészSoft
Customizable Set of Values for Record Types

- Entry types exist as part of the data model
- Possible statements (vocabulary) handled by a particular type are freely expandable
- At the statement level, the type of data, the precision of the data, the position of the data on the “canvas” defined by the statement can be defined
- Statement types protection: "Immutable" data

Source / Copyright: HerMészSoft
Authority Record simplified Graph Representation

- Individually configurable vocabulary set
- Elemental, individually positionable statements
- Normalized handling of complex data
- “Immutable” statements
- Historical managing
- Automatically derived name variants based on statements

Source / Copyright: HerMészSoft
- Entity name variants are formed automatically based on the statements on the entity and the specified configuration.
- Each name variant is represented by a separate Index item.
- For namespace entities, index items are populated.
- The item is constantly synchronized to changes in the entity.
- "Immutable" data
- Historical management

Source / Copyright: HerMészSoft
Identifying and Linking of Data

Functional Requirements for Authority Data

Resource Description and Access

Functional Requirements for Bibliographic Records

International Cataloguing Principles

Source / Copyright: Tiziana Possemato
Property Mapping

Contributor

dc:contributor
<name type= "personal"> 
bf:Contributor

MODS:

CONTRIBUTOR (property)
ISBN (property)
PUBLISHER (property)
TITLE (property)
LANGUAGE property

BOOK (object type)

Source / Copyright: HerMészSoft
More information about the projects:
Hungarian National Library Platform:
http://hnlp.oszk.hu
FOLIO: https://www.folio.org
https://wiki.folio.org

Miklós Lendvay, National Széchényi Library Hungary, lendvay.miklos@oszk.hu