Adapting MARC21 for Structuring Electronic Health Records Using JSON / MongoDB

LSC Bernardo Xartuni Magos[1], Dr. José Juan Hernández Mora[2], Dra. María Guadalupe Medina Barrera[3], Dr. Juan Ramos Ramos[4]

[1, 2, 3, 4]Tecnológico Nacional de México/Instituto Tecnológico de Apizaco, Carretera Apizaco Tzompantepec, Esquina Av. Instituto Tecnológico S/N, Conurbado Apizaco – Tzompantepec, Tlaxcla, Tlac. Mex. C.P. 90300
Email: *m23470002[at]apizaco.tecnm.mx*

Abstract: ***This paper presents an innovative adaptation of the bibliographic MARC21 (MAchine-Readable Cataloging) standard for structuring electronic health records (EHR). Traditionally used in libraries, MARC21 offers a highly flexible numbered field structure that, through systematic adaptation, provides significant advantages for managing complex medical information. The adaptation principles were applied, field mapping for clinical data was performed, and the implementation in a real system using MongoDB was carried out. It was demonstrated that this adaptation supports complex medical modeling while ensuring compatibility, extensibility, and traceability. The results were compared with established standards (HL7, FHIR), and the advantages, limitations, and appropriate use cases were discussed.***

Keywords: MARC21, Electronic Health Records, Interoperability, MongoDB, Health Informatics

1. Introduction

Efficiently managing medical information in electronic health records is among the most complex challenges in health informatics [1]. EHR systems must handle heterogeneous, dynamic, and highly structured data, from basic demographic information to detailed clinical records, laboratory studies, and diagnostic images [2].

1.1. The Challenge of Medical Data Structuring

Traditional approaches face significant limitations: rigid schemas in relational databases that hinder system evolution [3], lack of standardization between institutions limiting interoperability [2], growing complexity of medical data [4], and changing requirements due to evolving regulations and medical practices [5].

1.2. MARC21 and Justification for Adaptation

MARC21 is an international standard for representing bibliographic information developed by the Library of Congress [6]. Key characteristics include numbered field structure (001, 100, 245, etc.), alphabetic subfields ($a, $b, $c), repeatability as needed, and worldwide adoption [7].

The adaptation is justified by:
- Structural flexibility allowing information addition without breaking compatibility
- Adjustable granularity through subfields
- Proven scalability across millions of records
- Natural compatibility with NoSQL databases like MongoDB
- Ease of traceability and auditing through systematic numbering

1.3. Objectives

This work documents the principles of MARC21 adaptation for medical data, specific field mapping, MongoDB implementation, comparison with HL7 and FHIR, and analysis of advantages, limitations, and appropriate use cases.

2. Theoretical Framework

2.1. Existing Standards in Medical Informatics

HL7 (Health Level Seven): The most widely adopted standard for medical information exchange [2], with segment-based messaging (PID, OBR, OBX) and multiple versions (v2.x, v3, CDA, FHIR). Its primary limitation lies in its complex implementation requirements, especially for small institutions [8]

FHIR (Fast Healthcare Interoperability Resources): Modern HL7 standard based on RESTful APIs [9], with defined resources (Patient, Observation, Medication) in JSON/XML. Its challenge is requiring complex web service architecture and significant computational resources.

DICOM: Standard for medical image management [10], highly specialized in imaging but not designed for general clinical data.

2.2. NoSQL Databases

MongoDB and other NoSQL databases offer advantages for medical systems [11]: flexible schemas evolving without migrations, JSON documents naturally mapping to programming objects, horizontal scalability for large data volumes, and rich queries with specialized indexes. The combination of adapted MARC21 with MongoDB provides the best of both worlds: systematic structure with technical flexibility.

3. Adaptation Methodology

3.1. Fundamental Principles

The adaptation is based on five systematic principles:

Principle 1: Functional Field Ranges. A systematic division of MARC21 fields by function was established (Table 1), based on the conceptual structure of MARC21 [6] but adapted to medical needs.

Table 1: Functional ranges of adapted MARC21 fields

Range	Function	EHR Use
001-009	Identification	Unique codes, timestamps
020-099	References	Links to records, patients
100-199	Personal info	Demographic data
200-299	General info	General record data
300-499	Specific info	Specialized clinical data
500-799	Extended info	Additional details
800-899	Metadata	Classification, categorization
900-999	System	Auditing, status, control

Principle 2: Semantic Subfields. Each field can contain subfields with lowercase letters maintaining specific semantics (Avram, 2003). Example: field 100 with subfields a=first name, b=paternal surname, c=maternal surname, d=birth date, e=sex, f=national ID.

Principle 3: Unique Codes. Field 001 contains auto-generated unique code: PREFIX-NNNNNN (e.g., PAC-000001, DIAG-000523).

Principle 4: MARC Timestamps. Field 005 uses standard format YYYYMMDDHHMMSS.sss (Library of Congress, 2000).

Principle 5: Metadata in Field 900. Consistently reserved for system metadata (active status, creator user, creation date).

3.2. Record Type Mapping

3.2.1. Medical Catalogs

Catalogs (religions, nationalities, diagnoses) follow base structure:

Listing 1: Base catalog structure

```
{
  "001": "CAT-000001",
  "005": "20250117120000.000",
  "100": {
    "a": "Main name"
  },
  "200": {
    "a": "Description"
  },
  "active": true,
  "createdAt": "ISODate(...)",
  "updatedAt": "ISODate(...)"
}
```

3.2.2. Patient Records

Patient structure uses ranges 100-600:

Listing 2: Patient structure (simplified)

```
{
  "001": "PAC-000001",
  "100": { // Personal data
    "a": "Juan",
    "b": "Perez",
    "d": "1985-03-15",
    "e": "M"
  },
  "200": { // Catalog references
    "a": [
      { "code": "REL-001", "name": "Catholic"
}
    ]
  },
  "400": { // Contact
    "a": [
      { "street": "Main Ave", "city":
"Puebla" }
    ],
    "b": ["222-123-4567"]
  },
  "900": { // Metadata
    "a": true,
    "d": "ISODate(...)"
  }
}
```

3.2.3. ICD-10 Diagnoses

Most complex catalog requiring 25+ fields:

Listing 3: ICD-10 diagnosis (partial)

```
{
  "001": "DIAG-000523",
  "100": {
    "a": "E11",
    "b": "Type 2 diabetes mellitus"
  },
  "200": {
    "a": "IV", // Chapter
    "b": "Endocrine diseases"
  },
  "300": {
    "a": "BOTH", // Applicable sex
    "b": "18-99" // Age range
  }
}
```

4. Technical Implementation

4.1.1. MongoDB Integration

Implementation leverages NoSQL features while maintaining MARC21 structure [11].

4.1.2. Specialized Indexes

Strategic indexes are created to optimize frequent queries:

Listing 4: MongoDB indexes

```
// 1. Unique index on field "001"
// Ensures that not two documents share the
same value in field "001".
collection.createIndex(
  { "001": 1 },
  { unique: true }
);

// 2. Compound index on specific subfields
// Optimizes queries that filter by "100.a",
"100.b", and "100.f" together.
collection.createIndex(
  { "100.a": 1, "100.b": 1, "100.f": 1 }
);

// 3. Compound index with text fields
// Combines an ascending index on "001" with
text indexes on "100.a" and "100.b"
// for text searches ($text) on those fields.
collection.createIndex(
  { "001": 1, "100.a": "text", "100.b":
"text" }
);
```

Indexing on specific subfields is a MongoDB advantage that aligns with MARC21 structure [12].

4.2. Automatic Code Generation

Counter system for unique codes:

Listing 5: Unique code generation

```
/**
 * Generates a new sequential code based on a
prefix.
 * Uses a "counter" collection to handle the
increment atomically.
 */
async function generateCode(prefix) {

  // 1. Finds a counter document by its
prefix (e.g., "PAC")
  //    and increments the "value" field by
1.
  const result = await
counterCollection.findOneAndUpdate(
    { _id: prefix },          // The filter:
searches by prefix (e.g., _id: "PAC")
    { $inc: { value: 1 } },   // The
operation: increments "value"
    {
      upsert: true,           // If it
doesn't exist, create it (with value: 1)
      returnDocument: 'after' // Returns the
document AFTER updating it
    }
  );

  // 2. Formats the result.
  //    result.value will contain the new
number (e.g., 123)
  return `${prefix}-
${String(result.value).padStart(6, '0')}`;
  // Example output: "PAC-000123"
}
```

4.3. Embedded Data vs References

Key architectural decision: for frequently queried entities, complete catalog data is embedded in field 200. Advantages: single query retrieves all data, optimal performance, preserved history. Disadvantage: requires synchronization when catalogs update [11]. A synchronization method was implemented that updates all records referencing a catalog when it changes, maintaining consistency while optimizing queries.

4.4. Schema Validation

Although MongoDB is schema-less, explicit validation was implemented using JSON Schema and the AJV library, providing data security without losing MongoDB flexibility.

5. Comparison with established standards

5.1 MARC21-EHR vs HL7 v2

Table 2: MARC21-EHR vs HL7 v2 comparison

Aspect	MARC21-EHR	HL7 v2
Purpose	Structured storage	Message exchange
Complexity	Low	High
Implementation	Direct in NoSQL	Specialized parsing
Interoperability	Requires mapping	Native between HL7 systems
Cost	Low	Moderate-High

HL7 v2 is superior for information exchange between heterogeneous systems [2], while MARC21-EHR is optimal for internal storage with flexibility.

5.2 MARC21-EHR vs FHIR

FHIR is superior for systems requiring international interoperability [9]. MARC21-EHR is appropriate for: institutions with limited resources, internal systems without immediate interoperability requirements, and projects prioritizing development speed.

5.3 Hybrid Strategy

An optimal strategy combines: (1) internal storage with MARC21-EHR for flexibility, (2) translation layer MARC21-EHR ↔ FHIR for external exchange, and (3) selective FHIR resource implementation only for exchanged data.

6. Advantages and Limitations

6.1 Observed Advantages

a) **Flexibility without migrations:** Adding new fields does not require modifying existing records. MongoDB allows heterogeneity without performance penalty [11].
b) **Traceability:** Systematic numbering facilitates audits ("review changes in field 100.f"), analysis of most modified fields, and selective migration.
c) **Self-documenting:** Hierarchical structure is self-documented (001=ID, 100=Personal data, 100.a=Name), reducing errors and facilitating onboarding.
d) **Unexpected compatibility:** Existing MARC21 tools can be used for test record creation, structure validation, and XML conversion.

6.2 Identified Limitations

a) **Lack of global standardization:** Each implementation defines its schema, limiting direct interoperability. Mitigation: exhaustive documentation and translators to FHIR/HL7.
b) **Learning curve:** Developers without MARC21 experience require time. Mitigation: detailed documentation, complete examples, code templates.
c) **No formal semantics:** Unlike SNOMED CT, MARC21 does not provide computational semantics. Mitigation: integrate standard codes in subfields.
d) **Complex queries:** Queries crossing multiple levels can be verbose. Mitigation: materialized views or specialized indexes.

7. Practical Use Cases

7.1 Medical Catalog System

Regional hospital managing 10+ catalogs with uniform base structure. Results: uniformity (all share code base), extensibility (new catalog takes 2-3 hours), maintainability (changes apply easily).

7.2 Record with Embedded Data

Fast access to complete information without joins. Single query retrieves complete data, listing 1000 patients in 150-200ms, without need for 3-4 additional joins.

7.3 Varied Clinical Documents

Multiple types (admission notes, nursing sheets) with fields 001-200 standard, fields 300-800 specific. Advantage: common fields allow generic queries independent of specific type.

8. Lessons Learned

8.1 Best Practices

- **Exhaustive documentation:** Regularly updating documentation for each field and subfield is essential.
- **Utility functions:** Centralizing MARC21 structure generation reduces errors and improves consistency.
- **Consistent validation:** Applying JSON Schema at all endpoints ensures data quality.
- **Strict conventions:** Lowercase subfields, PREFIX-NNNNNN codes, MARC21 timestamps, collections with prefixes.

8.2 Common Errors

- **Subfield overload:** Solution: use arrays when semantically appropriate.
- **Forgetting field 900:** Solution: middleware that adds metadata automatically.
- **Inconsistent timestamps:** Solution: centralized function for timestamps.

9. Conclusions

9.1 Contributions

This work presents: (1) systematic MARC21 adaptation for EHR with formal principles, (2) detailed mapping for multiple record types, (3) practical MongoDB implementation, (4) objective comparison with HL7/FHIR, and (5) real use cases demonstrating viability.

9.2 Viability

MARC21 adaptation is viable for small-medium institutions with limited resources, internal systems without immediate interoperability requirements, projects prioritizing flexibility and speed, and initial phases of projects that will migrate to FHIR later.

9.3 Limitations

The approach: is not an international standard, requires adapters for interoperability, lacks formal computational semantics, and requires exhaustive mapping documentation.

9.4 Recommendations

Institutions should: (1) evaluate interoperability requirements, (2) document mapping in detail, (3) plan future FHIR migration, (4) train team with abundant documentation, and (5) validate consistently with JSON Schema.

9.5 Future Work

Promising directions include mapping standardization for Spanish-speaking community, automated tools (generators and validators), bidirectional MARC21 ↔ FHIR translator, integration with ontologies (SNOMED CT, LOINC), query optimization, and structural consistency verifier.

9.6 Final Reflection

MARC21 adaptation demonstrates that mature standards from other domains can provide practical solutions. While not replacing FHIR, it offers a viable alternative balancing flexibility, simplicity, and functionality for institutions with limited resources. Success relies on thorough documentation, strict adherence to conventions, planning for interoperability, and honest recognition of limitations.

References

[1] E. H. and C. J. J. Shortliffe, *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, 4th ed. London: Springer, 2014.

[2] T. Benson, *Principles of Health Interoperability HL7 and SNOMED*. London: Springer, 2010.

[3] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," *Commun ACM*, vol. 13, no. 6, pp. 377–387, 1970, doi: 10.1145/362384.362685.

[4] W. R. Hersh *et al.*, "Caveats for the Use of Operational Electronic Health Record Data in Comparative Effectiveness Research," *Med Care*, vol. 51, no. 8 Suppl 3, pp. S30–S37, 2013, doi: 10.1097/MLR.0b013e31829b1dbd.

[5] Committee on Patient Safety and H. I. Technology, *Health IT and Patient Safety: Building Safer Systems for Better Care*. Washington, DC: National Academies Press, 2012.

[6] Library of Congress, *MARC 21 Format for Bibliographic Data*. Washington, DC: Library of Congress, 2000.

[7] H. D. Avram, *MARC: Its History and Implications*. Washington, DC: Library of Congress, 2003.

[8] D. Bender and K. Sartipi, "HL7 FHIR: An Agile and RESTful Approach to Healthcare Information Exchange," in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, 2013, pp. 326–331. doi: 10.1109/CBMS.2013.6627810.

[9] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "SMART on FHIR: A Standards-

Based, Interoperable Apps Platform for Electronic Health Records," *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 899–908, 2016, doi: 10.1093/jamia/ocv189.
[10] O. S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*, 2nd ed. Berlin: Springer, 2012.
[11] K. Chodorow, *MongoDB: The Definitive Guide*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2013.
[12] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in Action*, 2nd ed. Manning Publications, 2016

Author Profile

Bernardo Xartuni Magosgas a degree in Computer Systems from the Universidad Popular Autónoma del Estado de Puebla. He is currently studying the masters in Computer Science in Software Engineering from the TecNM / Instituto Tecnológico de Apizaco.

José Juan Hernández Mora is a Computer Engineer from the Autonomous University of Tlaxcala. He holds a Master of Science in Computer Science from the National Center for Research and Technological Development (CENIDET), in Cuernavaca, Morelos, and a Doctorate in Teaching Excellence from the University of Los Angeles. He is a Professor with Desirable Profile as designated by PRODEP, is the leader of the academic group "Information Systems," and holds the rank of Candidate Member in the SNII of Conahcyt. His research areas include: Software Engineering, Development of Information Technology Applications, Digital Image Processing (DIP), Artificial Neural Networks (ANN), Heutagogy, and Cybergogy.

Dr. María Guadalupe Medina Barrera Doctor in Strategic Planning and Technology Management from the Universidad Popular Autónoma del Estado de Puebla (UPAEP). She holds a Master's degree in Computer Science from the National Center for Research and Technological Development (CENIDET) and a Bachelor's degree in Informatics from the Instituto Tecnológico de Tepic, Nayarit, Mexico. She holds the rank of Candidate Member in the National System of Researchers (SNII) of CONAHCYT, is recognized with the Desirable Profile designation by PRODEP, and is a member of the academic group "Information Systems." Her research areas include: Software Project Management and Development, Process Automation, Human-Computer Interfaces, and Pattern Recognition.

Juan Ramos Ramos has a degree in Computer Science from the Instituto Tecnológico de Apizaco, from 1993. He is also a Master in Computer Science and Telecommunications from the Instituto de Estudios Universitarios, A.C.; He also holds a Doctorate in Computer Systems from the University of the South; he works as a full-time professor at the Tecnológico Nacional de México (TecNM) / Instituto Tecnológico de Apizaco in the area of Systems and Computing, teaching at the undergraduate and postgraduate level, in the areas of Programming and Software Engineering..