Context-free grammar-guided generation of FHIR resources using Large Language Models

Johann Frei, Frank Kramer

Angaben zur Veröffentlichung / Publication details:

Frei, Johann, and Frank Kramer. 2026. "Context-free grammar-guided generation of FHIR resources using Large Language Models." In *Opening the personal gate between technology and health care: proceedings of MIE 2026*, edited by Mauro Giacomini, Jaime Delgado, Theodoros N. Arvanitis, Elisavet Andrikopoulou, Arriel Benis, Gabriella Balestra, Riccardo Bellazzi, et al., 2498–2502. Amsterdam: IOS Press. https://doi.org/10.3233/shti260725.

Opening the Personal Gate between Technology and Health Care
M. Giacomini et al. (Eds.)

doi:10.3233/SHTI260725

Context-Free Grammar-Guided Generation of FHIR Resources Using Large Language Models

Johann FREI[a,1] and Frank KRAMER[a]
[a]*University of Augsburg*
ORCiD ID: Johann Frei https://orcid.org/0000-0003-0323-0904, Frank Kramer https://orcid.org/0000-0002-2857-7122

Abstract. Large Language Models (LLMs) have shown remarkable capabilities in medical information extraction and data transformation tasks. However, their unstructured outputs pose significant challenges for integration into clinical data pipelines. Fast Healthcare Interoperability Resources (FHIR) provides a standardized data model for encoding and exchanging medical information, yet its complexity, involving deeply nested structures, standardized coding systems, and strict syntactical rules, makes reliable generation challenging. While LLMs can be guided by linguistic grammars during token decoding to enforce structured outputs, designing such grammars for complex standards like FHIR remains difficult. We present a pipeline for synthesizing Context-Free Grammars (CFGs) from FHIR-derived JSON Schemas, enabling schema-conformant structured generation with LLMs. Our approach introduces relaxed key ordering to maintain structural validity while avoiding limitations of strict schema-guided approaches. Through systematic evaluation on 17 test cases across 5 categories using FHIR R5 resources, we demonstrate that CFG-guided generation achieves 76.5% validity in producing valid resources, compared to 41.2% for JSON Schema-guided and 17.6% for unguided generation. Furthermore, our approach successfully enforces value constraints (enumerated types, certain UCUM unit codes) and structural compliance that other methods fail to guarantee. We release our grammar synthesis pipeline as open-source to facilitate adoption in healthcare information systems.

Keywords. Large language models, Constrained decoding, Grammar-guided generation, Structured output, FHIR

1. Introduction

Fast Healthcare Interoperability Resources (FHIR) has emerged as the dominant standard for health data exchange, adopted by major electronic health record (EHR) systems and regulatory bodies worldwide. FHIR defines numerous resource types (e.g., Patient, Condition, MedicationStatement) with complex nested structures, required coding systems (e.g., SNOMED CT, LOINC), and version-specific structural requirements. As the evolution of the standard introduces structural changes, it also complicates automated resource generation and requires careful attention to version-specific schemas. Large Language Models (LLMs) offer powerful capabilities for extracting structured medical

[1] Corresponding Author: Johann Frei, johann.frei@informatik.uni-augsburg.de

information from unstructured clinical text. However, their probabilistic nature produces outputs that frequently violate FHIR's structural requirements, use deprecated field names, or generate invalid coding values. Traditional approaches to structured generation include post-hoc validation (e.g., Pydantic or FHIR validation), iterative refinement loops, or code-based construction, each with limitations in reliability, latency, or complexity.

Recent work has explored various strategies for LLM-based FHIR generation. Tabari et al. [1] employ iterative refinement through LLM-validator loops to achieve schema compliance, though this approach incurs multiple inference cycles. Frei et al. [2] use agentic code-based resource construction, ensuring validity through programmatic instantiation but requiring complex orchestration logic. Li et al. [3] demonstrate LLM-based extraction of FHIR medication statements achieving over 90% exact match rates, though requiring task-specific NLP pipelines. These approaches demonstrate the viability of LLM-based FHIR generation while highlighting the trade-offs between validation strategies. Constrained decoding approaches can enforce structural validity by grammar rules that guide the LLM's token generation process. Grammar-constrained decoding has proven effective across diverse structured NLP tasks, outperforming unconstrained models without requiring task-specific fine-tuning [4]. Context-Free Grammars (CFGs) and JSON Schema are two prominent approaches, but they present different trade-offs. JSON Schema validation is well-established and easy to use, but can be overly strict, particularly regarding key ordering, depending on the implementation of the parser. CFGs offer greater expressiveness and flexibility but require careful design.

We present a pipeline that synthesizes Context-Free Grammars from FHIR-derived JSON Schemas to enable reliable constrained generation with LLMs. We generate JSON Schemas for selected FHIR resource types, apply targeted field filtering to focus on structured clinical data, and transform these schemas into Extended Backus-Naur Form (EBNF) grammars. Our grammar synthesis incorporates domain-specific constraints and introduces relaxed key ordering to mitigate generation traps that trigger undesired behaviors such as infinite field repetition or field omission caused by misalignments between schema constraints and LLM token preferences. Through systematic evaluation, we demonstrate that CFG-guided generation significantly outperforms JSON Schema-guided and unguided approaches in producing valid FHIR resources while enforcing domain-specific constraints critical for clinical data quality. We release our pipeline, test cases, and evaluation framework as open-source.

2. Methods

We focus our implementation on FHIR Release 5 (R5), which represents the latest published version of the FHIR standard as of this work. Our pipeline transforms FHIR JSON Schemas into CFGs suitable for LLM-guided generation through four stages:

2.1. Stage 1: Schema Generation

We use the *fhir.resources*[2] Python library (version 7.1.0) to programmatically generate JSON Schemas for four FHIR resource types: Patient, Condition, Procedure, and

[2] https://github.com/nazrulworld/fhir.resources/tree/7.1.0

MedicationStatement. These resources represent diverse structural patterns including deeply nested objects, arrays, choice types, and reference relationships.

2.2. Stage 2: Schema Filtering

To focus on structured clinical data, we systematically remove metadata (*id*, *meta*, *implicitRules*, *language*), narrative fields (*text*, *contained*), extensibility mechanisms (*extension*, *modifierExtension*), and specific FHIR types (*Annotation*, *Attachment*, *Meta*). Resource-specific fields less relevant for structured encoding (e.g., *photo*, administrative fields) are also excluded. These filters are configurable for specific use cases.

2.3. Stage 3: BNF Grammar Generation

We adapt the *json_schema_to_grammar.py*[3] script from llama.cpp to convert the filtered JSON Schema to a Backus-Naur Form (BNF) grammar. Key modifications include:

- **Selective enum enforcement**: Terminal symbols for constrained vocabularies in critical fields (e.g., *Patient.gender*, *MedicationStatement.status* restricted to *recorded*, *draft*, *entered-in-error*)
- **UCUM codes**: Unit codes restricted to valid UCUM values for timing fields (e.g., *Timing.repeat.periodUnit* accepts *h*, *d*, *wk*)
- **Relaxed ordering**: Required fields maintain strict ordering, but optional fields can appear on either side of required fields

The relaxed ordering addresses generation traps that may occur when, for instance, the LLM intends to generate a certain field during the document generation process, but is unable to do so due to the enforced schema restrictions. This misalignment between the schema and the LLM's intent may cause infinite repetition loops (repeatedly generating the same field or structure) or catastrophic field omission (skipping required fields when they don't align with the LLM's preferred generation order). Our approach permits ordering flexibility at the cost of possible field duplication, which can be resolved in post-processing.

2.4. Stage 4: EBNF Conversion

We translate BNF grammars to the Extended Backus-Naur Form (EBNF) required by the Outlines library [5] by parsing the BNF's abstract syntax tree (AST) and transforming the AST into an EBNF instantiation.

3. Results

3.1. Grammar Statistics

The synthesis pipeline generated a CFG comprising 364 production rules spanning 32 distinct FHIR- and JSON-specific types. The grammar exhibits hierarchical structure that

[3] https://github.com/ggml-org/llama.cpp/blob/669912d9a5bf927312c553332ff997f0a99da8fb/examples/json_schema_to_grammar.py

reflects FHIR's complex nested representations. The four primary resource types contribute 135 rules, with MedicationStatement generating 28 rules, Procedure 44 rules, Condition 39 rules, and Patient 24 rules.

3.2. Experimental Results

Model and Token Sampling. We use Llama 3.3 70B-Instruct [6] (quantized) with temperature 0 for deterministic outputs. Generation employs Outlines (version 1.2.7) for CFG-guided and JSON Schema-guided approaches.

Test Case Design. We designed 17 test cases across five categories to systematically evaluate constrained value enforcement (enum types and UCUM units), schema robustness (non-standard ordering and forbidden fields), version compliance (R5 vs R4 structures), structural validity (resource type consistency, nesting, arrays), and output cleanliness. Two cases deliberately requested schema-forbidden fields to demonstrate grammar failure modes.

Evaluation Methodology. Generated outputs are validated using *fhir.resources* R5 validators and classified as Valid (parseable and semantically correct), Partial (parseable with minor semantic errors), or Invalid (non-parseable or hallucinated structures). CFG-guided generation achieved 13 valid and 2 partial outputs (76.5% valid, 88.2% parseable), significantly outperforming JSON Schema-guided (41.2% valid, 7 of 17) and unguided generation (17.6% valid, 3 of 17), as shown in Table 1. The individual test cases and outputs are provided as material on GitHub.[4]

Table 1. Aggregate results across 17 test cases

Classification	CFG-guided	JSON Schema-guided	Unguided
Valid	13 (76.5%)	7 (41.2%)	3 (16.6%)
Partial	2 (11.8%)	0 (0%)	0 (0%)
Invalid	2 (11.8%)	10 (58.8%)	14 (82.4%)

Constrained Value Enforcement. CFG successfully enforced enumerated types and UCUM codes, rejecting invalid R4 status values and converting plain-text units (*hours*, *days*) to valid UCUM codes (*h*, *d*), while JSON Schema and unguided approaches retained invalid values. Two limitations emerged: the grammar ensured syntactical integrity but the LLM occasionally selected semantically inappropriate values (e.g., *entered-in-error* instead of *recorded*), and incomplete UCUM tables led to suboptimal unit selection (*week* → *d* instead of *wk*).

Robustness to Non-standard Ordering. CFG's relaxed ordering proved critical for non-standard attribute order, generating valid output while JSON Schema omitted required fields and hallucinated invalid structures. Both correctly excluded forbidden metadata fields, while unguided generation occasionally hallucinated these fields.

Version Compliance. CFG and JSON Schema consistently generated R5-compliant structures, while unguided generation produced deprecated R4 structures in 82.4% of cases.

Structural Reliability. JSON Schema exhibited unpredictable catastrophic failures (wrong resource types, empty documents), contrasting with CFG's consistent behavior.

Grammar Failure Modes. Deliberate forbidden field requests revealed trap behavior: some unavailable fields triggered infinite repetition loops, while others caused hallucinated complex structures.

[4] https://github.com/j-frei/CFG4FHIR

4. Discussion

CFG-guided generation achieved 76.5% validity with three key advantages: automatic constraint enforcement prevents downstream validation errors; relaxed ordering avoids generation traps from strict field sequences; and predictable failure modes addressable through grammar refinement or prompt engineering. JSON Schema's 41.2% success masks critical unpredictability with sporadic catastrophic errors (wrong resource types, empty outputs), making it less reliable for production without extensive prompt engineering. Unguided generation's 17.6% success stems from systematic R4/R5 confusion, suggesting training data dominated by older FHIR versions. CFG limitations include semantically inappropriate code selection (addressable through additional prompt context), limited enum and UCUM field coverage (expandable via grammar rules), and unpredictable trap behavior for forbidden fields. Future work should extend evaluation to additional resource types and LLMs, and validate with domain experts. For healthcare applications requiring automated FHIR generation, CFG-guided approaches provide crucial syntactic schema conformance with 76.5% validity, acceptable when combined with error handling. JSON Schema reliability depends on parser implementation, while unguided generation requires extensive post-processing.

5. Conclusion

We presented a pipeline for synthesizing CFGs from FHIR JSON Schemas enabling reliable LLM-guided generation of FHIR R5 resources. Across 17 test cases, CFG-guided generation achieved 76.5% validity versus 41.2% for JSON Schema and 17.6% for unguided approaches. Relaxed key ordering mitigates generation traps while automatic constraint enforcement ensures clinical data quality on a syntactic level. Identified limitations in semantic value selection and UCUM table coverage are addressable through additional prompt context and grammar refinement. The pipeline, test cases, and evaluation framework are available at https://github.com/j-frei/CFG4FHIR.

References

[1] Tabari P, Piscitelli A, Costagliola G, de Rosa M. Assessing the Potential of an LLM-Powered System for Enhancing FHIR Resource Validation. In: Intelligent Health Systems–From Technology to Data and Knowledge. IOS Press; 2025. p. 803-7.
[2] Frei J, Feldhus N, Raithel L, Roller R, Meyer A, Kramer F. Infherno: End-to-end Agent-based FHIR Resource Synthesis from Free-form Clinical Notes. arXiv. 2025.
[3] Li Y, Wang H, Yerebakan HZ, Shinagawa Y, Luo Y. FHIR-GPT Enhances Health Interoperability with Large Language Models. NEJM AI. 2024.
[4] Geng S, Josifoski M, Peyrard M, West R. Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning. In: Bouamor H, Pino J, Bali K, editors. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics; 2023.
[5] Willard BT, Louf R. Efficient Guided Generation for Large Language Models. arXiv. 2023.
[6] Meta AI. Llama 3.3 – 70B Instruction-Tuned Model; 2025. Instruction-tuned 70 B parameter model. https://www.llama.com/models/llama-3/