Bachelor Thesis: JSON DTDs

Meike Klettke,
in cooperation with external reviewers/supervisors:
Antoon Bronselaer and Uta Störl

February 2026

1 Motivation

Document formats for storing semi-structured data have been around for a long time. XML and JSON are formats designed for this task.

XML was developed some time ago, and the document format was developed together with a simple schema language (DTD—Document Type Definition). This schema language is intuitive, easy to learn, understandable, and, thanks to these characteristics, widely used. It includes:

- Definition of elements: sequences, alternatives, simple cardinality specifications (?, +, *), mixed content
- Definition of attributes with simple data types

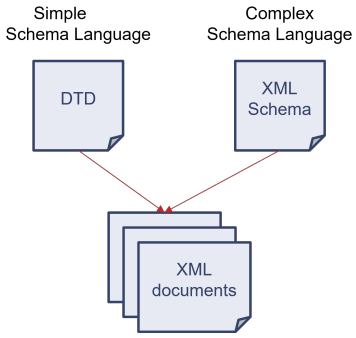

Figure 1: Schema Languages for XML documents

However, the simplicity of DTDs was also seen as a shortcoming. A few years later, XML Schema was developed, a powerful schema language that, in addition to the schema information also contained in DTDs, can also define data types and various constraints. However, this expressive language is also more difficult to read and is rarely created manually (more often generated from conceptual

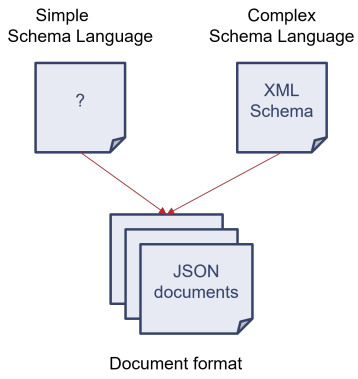

Figure 2: Schema Languages for JSON documents

models or XML sample data). Figure 1 sketches these schema languages for XML.

During the development of JSON, computer scientists have been able to build on the experience gained with XML. In this case, no simple schema language was developed; development began with the so-called JSON Schema, a language that corresponds to XML Schema in terms of its capabilities. The disadvantages of such a complex schema language are also visible in JSON Schema; the language is often generated (from conceptual models or JSON sample data) but rarely written manually. This is due in part to the less compact syntax, but also to the many different variants that the language offers. Figure 2 visualizes the currently available schema language for JSON.

Looking at the advantages and disadvantages of the available schema languages, there are many potential use cases for a *very simple JSON schema language*. Similar to DTDs, in this Bachelor's thesis language proposal for this purpose shall be developed.

The goal is to develop nothing more and nothing less than the *Document Type Definition (DTD) for JSON documents* under the name JDTD. To implement this, a language proposal should be made that is closely based on DTDs and addresses the special features of the JSON syntax.

2 Subtasks

In this Bachelor thesis the following subtasks shall be solved:

1. State-of-the-Art: analysis of available schema languages for structured (relational) and semi-structured data
2. comparison of the different language proposals and the characteristics/ dimension which can be defined with each of the languages
3. selection of characteristics to be expressed in the new schema language

4. definition of a suitable syntax proposal [1]

5. Development of sample schemas (in the DTD for JSON) and accordingly valid and non valid JSON documents

3 Literature

- XML and DTDs: https://www.w3.org/TR/xml11/
- Meike Klettke, Holger Meyer: XML & Datenbanken, dpunkt.verlag
- XML schema: Tutorial:

  `https://www.w3schools.com/xml/schema_intro.asp`

- JSON, https://www.json.org/json-en.html
- JSON schema, https://json-schema.org/
- Pierre Bourhis, Juan L. Reutter, Fernando Suárez, Domagoj Vrgoc: *JSON: Data model, Query languages and Schema specification.* PODS 2017: 123-135
- Kehinde Sotonwa, Johnson Adeyiga, Michael Adenibuyan, Moyinoluwa Dosunmu: Survey of Schema Languages: On a Software Complexity Metric. FICC (2) 2023: 349-361
- Oluwadamilare Falola, Sanjay Misra, Adewole Adewumi, Robertas Damasevicius: Evaluation and Comparison of Metrics for XML Schema Languages. ICADIWT 2017: 51-59

[1] Due to the acceptance of the new language, either a syntactic reference to DTDs is desirable or the choice of a JSON syntax.