S2DM: a Simplified Semantic Data Modeling approach

Daniel Alvarez-Coello[1,*], Adnan Bekan[1], Sebastian Schleemilch[1], Daniel Wilms[2], Stephen Lawrence[3], Ted Guild[4], Amir Sayegh[4], Chaitanya Podalakuru[5], Sebastian Schildt[6] and Andreas Achtzehn[7]

[1]*BMW Group. New Technologies, Digital Vehicle Infrastructure. Munich, Germany.*
[2]*SPREAD GmbH. Berlin, Germany.*
[3]*Renesas Electronics. Software & Digitalization Group, Platform Department. Bourne End, UK.*
[4]*Geotab Incorporated. Oakville, Canada.*
[5]*Ford Motor Company. In-Vehicle Network Architecture. Michigan, US.*
[6]*ETAS GmbH. Stuttgart, Germany.*
[7]*Robert Bosch GmbH, Cross-Domain Computing Solutions. Renningen, Germany*

Abstract

Controlled vocabularies play a central role in data-centric ecosystems, and their success depends heavily on the input of domain experts who are rarely trained in semantic data modeling. Consequently, the modeling language must remain easily approachable and lead to actionable outcomes. While some existing standards such as OWL or SHACL offer semantic rigor, their complexity can hinder adoption, leaving a gap between practical modeling needs and formal semantics. In this position paper, we argue that this gap should be bridged by a simplified approach that lowers the entry barrier for domain experts while remaining compatible with established standards. Based on relevant design principles, we outline the core ideas of a Simplified Semantic Data Modeling (S2DM) approach. S2DM builds on GraphQL SDL for schema definition and SKOS for lightweight classification and thesaurus features. Originally motivated by work in the COVESA automotive alliance, the approach is envisioned to be domain-independent. We conclude by highlighting directions for future work, aiming to refine the approach, strengthen interoperability, and validate it in transport and beyond.

Keywords

Data modeling, semantic data models, controlled vocabularies, GraphQL SDL, SKOS, data-centric architecture

1. Introduction

The importance of data is widely acknowledged. However, it is no longer sufficient to base decisions solely on data (i.e., to be data-driven), which has become the new norm. Equally important is having appropriate approaches for the development and evolution of vocabularies that underpin data exchange in the physical layer (e.g., databases, APIs, or applications). As domain models expand in scope and complexity, maintaining and enriching them becomes increasingly challenging. This challenge is not unique to a particular organization but is common across data-centric ecosystems.

A relevant example is the Connected Vehicle Systems Alliance (COVESA),[1] which has maintained for almost a decade the Vehicle Signal Specification (VSS) [1], a vocabulary of vehicle properties. VSS adopts a tree-based hierarchical structure and uses *YAML* with custom constructs to specify aspects such as *description*, *datatype*, and *unit*. Its simplicity has allowed many Subject Matter Experts (SMEs) (i.e., domain experts) to contribute with minimal data modeling expertise. Yet, as the vocabulary and the vehicle ecosystem evolve, the demand for semantic clarity and alignment with established standards has made it clear that the modeling approach must adapt. What began as an initiative to improve the

6th International Workshop On A Semantic Data Space For Transport. Co-located with the 21st International Conference on Semantic Systems (SEMANTiCS). September 03–05 of 2025. Vienna, Austria.
*Corresponding author.
daniel.alvarez-coello@bmwgroup.com (D. Alvarez-Coello)
0000-0001-5663-7543 (D. Alvarez-Coello); 0009-0008-8029-1472 (S. Lawrence); 0000-0002-3373-0224 (T. Guild); 0009-0000-7109-1307 (A. Sayegh); 0000-0002-8036-8651 (S. Schildt)

[1]https://covesa.global

modeling language of VSS has revealed a broader opportunity: to design a domain-agnostic Simplified Semantic Data Modeling (S2DM) approach with applicability beyond the automotive domain.

While the value of long-standing semantic standards is recognized, practical experience with the VSS project has shown that their complexity can hinder adoption by SMEs, thereby creating a gap between the fast-moving physical layer and the more formal semantic layers. We argue that this gap should be bridged by a simplified approach with a low entry barrier that remains compatible with existing standards. Building on this argument, this paper introduces the core ideas of S2DM.

2. S2DM design principles

The need for a simplified approach, such as S2DM, is grounded on the four components of the Design Science Research (DSR) methodology [2]. Namely: problem, requirements, goal, and artifacts.

Problem The prevalence of disparate (vehicle) data models and application-centric architectures without controlled vocabularies [3] highlights a gap: while existing standards offer semantic rigor, their complexity hinders SME adoption and sustained contributions to domain vocabularies.

Requirements The identified problem reflects a non-trivial trade-off between usability for SMEs and the semantic rigor offered by established standards, which is explicitly captured in Table 1.[2]

Criteria	Requirement
Simplicity	Understandable and usable with a low entry barrier for a SME.
Technology agnosticism	Applicable in any environment and independent of specific platforms or vendors.
Modularity	Smaller independent parts can be modeled and combined into a coherent whole.
Scalability & Maintainability	Supports growth and maintenance as domains expand.
Metadata resource uniqueness	Clear explicit identification and versioning of elements to avoid confusion or ambiguity.
Multi hierarchies	Enhances find-ability by supporting multiple ways to classify and organize concepts.
Cross-domain references	Enables linking and reuse of data across different domains and application areas.
Possible operations	Specifies data structures along with their associated capabilities.
Community & Tools	Compatible with open tools and broadly adopted industry standards for wide support.

Table 1
Design criteria and their corresponding requirements that were considered for *S2DM*.

Goal To lower the entry barrier for SMEs to engage in semantic data modeling, without sacrificing the rigor needed for clarity, consistency, and reuse.

Artifacts At least these two artifacts are foreseen: a *data modeling guideline* that prescribes how to formalize domain data models, and a set of *tools* that assist in their creation, validation, and governance.

3. S2DM approach

Rather than a full technical specification, this section outlines the core concepts of S2DM as a proposed solution, intended to frame discussion and guide future development. At its core, S2DM builds on two established standards: the Schema Definition Language (SDL)[3] [4] and the Simple Knowledge

[2] Note that *Reasoning* is not seen as a high-priority feature (in the context of the VSS project) and was not included.
[3] https://graphql.org/learn/schema/

Organization System (SKOS)[4] [5]. SDL defines data structures and operations and is best known as the schema language of the Graph Query Language (GraphQL) ecosystem. In S2DM, the focus lies on the schema definition itself rather than the GraphQL API. As a complement, SKOS is used to support multiple classification perspectives as well as thesaurus-like features, including broader/narrower relations and lexical variants.

The main ideas behind S2DM are summarized in six parts, illustrated in Figure 1 and explained next. Importantly, SMEs would actively interact only with step (1), whereas step (3) would be partially enhanced by a semantic data modeler (mostly for data cataloging purposes). All other steps can be automated by tooling.

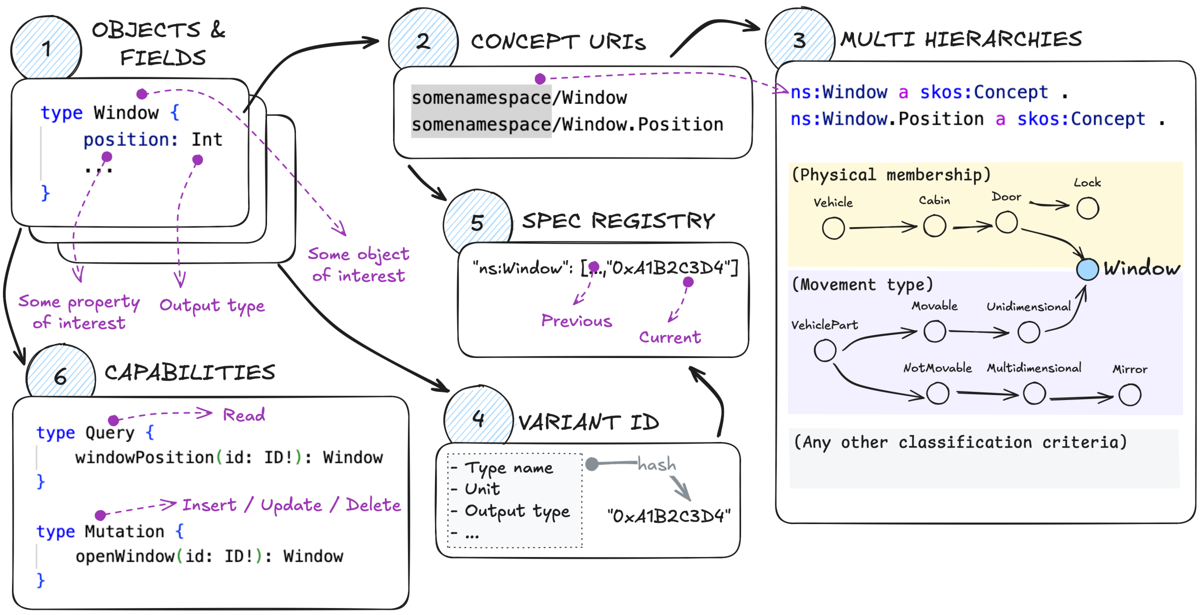

Figure 1: Overview of the S2DM approach. (1) Sets of objects and their fields are maintained by SMEs. (2) Concepts are assigned future-proof *URIs*. (3) Concept *URIs* are reused to build classification schemes. (4) Based on the metadata in the specification, a unique variant ID is created. (5) *URIs* and variant IDs are combined into a specification registry. (6) Possible operations on the data structures are specified.

(1) Objects & Fields A domain data model evolves around the objects of interest and their associated fields, expressed in SDL. For instance, the window position could be modeled as a *Window* object with a *position* field that captures its current state as an integer value.

(2) Concept URIs The elements of the schema (e.g., the *Window* object and the *position* field) are assigned Uniform Resource Identifiers (URIs)[5] that remain valid across versions. This persistent explicit identification also enables operation in the Resource Description Framework (RDF)[6] world.

(3) Multi-hierarchies Having URIs in place enables the use of SKOS to support thesaurus features, such as multiple classification schemes. For example, *Window* may be classified by its *physical membership* as *Vehicle→Cabin→Door→**Window***, or by its *movement type* as *VehiclePart→Movable→Unidimensional→**Window***.

(4) Variant IDs They refer to the unique combination of the metadata that realizes the concept of interest. When metadata changes in ways that may affect downstream use, variant IDs make those

[4]https://www.w3.org/2004/02/skos/
[5]https://datatracker.ietf.org/doc/html/rfc3986
[6]https://www.w3.org/TR/rdf11-concepts/

changes explicit and traceable. For instance, the *Window.position* first being defined as a *Float* and later as an *Integer*.

(5) Spec registry Concept URIs and variant IDs are persisted together in a registry, so that even as they evolve, their history can be tracked and earlier definitions remain accessible for governance purposes.

(6) Capabilities As part of SDL, one can specify the operations that are possible on a particular data structure. For example, reading the window position (i.e., *Query*), and opening a window (i.e., *Mutation*).

4. Related Work

There are multiple approaches to modeling vocabularies and data structures, each with distinct strengths and limitations. This overview is non-exhaustive and intended only to highlight positioning. Table 2 summarizes common approaches to achieve similar goals, which are briefly discussed next.

Approach	Strengths	Limitations (for SMEs)
VSPEC (i.e., VSS)	Simplicity, domain focus	Lacks cross-domain semantics and multiple classification schemes. Uses a proprietary language that requires considerable tooling effort.
OWL / SHACL / SKOS	Rich semantics, validation, classification support	Authoring complexity and steep learning curve.
JSON Schema / OpenAPI	Tooling support, interface clarity	Limited semantics and weak governance features.
GraphQL SDL	Flexible access, alignment with linked data	No built-in conventions for vocabulary governance.
S2DM (this work)	SME-first design, modular, lightweight semantics and governance via variant IDs	Requires export or mapping to richer semantic stacks.

Table 2
Comparison of approaches relevant to *S2DM* positioning.

Domain-specific languages. The so-called *VSPEC* language used in the VSS project [1] illustrates the value of simplicity and domain focus. However, it has limited expressiveness. For instance, it lacks support for cross-domain references and multiple classification schemes.

Semantic Web standards. The Web Ontology Language (OWL) [6] and the Shapes Constraint Language (SHACL) [7] provide expressive semantics and validation capabilities. However, their authoring complexity and steep learning curve limit adoption by non-specialists. This was experienced first-hand in COVESA, when an ontology extension for VSS was proposed [8, 9] but later discontinued [10]. In contrast, S2DM emphasizes SME usability (with SDL) while maintaining compatibility with others via URIs, exports, and mappings. The choice of SKOS [5] reflects only one way in which SDL models created by SMEs could be complemented when richer semantics are required.

Technology-agnostic schema frameworks. *OpenAPI* [11] and *JSON Schema* [12] provide machine-readable API contracts and validation vocabularies with strong tool ecosystems (e.g., code generation and schema validation). However, they focus on structural constraints and annotations rather than ontological semantics (e.g., class hierarchies, property semantics, or reasoning). In contrast, SDL [4] defines a typed schema and operations (i.e., queries and mutations) for data access independent of specific transports. Neither specification prescribes governance processes for domain vocabularies (e.g., global identifiers, versioning or deprecation policies, or cross-domain reference patterns), which are intentionally left to organizational practice or complementary tooling.

Position of S2DM. S2DM aims to combine the simplicity and modularity of domain-specific schemas (via SDL) with lightweight semantics and governance (via URIs and SKOS). It is not intended to compete with existing standards. Rather, it aims to embrace them in a way that empowers SMEs to develop models with selected best practices in semantic data modeling, but with low effort.

5. Conclusion and outlook

We argued that lowering the entry barrier to semantic data modeling is essential in organizations that maintain controlled vocabularies, where the main contributors are typically SMEs rather than professional data modelers. Our position is that a simplified approach, compatible with existing standards yet tailored for SME usability, can bridge the gap between practical modeling needs and formal semantics. To that end, we introduced the basic ideas of the S2DM approach, which leverages the established standards SDL for schema definition and SKOS for thesaurus-oriented features. Although initially motivated by the work in the COVESA automotive alliance, its design is envisioned to be domain-independent. S2DM is not intended to replace expressive semantic modeling languages, such as OWL or SHACL. Features beyond SDL, SKOS, or basic RDF vocabularies remain out of scope.

Looking ahead, we plan to validate and refine the ideas presented for S2DM with a public implementation.[7] A key near-term goal is to establish S2DM as the default modeling approach within COVESA and to enhance the semantics of VSS. Moreover, planned work includes developing tooling for release governance and continuous integration, incorporating standard vocabularies, and exploring semi-automated modeling with Large Language Models (LLMs). Future work will also explore strategies for migration and backward compatibility, for example by leveraging variant IDs in combination with version-aware data transformations, so that downstream systems can adapt gracefully to specification changes. To reduce duplication and foster interoperability across vocabularies, further efforts will include curating a set of predefined elements that SMEs can reuse (e.g., custom scalars, directives, and units).

[7]Most up-to-date implementation details and ongoing developments can be consulted online at: https://covesa.github.io/s2dm/

Declaration on Generative AI

During the preparation of this work, the authors used *GPT-4o* and *GPT-5* in order to: grammar and spelling check, paraphrase and reword, improve writing style, and abstract drafting. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] COVESA, Vehicle Signal Specification (VSS), 2025. URL: https://covesa.github.io/vehicle_signal_specification/.

[2] R. J. Wieringa, Design Science Methodology for Information Systems and Software Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. URL: http://link.springer.com/10.1007/978-3-662-43839-8. doi:10.1007/978-3-662-43839-8.

[3] D. Mccomb, Software Wasteland: How the Application-Centric Mindset is Hobbling our Enterprises, Technics Publications, Basking Ridge, NJ, 2018.

[4] GraphQL Contributors, GraphQL Specification, 2021. URL: https://spec.graphql.org.

[5] A. Miles, J. R. Pérez-Agüera, SKOS: Simple Knowledge Organisation for the Web, Cataloging & Classification Quarterly 43 (2007) 69–83. URL: http://www.tandfonline.com/doi/abs/10.1300/J104v43n03_04. doi:10.1300/J104v43n03_04.

[6] Web Ontology Language (OWL), 2012. URL: https://www.w3.org/TR/owl-overview/.

[7] Shapes Constraint Language (SHACL), 2017. URL: https://www.w3.org/TR/shacl/.

[8] B. Klotz, R. Troncy, D. Wilms, C. Bonnet, VSSo - A vehicle signal and attribute ontology, in: SSN 2018, 9th International Semantic Sensor Networks Workshop, 9 October 2018, Monterey, CA, USA, Monterey, ÉTATS-UNIS, 2018. URL: http://ceur-ws.org/Vol-2213/paper5.pdf.

[9] D. Wilms, D. Alvarez-Coello, A. Bekan, An Evolving Ontology for Vehicle Signals, in: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), IEEE, Helsinki, Finland, 2021, pp. 1–5. URL: https://ieeexplore.ieee.org/document/9448884/. doi:10.1109/VTC2021-Spring51267.2021.9448884.

[10] B. Klotz, R. Troncy, D. Wilms, VSSo: Vehicle Signal Specification Ontology, 2024. URL: https://www.w3.org/TR/vsso/, w3C Discontinued Draft.

[11] OpenAPI Specification, 2024. URL: https://spec.openapis.org/oas/latest.html.

[12] JSON Schema specification, 2020. URL: https://json-schema.org/specification.