

The CIDOC CRM Periodic Table: Documentation You Can Test to Support the Adoption of Scientific Research Ontologies for the Humanities

Remo Grillo^{1,2}

¹Alma Mater Studiorum — Università di Bologna, Bologna, Italy

²I Tatti — The Harvard University Center for Italian Renaissance Studies, Florence, Italy

Abstract

The CIDOC CRM family is not only a schema for museum collections. Its extensions formalise the apparatus through which research knowledge is produced and justified: CRMsci models scientific observation and measurement, CRMdig the provenance of digital processes, CRMinf inference and belief. Together they constitute a schema for research data, and the obstacle to their adoption is documentary rather than theoretical: four normative documents, four code conventions, and eighty-nine typed links crossing document boundaries. This paper presents the CIDOC CRM Periodic Table, a single-page browser rendering 131 classes and 480 property entries from CIDOC CRM 7.1.3, CRMsci 3.2, CRMdig 5.0 and CRMinf 1.2.1 as one navigable surface, with two inference facilities: a one-step triple builder admitting only the class–property combinations licensed by declared domains, ranges and the subclass closure, and a multi-step path finder enumerating property chains between two classes. Because all loaded models are merged into one code-keyed dictionary before the hierarchy closures are computed, both resolve chains that cross model boundaries; we document the addition of CRMsci and the three cross-model tests it made possible, and report preliminary uptake signals from 56 months of analytics, with the reading they support.

Keywords

CIDOC CRM, research data, ontology documentation, scientific observation, provenance, RDFS entailment

1. Introduction

A standard is adopted through its documentation, and a standard that grows by extension fragments its documentation along the same seams: what a modeller needs to know ends up distributed across the very modules that make the standard extensible. The CIDOC Conceptual Reference Model [1], standardised as ISO 21127, is accompanied by a family of extensions that, taken together, describe how empirical claims come to be: CRMsci [2] defines itself as “a global schema for integrating metadata about scientific observation, measurements and processed data in descriptive and empirical sciences such as life sciences, geology, geography, archaeology, cultural heritage conservation and others in research IT environments and research data libraries”; CRMdig [3] models the provenance of digital processes; CRMinf [4] models inference, argumentation and belief. A researcher who wants to say what was observed, by which instrument, through which digital transformation, and on what grounds the resulting statement is believed, already has a normative vocabulary for all four moves.

What that researcher does not have is a workable way to read it. Nine years ago, in a peer-reviewed venue, Guillem and Bruseker argued that CIDOC CRM adoption in concrete practice is limited because ontologies are “considered too complicated and abstract for application in real life scenarios” [5]. Since then the specification has grown rather than shrunk. Covering the four models discussed here means reading four separate documents that declare 131 classes and 480 property entries, use four disjoint code conventions (E/P, S/O, D/L, I/J), and connect to each other through 89 subClassOf and

Sci-K 2026: 6th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment, co-located with ISWC 2026, October 25–26, 2026, Bari, Italy

✉ remo.grillo@unibo.it (R. Grillo)

🌐 https://remogrillo.github.io/cidoc-crm_periodic_table/ (R. Grillo)

🆔 0000-0003-3585-0198 (R. Grillo)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

subPropertyOf declarations that point out of the document one is currently reading. The barrier is documentary, and it grows with every extension the family gains.

This paper describes the CIDOC CRM Periodic Table [6], a client-side tool¹ that addresses this barrier, and reports on the recent addition of CRMsci. Its contributions are:

- a single-page interface presenting CIDOC CRM 7.1.3 with CRMsci 3.2, CRMdig 5.0 and CRMInf 1.2.1 as one surface, with per-model filtering and shareable deep links (Section 4);
- two inference facilities — a constrained one-step triple builder and a bounded multi-step path finder — whose coverage and non-coverage we state exactly, and a demonstration that the family’s cross-model links are resolved structurally rather than case by case, through three tests enabled by the CRMsci import (Section 5);
- preliminary uptake signals over 56 months, with the caveats a reader needs in order not to over-read them (Section 6).

2. Background: the CRM family as research-data infrastructure

Read individually, each extension looks like a specialist concern. Read together, they cover the lifecycle of an empirical claim. Take a mundane research workflow: a sample is taken, it is measured with an instrument, the instrument writes a file, and someone draws a conclusion. Every step is already declared, across three of the four models: S2 Sample Taking -O5 removed-> S13 Sample; S21 Measurement -O24 measured-> S13, admissible because S13 descends from S15 Observable Entity, the declared range, with S21 -O39 observed dimension-> E54 Dimension for the value; D11 Digital Measurement Event $\sqsubseteq$ S21, CRMdig subclassing CRMsci, with D11 -L20 has created-> D9 Data Object; and S5 Inference Making $\sqsubseteq$ I1 Argumentation, CRMsci subclassing CRMInf, with I1 -J2 concluded that-> I2 Belief. This is a schema for research data, not for catalogue records, which is what Beretta’s SDHSS ecosystem argues from the side of the social sciences and humanities [7]. Its provenance half overlaps with what PROV-O [8] models generically; CRMdig differs in committing to the same event-centric ontology as the observation and inference halves — which is precisely why the four documents have to be read together.

The decisive point for tooling is that these models are *interlocked*, *not parallel*: they are not four self-contained documents, each readable on its own, but documents that declare entities whose superclass or superproperty is defined in another one. S5 Inference Making is declared a subclass of CRMInf’s I1 Argumentation; S27 Observation likewise; CRMdig’s D11 Digital Measurement Event is a subclass of CRMsci’s S21 Measurement, and L1 digitized a subproperty of CRMsci’s O24 measured. Counting from the machine-readable releases, the four models declare 89 links whose target lives in a different document (Appendix A): 31 from CRMdig, 30 from CRMsci and 28 from CRMInf. Ten of CRMsci’s point at CRMInf rather than at base CRM, and three of CRMdig’s point at CRMsci.

The practical consequence is that reading one specification is never enough, and the workflow above shows it concretely. A modeller who opens CRMdig alone sees that D11 creates a D9, and misses that D11 is a measurement and therefore already carries O24 measured and O39 observed dimension; a modeller who opens CRMsci alone cannot see that the properties of I1 Argumentation apply to S5. The official HTML declarations, published per model and per version, faithfully mirror the documents — and therefore mirror the fragmentation too. This is the gap the interface targets, and it is a documentation problem of the kind the community meets whenever scholarly knowledge is represented in a modular standard: comparable infrastructures for scholarly knowledge, such as the Open Research Knowledge Graph [9], invest heavily in making their model legible to contributors for the same reason.

¹Live interface: https://remogrillo.github.io/cidoc-crm_periodic_table/; source code: https://github.com/RemoGrillo/cidoc-crm_periodic_table; archived release: <https://doi.org/10.5281/zenodo.21821748> (concept DOI). The interface is a static page: serving it locally needs no build step and no dependencies, and the repository README gives the commands for that and for regenerating the data files from the official RDFS sources.

Table 1

What each genre offers for the four capabilities this paper claims. Compiled from each resource’s own public documentation, consulted 4 August 2026: “–” means the capability is not among the documented features, not that it is unattainable. *One surface* = every loaded model visible in a single view; *cross-document closure* = a subclass chain whose target is declared in another model’s document is resolved; *constrained triple* = admissible classes and properties are derived from declared domains, ranges and the subclass closure; *chain search* = property paths between two given classes are enumerated. The ontology editor is listed as an upper bound on raw capability rather than as a documentation resource: what separates it from the last row is purpose, not expressive power.

	One surface	Cross-document closure	Constrained triple	Chain search
Official HTML declarations	per model	–	–	–
Generated docs (WIDOCO)	per model	–	–	–
Graph visualisers (WebVOWL)	per model	–	–	–
Ontology management (OntoME)	–	✓	–	–
Ontology editor + reasoner	–	✓	✓	partial
Periodic Table	✓	✓	✓	✓

3. Related work

Three genres of resource already address parts of this problem, and it is worth being precise about which part (Table 1).

Specification renderings are faithful to their input and inherit its boundaries: the CRM-SIG’s machine-generated declarations give one HTML page per model per version, and generated documentation such as WIDOCO [10] or visualisers such as WebVOWL [11] do the same for arbitrary vocabularies. In all of them a reference into another model stays a string.

Ontology management. OntoME is the closest neighbour — a collaborative environment handling multiple namespaces, profiles and versioning, which lets users align their own classes to CIDOC CRM and compatible models [12] and does resolve the family as one graph. Its centre of gravity is authoring and curation, one page per entity; ours is reading a whole model at once and testing what it licenses.

General-purpose tooling. One can load all four RDFS files into an editor such as Protégé [13] with an OWL reasoner behind it, and in terms of raw capability that suffices. The difference is one of purpose. An editor supports *building and verifying* an ontology: its unit of work is the axiom, its question whether a model is consistent. The task here is downstream and far more common — *applying* a standardised ontology one did not design, translating “this sample was measured, and here is the file the instrument wrote” into its codes. An editor supplies the vocabulary but no help in choosing within it, and asking “which properties may hold between a sample and a data object” costs a reasoning environment and a formalised query; here it costs a URL. Documentation and modelling support are usually separate artefacts; for a modular standard they should not be.

Three neighbours sit outside the table. Mountantonakis et al. catalogue which *datasets* use which classes and properties [14]: instance-level evidence, where we offer schema-level navigation. Linked Art narrows the model instead, publishing a profile over a subset of CIDOC CRM 7.1.3 [15] — a subset helps the modeller who accepts its scope, navigation helps the one who cannot. Closest in spirit at this workshop, Tietz et al. treat cultural-heritage material as research data and build discovery services over it [16]: the same premise, applied to the data rather than the schema. Tools we could not verify are deliberately not cited.

4. Design

The interface renders every class of every enabled model as a cell on a single scrolling page, and every property as a line in a companion view (Figure 2 schematises that surface and Figure 3 shows one block

of it as rendered). Four decisions do most of the work.

Placement follows the code. A cell's code prefix determines which model it belongs to and whether it is a class or a property; the four code spaces are disjoint, so a code denotes exactly one entry across the loaded family. Cells then flow inline and wrap, so the grid reflows to the width available instead of being pinned to fixed columns. Each model occupies one contiguous block, introduced by a full-width header in the model's own hue, so provenance is legible without a legend.

Colour is inherited through the hierarchy. Seven anchor classes — E4 Temporal Entity, E18 Physical Thing, E39 Actor, E28 Conceptual Object, E41 Appellation, E53 Place, E55 Type — define seven colour families, and every class takes the colour of the anchor it descends from through the `subClassOf` closure. Nothing is hard-coded per class, so adding a model costs no colour work: 20 of CRMsci's 28 classes were coloured as intended the moment they were loaded. Grouping CRM's classes into colour-coded families is not our idea: we inherit it, along with the association between family and hue, from the CIDOC CRM Game [5, 17]. The anchors, the values, the derivation of colour by inheritance, and the second colour axis marking which model an entry comes from are ours; no graphic material from the game was reused.

Inheritance depth is legible on the cell. Each cell carries the codes of its immediate superclasses and a marker that shrinks with the length of its longest inheritance path, separating E1 from a class eight steps below it.

The URL carries the state. The query string of the browser URL — `?ont=crm, crmsci, ?code=S4, ?path=S13, D9` — is read on load and rewritten on interaction, so any class, filter combination or path query is a citable link, and a deep link silently enables the model that owns its entry. These are view parameters and not identifiers: an entity's URI remains the one its ontology declares, carried on the cell and on the copy action, though it deserves to be shown in the detail panel too.

Accommodating CRMsci exercised exactly these mechanisms. It uses S for classes and O for properties, a convention shared with neither base CRM nor the other extensions, so it needed a registry entry, two code-prefix declarations, a data file and a provenance hue — and nothing else. The hue was chosen by measurement rather than taste: indigo (#3A3FA0) sits 37 degrees of hue from its nearest neighbour among the two model hues and the seven family colours, at 8.8:1 contrast on white.

5. Inference support

A documentation browser is easy to build; letting a modeller assemble a triple and see what the specification licenses is the distinctive part, and the place where over-claiming is most tempting, so we state the boundaries plainly.

One step. The user picks a domain class, a property and a range class in any order, and the table dims every entry that the choice has excluded. The rule is subsumption over declared domains and ranges: a class is admissible in a slot if it is the declared domain or range of the property, or a subclass of it through the transitive `subClassOf` closure; a property is admissible if the classes already chosen fall within its declared domain and range in the same sense. The completed triple is read back as an English sentence and as parsable Turtle. The Turtle is built at instance level, with blank nodes and only the `@prefix` declarations the triple actually uses, because CRM properties hold between instances and a triple asserted between classes would be meaningless.

Multiple steps. The path finder takes two classes and enumerates property chains between them, using a bounded bidirectional breadth-first search over the domain/range graph, with subsumption applied at every hop: a property is available on a class if the class or any of its superclasses is the declared domain. Results are grouped by chain length and ranked by the specificity of the properties involved. The bounds are what make this tractable: each side expands at most half the requested depth, every level is beam-limited to its 600 most specific states and the answer set to 100, so the work per query grows with those caps and the branching of the domain/range graph rather than exponentially with depth, and any truncation is reported on screen. The search runs synchronously in the browser over the whole loaded family; it is the caps, not the size of the family, that keep it interactive. An

optional *loose* mode admits hops that are valid only if the intermediate instance belongs to a more specific class than the chain guarantees — the downcasts that are common in real modelling — and such chains are dashed and labelled in the output, never silently mixed with sound ones (Figure 1).

Cross-model chains are structural. Every loaded model is merged into one dictionary keyed by bare code, and the subclass and subproperty closures are computed after the merge, over the union. Nothing in the search knows about namespaces; a chain crosses a model boundary the same way it crosses any other edge. The CRMsci import let us test this on the three cases that exist in the data; each is a deep link, so each is re-enterable (`?ont=crm, crminf, crmdig, crmsci&path=`). First, CRMsci to base CRM: `S4, E53` yields 11 sound chains within two steps, five of them touching three models. Second, CRMsci to CRMinf: `S5, I2` yields exactly one sound chain, `S5 -J2-> I2`, and it exists only because `S5 Inference Making` inherits the domain of a CRMinf property through `S5  $\sqsubseteq$  I1 Argumentation`. This is the case a reader of CRMsci alone cannot see, and it is the example we would put in front of a sceptic. Third, CRMdig to CRMsci, `D11, S13`: before the import, `D11`'s superclass `S21` was an inert label and no chain could pass through it; after it, `D11 -O24-> S13` is found and flagged loose, while `S13, D9` returns `S13 -O5i-> S2 -P70i-> D9`, which spans CRMsci, base CRM and CRMdig soundly.

What is verified, and what is not. Two things were checked and are reproducible from the script distributed with this paper: the loaded entity counts against the official class and property declarations (28 classes and 39 forward properties for CRMsci 3.2, plus 38 materialised inverses), and the transitive subclass closure across model boundaries. The three path results reported above were obtained from the deployed interface and are stated here with the queries that produce them, so a reader can re-enter each one — they are documented, not scripted. We did *not* verify the implementation systematically against the full set of entailments the specifications declare, and we do not claim to. Concretely, the following are absent by design and should not be inferred from the word “inference”: `subPropertyOf` entailment is not applied, so asserting a subproperty does not surface its superproperty — a real limitation here, given that CRMsci declares 40 subproperty links, 10 of them into base CRM; inverse properties are present because the RDFS materialises them, not because an `owl:inverseOf` rule is applied; property transitivity, cardinality and the CRM quantifier statements are not modelled; shortcut properties are not held equivalent to their full paths; and there is no consistency checking, so the tool prevents an inadmissible triple rather than diagnosing one. The multi-step search is incomplete by construction — depth is capped at four steps, each frontier is beam-limited, and at most 100 results are returned — and the interface reports truncation on screen. A “no path found” answer is therefore not a proof of absence, and we would rather say so than let a reader assume otherwise.

In the vocabulary of research assessment, the two facilities make a representation's *coverage* countable (Appendix A) and its *interpretability* testable: what the schema licenses becomes something one can question instead of reconstruct. That is not a quality metric, and nothing here substitutes for one.

6. Preliminary uptake signals

The tool carries a Google Analytics 4 property whose record begins on 1 December 2021; the repository is older. Over the 56 months to 4 August 2026 the property recorded 2 552 active users and 94 078 events, with a mean engagement time of 540 seconds per active user, and 60 517 page views on its main title. Users were attributed to 654 distinct cities, 111 of them with five or more users, and the author's own institution accounts for 82 sessions out of 9 222: use is overwhelmingly external.

Two patterns in the referrers are more informative than the totals (Appendix B lists the hosts). Teaching circulation is visible: 302 sessions arrive from course environments, including Google Classroom and the Moodle instances of five universities. Professional circulation is visible too: 224 sessions arrive from links opened inside working documents and meetings — Word documents served through `officeapps.live.com`, Teams chats, Webex and Zoom. Community referral comes chiefly from GitHub (302 sessions) and `linked.art` (97 sessions). Separately, three page titles that appear in the property but in no version of this repository account for 2 587 views by 150 users; the titles announce further extensions, so the most economical reading is an independently deployed derivative that inher-

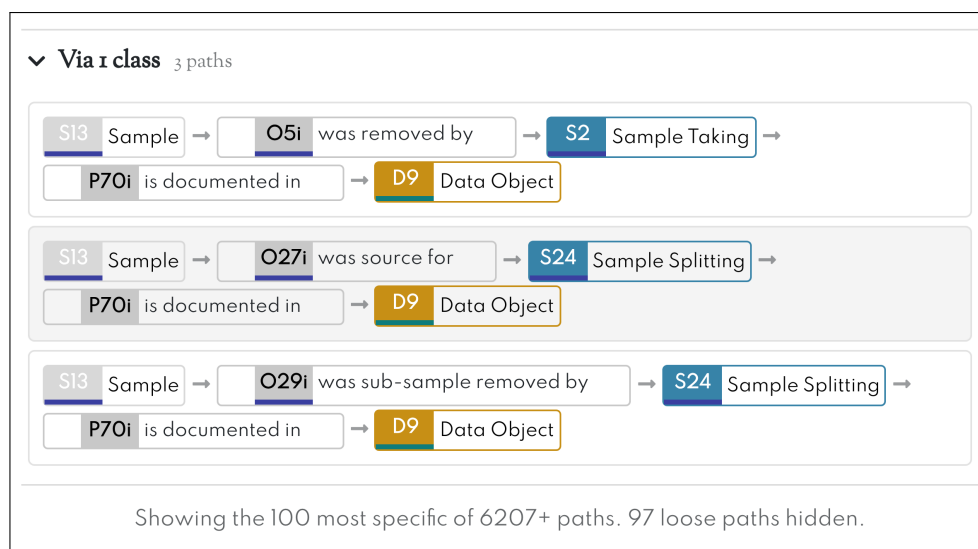


Figure 1: Multi-step output for S13 Sample to D9 Data Object at three steps, sound chains only. All three cross three models: CRMsci classes (indigo stripe) reach a CRMdig class (teal stripe) through the base-CRM property P70i, resolved by inherited domains and not by any namespace-specific rule. The footer is reproduced from the interface, truncation notice included.

ited the analytics tag. We cannot verify that from the outside, we exclude those views from every other figure in this section, and we mention them only because a reused tool is worth more than a consulted one.

These data support statements about reach and about the kinds of setting in which the tool is opened. They support no statement whatsoever about learning, usability or didactic effectiveness: there has been no user study, and none is claimed. Four caveats bound even the reach claims. First, only GA4's default bot exclusion is in force, so sophisticated headless crawlers may be counted. Second, ad blockers systematically depress these counts, and this audience is unusually likely to run them. Third, and most easily misread: because the interface routes by code, inspecting a class generates a page view, so the view count is a proxy for depth of exploration and not for distinct visits — a reviewer who opens the site will confirm this in seconds. Fourth, the property was never filtered for the author's own development traffic; 31 sessions from localhost and self-referrals are still in the data.

7. Limitations and roadmap

The tool has never been evaluated with users, and the analytics above are not a substitute. Coverage of the family is partial: CRMgeo, CRMarchaeo and CRMba are not loaded, and references into them remain inert labels. The inference facilities cover the fragment described in Section 5 and no more. There is no automated test suite. Two interface decisions belong here as well. The layout assumes a desktop viewport: the premise of the tool is a whole model in view at once, which is desk work, and it does not lay out for small screens. And the colour families, derived and documented in Section 4, have no legend in the interface itself — the per-model chips double as one for provenance, the seven family hues have no equivalent.

How far this generalises. Within the CRM family the answer is settled by construction: a further model costs a registry entry, its code prefixes, a data file and a hue, as CRMsci did; models with two-letter prefixes — CRMgeo's SP, CRMarchaeo's AP — additionally need the code pattern widened, a contained change. Beyond the family the boundary is not in the code but in an assumption about the domain. The table rests on a convention that is universal in the CRM family and general nowhere: that every class and property carries a short mnemonic code of one letter and a number. That code decides the cell, the owning model and the class/property split — and it is declared nowhere in the RDF, there being no

property that states it, so it is recovered heuristically from the local name of the URI. A vocabulary without such codes could still have the cross-document closure, the constrained triple builder and the path finder, which need only RDFS; it could not have the periodic table. We would therefore claim generalisation to modular, code-bearing standards, and no further.

The roadmap follows from that list, in order of what would change most: a teaching study in a graduate course, where the tool is already being used informally; systematic verification of the implemented entailments against the normative declarations, beginning with `subPropertyOf`; the remaining compatible extensions, for which the CRMsci import is now the template; and two small debts to the reader, a legend for the colour families and the entity URI shown in the detail panel.

8. Conclusion

The CIDOC CRM family already offers a normative vocabulary for observation, digital provenance and inference — for the apparatus by which research data is produced and justified. What it does not offer is a way to read four interlocking documents at once and check what they license together, and that gap is not a detail of presentation: it is where the cost of adopting the standard is actually paid.

The Periodic Table takes the modest position that this is a tooling problem rather than a modelling one. Its design result is small enough to state in a sentence — merge every loaded model into one dictionary keyed by bare code, then compute the hierarchy closures over the union — and its consequence is that cross-model inspection stops being a special case and becomes the default. Adding CRMsci exercised precisely that: a registry entry, two code prefixes, a data file and a colour, after which chains cut at the document boundary resolved on their own, and the same four steps — with the code pattern widened where a model's prefixes run to two letters — are what it would take to bring in CRMgeo, CRMarchaeo or CRMba.

Two limits travel with the claim. The inference facilities implement a fragment — subclass transitivity and domain/range applicability — and we say which entailments are absent, because a tool that answers “what may I say about this?” is trusted, and trust must not be borrowed against. And it has never been evaluated with users: 56 months of traffic establish that people reach it, not that it helps them. What we would draw from this is more general than CIDOC CRM: modular standards distribute their meaning across documents and leave the reader to recompose it; where that recomposition is mechanical, the documentation can perform it.

Declaration on Generative AI

During the preparation of this work, the author used Claude (Anthropic) in order to: Grammar and spelling check. Further, the author used Claude (Anthropic) during the development of the CIDOC CRM Periodic Table in order to: Generate code, confined to bug fixing in an existing codebase. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of this publication.

No generative AI system was used to generate or analyse research data, or to draft the scientific content of this manuscript. The interface described here was written by hand and its development began before such assistants were available; AI assistance came late and was limited to the bug fixing stated above. All bibliographic references were verified against primary sources, and every quantitative statement was recomputed from the source data.

References

- [1] C. Bekiari, G. Bruseker, E. Canning, M. Doerr, P. Michon, C.-E. Ore, S. Stead, A. Velios, Volume A: Definition of the CIDOC Conceptual Reference Model, Version 7.1.3, Standard specification, CIDOC CRM Special Interest Group, 2024. URL: <https://cidoc-crm.org/versions-of-the-cidoc-crm>, editors as named on the title page. Standardised as ISO 21127.

- [2] M. Doerr, G. Hiebel, A. Kritsotaki, Y. Rousakis, W. Schmidle, M. Theodoridou, A. Velios, Definition of the CRMsci: An Extension of CIDOC CRM to support scientific observation, Version 3.2, Standard specification, CIDOC CRM Special Interest Group, 2026. URL: <https://cidoc-crm.org/crmsci/ModelVersion/crmsci-v3.2>, contributors as named on the title page (“and others”); maintained by FORTH.
- [3] M. Doerr, A. Kritsotaki, S. Stead, M. Theodoridou, Definition of the CRMdig v 5.0: Model for provenance metadata, Version 5.0, Standard specification, CIDOC CRM Special Interest Group, 2025. URL: <https://cidoc-crm.org/crmdig/ModelVersion/crmdig-v5.0>.
- [4] M. Doerr, P. Fafalios, A. Kritsotaki, C.-E. Ore, S. Stead, Definition of the CRMinf: An Extension of CIDOC CRM to support argumentation, Version 1.2.1, Standard specification, CIDOC CRM Special Interest Group, 2026. URL: https://cidoc-crm.org/crminf/fm_releases, contributors as named on the title page (“and others”); maintained by Paveprime Ltd and FORTH.
- [5] A. Guillem, G. Bruseker, The CIDOC CRM game: A serious game approach to ontology learning, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W5* (2017) 317–323. doi:10.5194/isprs-archives-XLII-2-W5-317-2017.
- [6] R. Grillo, Cidoc-CRM periodic table, version 1.0.0, Software, archived on Zenodo, 2026. URL: <https://doi.org/10.5281/zenodo.21821748>. doi:10.5281/zenodo.21821748, concept DOI, resolving to the latest release; the version described here is 1.0.0, DOI 10.5281/zenodo.21822157. Source: https://github.com/RemoGrillo/cidoc-crm_periodic_table.
- [7] F. Beretta, Semantic data for humanities and social sciences (SDHSS): an ecosystem of CIDOC CRM extensions for research data production and reuse, arXiv preprint arXiv:2402.07531, 2024. URL: <https://arxiv.org/abs/2402.07531>.
- [8] T. Lebo, S. Sahoo, D. McGuinness, PROV-O: The PROV Ontology, W3C Recommendation, W3C, 2013. URL: <https://www.w3.org/TR/prov-o/>, editors as named on the specification; 30 April 2013.
- [9] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D’Souza, G. Kismihók, M. Stocker, S. Auer, Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge, in: *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP ’19)*, ACM, 2019, pp. 243–246. doi:10.1145/3360901.3364435.
- [10] D. Garijo, WIDOCO: A wizard for documenting ontologies, in: *The Semantic Web – ISWC 2017, Lecture Notes in Computer Science*, Springer, 2017, pp. 94–102. doi:10.1007/978-3-319-68204-4_9.
- [11] S. Lohmann, V. Link, E. Marbach, S. Negru, WebVOWL: Web-based visualization of ontologies, in: *Knowledge Engineering and Knowledge Management, Lecture Notes in Computer Science*, Springer, 2015, pp. 154–158. doi:10.1007/978-3-319-17966-7_21.
- [12] F. Beretta, A challenge for historical research: Making data FAIR using a collaborative ontology management environment (OntoME), *Semantic Web 12* (2021) 279–294. doi:10.3233/SW-200416.
- [13] M. A. Musen, The Protégé project: a look back and a look forward, *AI Matters 1* (2015) 4–12. doi:10.1145/2757001.2757003.
- [14] M. Mountantonakis, I. Theocharakis, Y. Tzitzikas, Why we need ontology-specific data portals: A case study for CIDOC-CRM, in: *Proceedings of the International Workshop on Semantic Web and Ontology Design for Cultural Heritage (SWODCH 2023)*, volume 3540 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: <https://ceur-ws.org/Vol-3540/paper9.pdf>.
- [15] Linked Art Editorial Board, Linked Art Data Model, version 1.0.0, Community application profile of CIDOC-CRM 7.1.3, 2024. URL: <https://linked.art/model/>, version and governance as stated on the model documentation site.
- [16] T. Tietz, E. Posthumus, L. Söhn, J. J. Steller, O. Bruns, J. Waitelonis, T. Schrade, H. Sack, Knowledge representation and discovery for cultural heritage research data with CTO and SHMARQL, in: *Proceedings of the 5th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment (Sci-K 2025)*, volume 4065 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 117–133. URL: <https://ceur-ws.org/Vol-4065/>.
- [17] G. Bruseker, A. Guillem, O. Marlet, F.-X. Talgorn, The CRM game: Digital edition, Presentation to the 51st CIDOC CRM and 44th FRBR CRM SIG meeting, 2021. URL: <https://cidoc-crm.org/sites/>

default/files/CIDOC-Game.pdf, development financed and supported by the MASA consortium, as stated in the slides.

A. Full coverage table

Table 2 reports the coverage figures quoted in the body split by property direction, together with the cross-model link counts and the alternative CIDOC CRM 6.2.1 base, which the tool also offers and for which no extension has a compatible machine-readable release. All numbers are produced by `count_coverage.py` from the JSON files the interface loads at runtime; those files are generated from the official RDFS releases by the converter in the repository, so the chain from specification to figure is reproducible.

Table 2

Coverage and cross-model links, counted from the machine-readable releases. Forward properties exclude the inverse entries materialised in the RDFS. CRMsci 3.2 declares 39 forward properties and 38 inverses because O30 determined position has a literal range and therefore no inverse.

Model	Ver.	Cls.	Fwd.	Inv.	Links out of the document
CIDOC CRM	7.1.3	75	164	144	—
CRMsci	3.2	28	39	38	10 + 10 to CRM; 4 + 6 to CRMinf
CRMdig	5.0	14	22	21	8 + 20 to CRM; 1 + 2 to CRMsci
CRMinf	1.2.1	14	27	25	4 + 24 to CRM
Total		131	252	228	89
CIDOC CRM	6.2.1	84	153	133	alternative base, no extensions

Link counts are given as *subclass + subproperty*. The 89 cross-document links are what makes the merged view more than a convenience: each one is a fact about the family that no single specification document states in full.

CIDOC CRM 6.2.1 declares more classes than 7.1.3 (84 against 75) because the 7.x revisions deprecated and merged several classes of the 6.x series; the figure is not a counting error, and the tool offers 6.2.1 only as an alternative base, with no extension attached.

B. Referring hosts

Sessions by referring host, from the same export, grouped as in Section 6. Course environments: classroom.google.com 158, moodle.lmu.de 98, e.fhs.upr.si 15, online.uni-marburg.de and ddk-data.online.uni-marburg.de 15, virtuale.unibo.it 8, moodle.univie.ac.at 6, confluence.uni-muenster.de 1, redmine.ihpan.edu.pl 1. Documents, chat and meetings: statics.teams.cdn.office.net 118, the four regional *-word-edit.officeapps.live.com hosts 50 in total, mail.google.com 15, unito.webex.com 13, docs.google.com 9, zimbra.univ-lille.fr 8, zimbra.inha.fr 5, web.skype.com 3, st1.zoom.us 2, teams.microsoft.com 1. Community: github.com 302, linked.art 97, t.co 8, intarch.ac.uk 5, researchgate.net 1. Search engines 3 464, of which google 3 352. AI assistants: chatgpt.com 22, copilot.microsoft.com 3. Author's institution: dh.itatti.harvard.edu 82. Development traffic, unfiltered: remogrillo.github.io 22, localhost:8000 4, 127.0.0.1:5500 3, localhost:5500 1. Direct, with no referrer, 4 676. Total 9 222 sessions from 44 distinct sources.

C. Additional figures



Figure 2: Layout logic. Classes of every enabled model share one scrolling surface: base CIDOC CRM first, then one contiguous block per extension, each introduced by a full-width header in the model's provenance hue (violet CRMMinf, teal CRMdig, indigo CRMsci). Cell fill encodes the inherited class family, the left stripe encodes the model, the small marker encodes inheritance depth, and grey marks a class that descends from no family anchor. Toggling a model hides its cells without unloading it, so hierarchy closures stay intact. The inspector on the right flips to reveal the two inference panels of Section 5.

CRMsci 3.2					
E7 S1 Matter Removal S2	S1 S2 Sample Taking S3,S24	S21,S2 S3 Measurement by Sampling S21,S2	E13,S27 S4 Single Observation S4	II S5 Inference Making S6,S7,S8	S5 S6 Data Evaluation S5
S5 S7 Simulation or Prediction S5	S5 S8 Categorical Hypothesis Building S5	E55 S9 Property Type E55	E70,S15 S10 Material Substantial S11,S14	S10 S11 Amount of Matter S12,S13	S11,S14 S12 Amount of Fluid S11,S14
S11 S13 Sample S11	S10 S14 Fluid Body S12	E1 S15 Observable Entity S10	E63,S18 S17 Physical Genesis S17	E5 S18 Alteration S17	S27 S19 Encounter Event S27
E26,E53 S20 Rigid Physical Feature S22	S27 S21 Measurement D11,S3	S20 S22 Segment of Matter S20	S27 S23 Position Determination S27	S2 S24 Sample Splitting S2	E54 S25 Relative Dimension S26
S25 S26 Angle S25	II S27 Observation S4,S19,S21,S23	III S28 Observable Situation S29	II7,S28 S29 Observable Proposition S29		

Figure 3: The CRMsci block as it appears in the table after the import: 28 classes under a header in the model's hue, each cell carrying an indigo provenance stripe, the codes of its immediate superclasses, a marker for inheritance depth, and the fill of the class family it inherits. Blue cells descend from E4 Temporal Entity, green from E53 Place, ochre from E28 Conceptual Object, orange from E55 Type; grey marks the classes that descend from no anchor. Note the cross-model superclass codes: I1 above S5 and S27, I11 above S28. Cells are shown at full opacity; in use, the interface dims those unrelated to the current selection.



Figure 4: One-step output for S4 Single Observation, O8 observed, S15 Observable Entity: the natural-language read-back and the instance-level Turtle with the prefixes the triple actually uses. Only 6 of the 131 loaded classes are admissible in the range slot for O8 — S15 and its five subclasses — and the table dims the rest while the choice is open.