Discovering Digital Art Collections using Link-Traversal–based Query Processing

Martijn Bogaert
Ghent University
Ghent, Belgium
martijn.bogaert@ugent.be

***Abstract*—This master's thesis explores the exploration of digital art collections through Link-Traversal-based Querying, with a focus on the *Collections of Ghent*. The Comunica platform plays a key role in harnessing valuable data in RDF format, although some links pose challenges with RDF compatibility. Two web application ideas are proposed to assist both non-technical users and professionals in exploring the CoGhent collection. The ultimate goal is to associate discovered data with IIIF Manifests for visualization, thus increasing the accessibility of art collections.**

***Keywords*—Linked Data, Link Traversal, LTQP, CoGhent, IIIF**

Introduction

Digital art collections embody human creativity and cultural development. Through technological advancements, these collections have been digitized, rendering them globally accessible and open to profound exploration. However, navigating and interrogating this data presents challenges, particularly for non-technical professionals and art enthusiasts. These limitations impede their ability to gain insights and become fully immersed in the realm of digital art.

The cultural data of the Collections of Ghent (CoGhent) are published following the principles of Linked Data, firmly anchoring them within the semantic web. Yet, to fully harness the potential of these extensive data, Link-Traversal–based Query Processing (LTQP) is required. LTQP, in essence, empowers users to transcend the boundaries of the dataset, unveiling layers of knowledge and connections that would otherwise remain concealed.

The research dissects the *exploration* of CoGhent data into three fundamental components: formulating queries, executing queries using link traversal — the focal point of the study — and processing query results, notably their visualization and storage. This dissected approach lays the groundwork for a more profound exploration of the CoGhent data and potentially digital art collections at large.

I. Related Work

A. Collections of Ghent

This research primarily focuses on the data of the *Collections of Ghent* (CoGhent), or *Collectie van de Gentenaar* (CoGent) in Dutch. CoGhent is a collaboration between the city of Ghent, Design Museum Gent, Digipolis, and other local organizations. Together, their objective is to gather and digitize the city's cultural heritage into a centralized collection, encouraging the residents of Ghent to contribute their own heritage stories and objects. Although the CoGhent partnership concluded in June 2023, the infrastructure remains intact. [1] [2]

The data from the participating cultural institutions, namely Design Museum Gent (DMG), Huis van Alijn (HVA), Industriemuseum, STAM, and Archief Gent, are managed using Linked Data Event Streams (LDES). LDESs are collections of immutable objects represented by RDF triples. The immutability of these objects implies that once an object is added, it remains unchanged. New versions of objects are introduced instead of updating existing ones. [3] [4]

The LDESs of CoGhent, in particular, encompass *Human-Made Objects*, or *Mensgemaakte Objecten* in Dutch. These objects represent both tangible and intangible items created or influenced by humans, ranging from artworks and books to traditions and crafts. The *Open Standaarden voor Linkende Organisaties* (OSLO) initiative plays a pivotal role in standardizing these Human-Made Objects. Furthermore, these objects are fully aligned with international standards to ensure semantic interoperability within the domain of cultural heritage. [5] [6]

B. International Image Interoperability Framework

Each Human-Made Object within CoGhent's collections contains, alongside its descriptive data, a link to a IIIF Manifest. These manifests are structured RDF resources that aggregate specific information about an object, ranging from details such as dimensions and notes to copyright information. The International Image Interoperability Framework (IIIF) defines, through its Presentation and Image APIs, the guidelines for constructing these manifests. In the case of CoGhent manifests, the structure is straightforward: each manifest comprises a single sequence, which in turn contains a single canvas, which subsequently includes a single annotation with the image link and metadata. [7] [8] [3]

In addition to data storage, IIIF Manifests are particularly advantageous for visualizing cultural data. Multiple IIIF Viewers exist that facilitate this process. For a given manifest, these viewers offer a standardized representation of the data it contains. [9]

C. Link-Traversal-based Query Processing

The fact that the CoGhent collections are part of the Linked Data web implies that they can potentially generate far more knowledge than when querying the CoGhent data in isolation. However, attempting to access this *external* data with a single SPARQL query can only be accomplished if the executing query engine can *jump* from resource to resource. Link Traversal-based Query Processing (LTQP) makes this practically feasible by dynamically following links between documents. [10]

Nevertheless, without imposing constraints on the links to be followed, LTQP becomes impractical. Therefore, O. Hartig introduced three *reachability criteria* [11]:

- *cAll* follows all links without restriction.
- *cNone* follows no links at all.
- *cMatch* follows only links that are part of quads matching a quad pattern in the query.

Thanks to its modularity and adaptability, the above-mentioned capacities and others can be endowed to a Comunica engine, thus making LTQP feasible in practice. [12] [13]

II. CoGhent Data and Link Traversal

A. CoGent Data Sources

CoGhent provides a separate LDES for each participating cultural institution. This can be useful to differentiate between various collections at the beginning of the querying process. In theory, the order in which the URIs of these LDESs are provided as data sources to a Comunica link traversal engine determines which collection will be queried first and ultimately yield the initial results. However, in practice, the situation differs. When a link traversal engine performs multiple HTTP requests, it cannot be predetermined in which sequence the corresponding HTTP responses will reach the engine. Put differently, as the engine progresses in its *link traversal process*, a higher degree of *randomness* can be observed.

This not only implies that the sequence in which LDESs are specified is, in principle, of limited importance, but also that the Human-Made Objects present within a specific LDES may not necessarily be returned in the same order. Alongside its numerous advantages, it is essential to be acutely aware that LTQP also presents clear drawbacks. However, given that the CoGhent collections are inherently LDESs, one should never assume that the same query will yield identical results at different instances. Indeed, LDESs are effectively characterized by their significant variability.

B. Comunica Link Traversal Engine Configuration

Comunica already provides several modules and configurations to facilitate LTQP in various ways. When constructing a configuration for a link traversal engine, certain actors must be considered initially. They are responsible for the fundamental functionality of any link traversal engine. This foundational configuration is presented in a distinct configuration file, *config-base.json*, and should therefore certainly be incorporated into the final configuration deemed most suitable for executing LTQP on the CoGhent LDESs.

A pivotal decision that needs to be made for each configuration, involves the selection of a link extractor. This type of actor determines, for each incoming document, which links should be added to the link queue and subsequently visited. The most straightforward choices in this regard are the *All Extract Links Actor* and the *Quad Pattern Query Extract Links Actor*. In essence, these are implementations of the respective *cAll* and *cMatch* reachability criteria. However, it comes as no surprise that the *All Extract Links Actor*, without additional constraints, is not practically feasible. After all, an engine that simply follows every link could potentially traverse links *ad infinitum*, ultimately leading to documents that do not contain the sought-after query information. Conversely, the *Quad Pattern Query Extract Links Actor* is a viable option, particularly from the perspective of this research. Namely, the research seeks data points that are specifically associated with Human-Made Objects. In other words, the *paths* from a Human-Made Object to the relevant data points are predetermined. Since these paths are represented by the query, a *Quad Pattern Query Extract Links Actor* will at least follow the *correct* links and, above all, disregard a potentially large number of *incorrect* links.

In addition to these *standard* link extractors, Comunica offers several supplementary ones. Among them, the *Predicates Extract Links Actor* is particularly intriguing for this research. In fact, the *Predicates Extract Links Actor* conducts an even more targeted search for links, considering only those that appear as objects in quads. However, these links are added to the link queue only when their predicate matches one of the regexes defined in the actor configuration. Since the predetermined *paths* from Human-Made Objects to the sought-after data points are typically determined solely by sequences of predicates, this link extractor guarantees the fastest execution time. Yet, a significant drawback of this actor is that a new engine must be created for each new query, making its use less accessible. However, within the scope of this research, this should not pose a considerable problem. After all, the research ultimately leads to user-centric applications that, alongside their primary functionalities, can also *abstract away* this technical complexity from users.

With the explicit setting of predicates, a new challenge arises: the links referring to the previous and/or next page for a given LDES page are no longer followed, causing the engine to consider only one page per specified LDES. The various predicates leading to these links could therefore potentially be added to the predicate list, were it not for the existence of a link extractor that explicitly seeks *TREE-specific* links. Indeed, as the LDES specification is built upon the TREE specification, it is advisable to expand the current configuration with this *Extract Links Tree Actor* to encompass the entire collections in the query process. [14]

C. Links to Follow

Having the described configuration at hand, the next step should involve crafting queries. However, the practical *querya-bility* of the semantic web turns out to be less seamless than anticipated. A significant problem arises from the fact that certain resources are not hosted in full compliance with the RDF guidelines. Additionally, some of the resource types to which CoGhent Human-Made Objects refer, are affected by this issue, making it notably challenging and sometimes even impossible to include them during the link traversal process.

1) CoGhent IIIF Manifests: Starting with the positive news: the IIIF Manifests that describe the visual component of Human-Made Objects are readily accessible and interpretable for a link traversal engine. In other words, the digital image of a Human-Made Object can be fetched without difficulty alongside any other (textual) data.

2) Wikidata: Likewise, link traversal engines should generally encounter no issues with Wikidata resources. However, a note of caution is warranted. Wikidata provides two URIs for each resource and property. The *standard* URIs that Wikidata prominently *advertises* are the type of URIs that other sources - including the CoGhent LDESs - typically reference. Nonetheless, these are not the URIs that Wikidata employs *behind the scenes* to describe its RDF data. For a link traversal engine, this poses no problem, as it automatically gets redirected to the correct RDF URI. However, users must remain vigilant. When a Wikidata URI needs to appear in a query, it is imperative to explicitly employ the RDF-specific variant. As a matter of fact, this is crucial for the types of queries central to this research. After all, they typically attempt to reach data points through *paths* comprised of one or multiple explicitly defined predicate URIs.

3) Stad Gent data: Unfortunately, when a Comunica link traversal engine attempts to query a Stad Gent resource, it consistently fails. This can be attributed to a configuration error on the Stad Gent server. The server consistently responds with a *Content-Type* of *application/json* to the *Accept* header set by Comunica for its HTTP requests, even though the content is indeed a valid JSON-LD document. Ideally, this should not pose an issue, except that the server fails to provide a context link header alongside its *JSON file*, which Comunica (rightfully) expects. Until the Stad Gent server is configured correctly - not the case at the time of this research - its resources cannot be accessed by a Comunica link traversal engine.

4) Getty Vocabularies: The Getty Vocabularies server appears to suffer from a similar configuration error. It also returns JSON content based on Comunica's *Accept* header without the expected context link header. Fortunately, there is a workaround for Getty Vocabularies resources: by explicitly adding the *.json-ld* extension to their URIs, the server will actually respond with a *Content-Type* of *application/ld+json*. However, to provide a Comunica link traversal engine with this capability, a custom actor must be created that iterates over each link in a given document and, if necessary, appends the extension. Thanks to this intervention, it becomes possible to involve Getty Vocabularies resources in the link traversal process. Yet, it is evident that this solution is not optimal.

III. Tools for Query Building

An important goal of the research is to provide non-technical users with the ability to explore the CoGhent collections in conjunction with all referenced data. In other words, users without technical backgrounds should be enabled to formulate the necessary queries — albeit simple ones. In this context, the research introduces two user-friendly tools to assist in this process. However, both tools rely on the same fundamental idea: generating queries based on provided input. Hence, they both utilize a different, more *low-level* application.

A. Building Queries from Predicate Sequences

While SPARQL queries can assume complex forms, this research focuses on the simpler kind of queries that retrieve one or more characteristic *properties* for a specific type of resource — Human-Made Objects — by specifying *paths* of predicates —*predicate sequences* — in the query. This specific approach allows for the creation of a simple application that can generate a query based on a predefined sequence — can be just one — of *property* names, each specifying a sequence of predicates. Additionally, each *property* can be marked as *optional* and/or filtered.

B. User-centric Tools

Furthermore, two additional applications are introduced, primarily aiming to offer user-friendly interfaces for query construction while relying on the preceding tool for the actual query generation process.

The first application is intended for the least technical users and is consequently the simplest: users are presented with a list of pre-defined *properties* from which they can make a selection. Moreover, among other features, they are also given the option to specify certain filters. Finally, with a simple *click of a button*, they get to see the corresponding query.

The second application is slightly more challenging to use but does not limit users to only the *properties* that have been pre-selected by others. Users are expected to manually provide a resource from which they can branch out a tree of predicates and other resources. This not only provides insight into the kind of data the given resource type provides access to, but also allows users to select resources from the obtained tree as *properties* and, among other features, set filters. Once again, users are presented with the corresponding query through a simple *click of a button*.

IV. Handling Query Results

A. Visualizing Query Results

Given the research's focus on art collections, visual data holds significant importance. In the case of the CoGhent collections specifically, each Human-Made Object has a digital image associated with it. To display these images, one option would be to map all the data to a IIIF Manifest and subsequently visualize it using a IIIF Viewer of choice. The

advantage of this method lies in the avoidance of the need to build a IIIF Viewer from scratch. However, in situations where more flexibility is desired, the option to develop a custom visualization tool may be preferable.

B. Saving Query Results

Finally, saving query results might also be a crucial requirement for certain users. Once again, IIIF Manifests can be employed here. However, it is worth noting that this method necessitates some mapping system. On the other hand, the significant advantage of this approach — unlike simply storing results as *flat* files or in a database — is that the stored data can immediately be visualized.

The notion of *query results* can also be viewed from an entirely different angle. After all, in some situations, the desire might be to not hold on to specific query results but rather to the *instructions* that led to those results. The previously introduced concept a data structure that maps *property* names onto predicate sequences, for instance, meets this criterion. Nevertheless, although maintaining such a data structure might mean it can serve as input to the aforementioned applications, it obviously presents a somewhat *niche* method of storing valuable data. From this perspective, the more straightforward approach of retaining the SPARQL query itself seems to be the better idea. However, when a query is crafted with LTQP in mind, it is important to note that it cannot simply be executed with any standard SPARQL query engine. Therefore, to yield results, a users should always fall back on a link traversal engine, perhaps even the same one consistently.

Conclusion

The research into the discovery process of digital art collections, specifically CoGent's, demonstrates that LTQP can add valuable insights beyond the already known data. However, the success of this approach depends not only on well-crafted queries but also on the chosen link traversal engine. When using Comunica, the *Quad Pattern Query Extract Links Actor* proves to be an excellent link extractor. However, if the predicates to follow are extractable from the query - or the mapping between *properties* and predicate sequences - the combination of the *Predicates Extract Links Actor* and the *Extract Links Tree Actor* offers better time guarantees. Therefore, while technical knowledge is required, configuring a Comunica link traversal engine is manageable.

A more significant challenge arises from servers that are not set up in strict adherence to RDF standards. Such servers often hinder the proper functioning of Comunica link traversal engines. This is observed with Stad Gent and Getty Vocabularies resources. A specific actor has however been developed for Getty Vocabularies to work around this issue, yet this solution remains suboptimal.

In summary, the research provides valuable insights and tools for discovering digital art collections, while also highlighting the inherent challenges of the process. Link traversal undoubtedly holds the potential to uncover hidden data, but challenges like its unpredictability and typically long execution time persist. Tools that make query construction more accessible unfortunately cannot change these fundamental aspects. Provided that link traversal becomes more reliable and faster through further technological advancements, it could potentially become widely accessible in the future. However, currently, the technology still demands a certain level of technical expertise.

Acknowledgment

I would like to express my gratitude to several people. First and foremost, Bryan-Elliott Tam, for his invaluable assistance regarding link traversal, as well as his encouraging words. I would also like to thank Pieter Colpaert and Brecht Van de Vyvere for introducing me to and guiding me in the realm of Linked Data. Additionally, I am deeply appreciative of Olivier Van D'huynslager for his numerous insights into the CoGent data.

Furthermore, I want to expressly thank my family. To my parents, brother, and sister, thank you for your unwavering support. I am also very grateful to my grandparents for their constant care. And, of course, I am immensely thankful to my girlfriend for continuously boosting my spirits during challenging moments and helping me successfully conclude my research.

References

[1] P. Van Leemputten, "Gent gaat cultureel erfgoed virtueel samenbrengen," *DataNews*, July 2020, https://datanews.knack.be/nieuws/gent-gaat-cultureel-erfgoed-virtueel-samenbrengen/.

[2] W. Schouppe, "Gent roept inwoners op erfgoed in te sturen én te onderzoeken op een nieuw online platform: "we hopen op 50.000 inzendingen"," *VRT NWS*, September 2022, https://www.vrt.be/vrtnws/nl/2022/09/27/gent-vraagt-inwoners-erfgoed-in-te-sturen-en-te-onderzoeken-op-e/.

[3] "Coghent data," June 2023, https://coghent.github.io/LDES/.

[4] P. Colpaert, "Linked data event streams," W3C, W3C Living Standard, April 2023, https://semiceu.github.io/LinkedDataEventStreams/.

[5] B. Van de Vyvere, O. V. D'Huynslager, A. Atauil, M. Segers, L. Van Campe, N. Vandekeybus, S. Teugels, A. Saenko, P.-J. Pauwels, and P. Colpaert, "Publishing cultural heritage collections of ghent with linked data event streams," in *Metadata and Semantic Research: 15th International Conference, MTSR 2021, Virtual Event, November 29–December 3, 2021, Revised Selected Papers*. Springer, 2022, pp. 357–369.

[6] N. Vanderperren, "Publicatie:oslo cultureel erfgoed," June 2021, https://www.projectcest.be/wiki/Publicatie:OSLO_Cultureel_Erfgoed.

[7] "Presentation api 2.1.1," June 2017, https://iiif.io/api/presentation/2.1/.

[8] J. P. Emanuel, "Stitching together technology for the digital humanities with the international image interoperability framework (iiif)," in *Digital Humanities, Libraries, and Partnerships*. Elsevier, 2018, pp. 125–135.

[9] S. Snydman, R. Sanderson, and T. Cramer, "The international image interoperability framework (iiif): A community & technology approach for web-based images," in *Archiving conference*, vol. 2015. Society for Imaging Science and Technology, 2015, pp. 16–21.

[10] R. Taelman, "Link traversal-based query processing," May 2023, https://www.rubensworks.net/raw/slides/2023/ugent-webfundamentals-linktraversal/.

[11] O. Hartig and J.-C. Freytag, "Foundations of traversal based query execution over linked data," in *Proceedings of the 23rd ACM conference on Hypertext and social media*, 2012, pp. 43–52, https://arxiv.org/pdf/1108.6328.pdf.

[12] R. Taelman, J. Van Herwegen, M. Vander Sande, and R. Verborgh, "Comunica: a modular sparql query engine for the web," in *Proceedings of the 17th International Semantic Web Conference*, Oct. 2018. [Online]. Available: https://comunica.github.io/Article-ISWC2018-Resource/

[13] R. Taelman, “Link traversal for comunica,” 2019, https://github.com/comunica/comunica-feature-link-traversal.
[14] P. Colpaert, “The tree hypermedia specification,” W3C, W3C Draft, May 2023, https://treecg.github.io/specification/.