

# Real-time Collaboration in Linked Data Systems

Jonathan Gruss<sup>1,\*</sup>, Andrei Ciortea<sup>1</sup>, Guido Salvaneschi<sup>1</sup> and Simon Mayer<sup>1</sup>

<sup>1</sup>*Institute of Computer Science, University of St.Gallen, St. Gallen, Switzerland*

## Abstract

Real-time collaboration has become commonplace in centralized Web applications, but decentralized Linked Data systems still lack readily accessible mechanisms. This demo paper proposes a novel approach that provides a viable solution to implement collaborative Linked Data in the Solid ecosystem using Conflict-free Replicated Data Types (CRDTs) and hypermedia-driven interaction. Specifically, we introduce a dedicated vocabulary for describing interactions with CRDT-based resources hosted in Solid Pods, empowering software clients to dynamically discover means for collaborative editing at run time. In contrast to current solutions for collaborative RDF, our approach works in combination with industry standard CRDTs to offer a seamless co-editing experience in decentralized Linked Data systems. To demonstrate the practicality of our approach, we showcase a Solid-hosted website that utilizes the vocabulary to expose hypermedia controls and a browser extension that effectively consumes these descriptions to enable real-time collaborative editing through CRDTs. By strategically shifting intelligence to the client-side, our approach significantly lowers the entry barrier for publishing real-time collaborative resources on the (Semantic) Web.

## Keywords

Real-Time Collaboration, Linked Data, CRDT, Solid, Ontology, RDF

## 1. Introduction

Many online co-editors, such as Google Docs or Overleaf, enable collaboration on Web resources in real time through Operational Transformation (OT) — a technique for transforming and applying concurrent operations on shared data without conflicts [1]. However, OT algorithms rely on centralized architectures, which introduce scalability issues, limit interoperability, and require end users to give up control over their data. In decentralized Linked Data systems, such as those based on Solid [2], there are no straightforward solutions for real-time collaborative editing.

Related work on real-time collaborative Linked Data is using Conflict-free Replicated Data Types (CRDTs) [3] to implement strong eventual consistency for RDF graphs [4, 5, 6]. While these approaches focus on the implementation of a shared RDF data type, they do not consider the discoverability of the collaboration interface, the integration of popular shared data formats, or the hosting and access control of the Linked Data. Additionally, current CRDT implementations

---

*ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece*

\*Corresponding author.

✉ jonathan.gruss@student.unisg.ch (J. Gruss); andrei.ciortea@unisg.ch (A. Ciortea); guido.salvaneschi@unisg.ch (G. Salvaneschi); simon.mayer@unisg.ch (S. Mayer)

🆔 0009-0004-2927-0298 (J. Gruss); 0000-0003-0721-4135 (A. Ciortea); 0000-0002-9324-8894 (G. Salvaneschi); 0000-0001-6367-3454 (S. Mayer)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

for RDF use RDF triples as the smallest directly manageable piece of knowledge, while our approach allows for more fine-grained collaborative editing. Instead of providing a single shared data type for RDF, our approach allows for a choice of CRDT implementations such as Yjs [7] and Automerge [8] and a combination of arbitrary data types. This makes it possible to have more fine-grained collaborative editing of Linked Data, e.g. using text-based CRDTs with single-character updates for an RDF node, which creates a collaborative editing experience that is similar to popular co-editors.

In this paper, we propose a novel approach based on CRDTs for enabling *collaborative Linked Data* in the Solid ecosystem. We introduce a vocabulary for describing interactions with CRDT-based resources hosted in Solid Pods, which allows software clients to discover at run-time the required means for collaboratively editing the resources. To demonstrate our approach, we showcase a Web blog stored in a Solid Pod that uses our vocabulary to expose hypermedia controls for its collaboration interface. A Chrome extension then consumes these descriptions and makes the Web blog’s content and its Linked Data annotations collaboratively editable through CRDTs. In contrast to relying on OT servers, our approach shifts the intelligence to the client – and lowers the entry-barrier for making Web resources collaborative in real-time. The full code and ontology is publicly available at <https://tinyurl.com/55epp24d><sup>1</sup> and a demonstrator video is available online<sup>2</sup>.

## 2. Building Real-time Collaborative Linked Data Systems

Our approach to *collaborative Linked Data* is inspired by the principles of local-first software [9] to enable user-controlled and privacy-preserving collaboration using CRDTs and intelligent clients. We use the Solid protocol to provide decentralized and portable data hosting with access control, and introduce a vocabulary that complements the Hypermedia Controls Ontology (HCTL)<sup>3</sup> to create hypermedia controls for real-time collaboration on CRDT-based resources.

**CRDT-based Representation of Linked Data Resources** An alternative to the OT approach used by traditional collaborative editors are Conflict-Free Replicated Data Types (CRDTs), which allow replicas to be updated concurrently and without coordination: inconsistencies are automatically resolved and the system is guaranteed to eventually converge. Hence, CRDTs allow to represent and manage collaborative resources without relying on a central coordinator, which makes them well-suited for decentralized Linked Data systems – since we do not require an expensive origin server to integrate the changes.

While some efforts have been made to implement CRDTs for RDF, they do not offer the update granularity of popular text-based co-editors. To address this shortcoming, we propose to create a solution that can be used with various CRDT implementations and specifically test it on the widely-used Yjs and Automerge libraries. Both of these libraries are based on documents containing a combination of various CRDTs such as maps, arrays, or text, and the entire document state can be represented in a JSON format with the specific CRDTs as

---

<sup>1</sup>GitHub repository

<sup>2</sup><https://clipchamp.com/watch/tZdpD5ONm6t>

<sup>3</sup><https://www.w3.org/2019/wot/hypermedia#>

properties. As a result, by adding specific keywords such as a `@context` to the document, it becomes feasible to create CRDTs that can be represented in the JSON-LD format using popular CRDT libraries. Thus, this allows us to provide more targeted and fine-grained collaboration on RDF nodes by giving users more choice over the employed CRDT, e.g., using text-based CRDTs for nodes with longer text. Additionally, this open approach allows us to potentially support a large range of widely-used CRDT frameworks.

Additionally, CRDT documents are often stored and shared in binary implementation-specific formats for conciseness and performance reasons, necessitating the specific CRDT framework to interpret the data. To address this, we introduce a textual representation of the binary CRDT, allowing any Web client to access the most recent state of the CRDT; only clients actively collaborating require specialized software (e.g., a Web browser plugin) to resolve the CRDTs. The server stores both the binary CRDT document including the complete change history and a textual representation of the latest state. Updating the textual representation operates similarly to a Git commit-and-push, enabling users to contribute their local state.

**Collaborative Resource Description Ontology (CRDO)** Traditional collaborative editors rely on fixed contracts imposed by static APIs, which necessitates hard-coding into clients and limits interoperability among different systems. In contrast, the Semantic Web already provides means to support more open interactions on the Web through hypermedia controls, by using vocabularies such as Hydra and HCTL.

We propose CRDO, a vocabulary that complements existing ones to create hypermedia controls for collaborative Web resources, describing the collaborative document and advertising possible client actions for reading, manipulating, and synchronizing with peers.

The `crdo:CollaborativeResourceDescription`, which is used to declare the entry point of a collaborative Web resource, is central to the CRDO vocabulary. Such descriptions may contain `crdo:DocumentDescription` and `crdo:TextualRepresentationDescription`. The former lists possible actions for the CRDT document and metadata such as the utilized CRDT framework, which is relevant because mainstream CRDT frameworks use implementation-specific formats and synchronization mechanisms. The latter specifies possible actions for clients to interact with the current public version of the CRDT state in textual form.

We refer to the possible actions that are advertised to clients by both descriptions as *CollaborativeOperations (CoOp)*; these are the key concept of our collaborative resource vocabulary. To foster reuse and interoperability, CoOp are based on the *Form* concept from HCTL. Our vocabulary additionally introduces a differentiation between Web CoOp and Framework CoOp. Web CoOp represent classic HTTP-based interactions and is aligned with HCTL. Framework CoOp are used for interactions that are mainly handled or executed by the CRDT framework of the resource. Framework CoOp can be necessary, for example, to establish real-time synchronization with a server or peers using WebSocket or WebRTC.

Beyond facilitating collaborative resource descriptions and hypermedia controls, CRDO also introduces terms to describe textual CRDT representations in a Linked Data format. The CRDO vocabulary provides classes, properties, and data types to declare the Web resource as a textual representation, link to the collaborative resource description, and annotate resource values with the utilized CRDT data type.

**Solid Hosting and Access Control** To support open collaboration on Linked Data, our approach uses Solid Pods to host CRDT documents, their textual representations, and their semantic descriptions. By hosting these resources on a Solid Pod, we can utilize its fine-grained, decentralized access control mechanism which makes it possible to, e.g., create a publicly discoverable version of the resource representation in RDF while restricting access to the hypermedia controls and shared resources for collaboration. Another important aspect is that the control of data is given back to the end user, which is consistent with the use of CRDTs for collaborative editing — since CRDTs do not require a central coordinator. Thus, the user is in full control not only of the local CRDT document, but also of its online counterpart and RDF representation.

### 3. Demonstration

To showcase the practical implementation of our approach, we demonstrate real-time collaboration on a static website hosted in a Solid Pod, allowing collaborative editing of the website's content directly within a web browser. In addition, we integrate a GUI-based Linked Data editor, enabling users to update the Linked Data context of the website's content.

We accomplish this by providing a collaborative resource description of the content using our vocabulary. This description can be consumed by a software client; in our case, we developed a Chrome browser extension for this purpose. The extension utilizes the discovered CRDT state or creates a new CRDT state of the content using the Yjs<sup>4</sup> CRDT framework. Moreover, it leverages the CoOp specified in the description to facilitate user collaboration through three main functionalities: (i) presenting a pop-up with actions to initiate the collaboration, synchronize changes in real-time with other peers using WebRTC, and commit the current state to the Solid Pod; (ii) converting the website elements that utilize the collaborative content into editable elements and synchronizing their content with the state of the CRDT document; and (iii) introducing a Linked Data editor panel offering utility functions to read and update the term definitions of the content's context.

### 4. Conclusion

Our approach significantly simplifies the implementations of real-time collaboration in Linked Data Systems by providing vocabulary to describe the collaborative interface. Moreover, it ensures access control, interoperability, and discoverability of shared data types by leveraging the Solid protocol, hypermedia controls, and CRDTs with textual representations. Ongoing and future work on our current approach include testing and expanding our system's support for various CRDT implementation frameworks and formats as well as the validation of the approach in other realistic use cases to assess its versatility and effectiveness. We furthermore plan to publish a specification and stable implementation for broader adoption and feedback, and to explore the potential extension of our approach beyond the Solid ecosystem as well as its application in existing Linked Data systems.

---

<sup>4</sup><https://github.com/yjs/yjs>

## References

- [1] C. Sun, C. Ellis, Operational transformation in real-time group editors: Issues, algorithms, and achievements, in: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, CSCW '98, Association for Computing Machinery, New York, NY, USA, 1998, p. 59–68. URL: <https://doi.org/10.1145/289444.289469>.
- [2] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Abounaga, T. Berners-Lee, A demonstration of the solid platform for social web applications, in: Proc. of WWW '16 Companion, WWW Conferences Steering Committee, 2016, p. 223–226. URL: <https://doi.org/10.1145/2872518.2890529>.
- [3] M. Shapiro, N. Preguiça, C. Baquero, M. Zawirski, Conflict-free replicated data types, in: X. Défago, F. Petit, V. Villain (Eds.), Stabilization, Safety, and Security of Distributed Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 386–400.
- [4] L. D. Ibáñez, H. Skaf-Molli, P. Molli, O. Corby, Synchronizing semantic stores with commutative replicated data types, in: Proc. of WWW '12 Companion', ACM, 2012, p. 1091–1096. URL: <https://doi.org/10.1145/2187980.2188246>.
- [5] M. D. Mechaoui, N. Guetmi, A. Imine, Towards real-time co-authoring of linked-data on the web, in: Computer Science and Its Applications: 5th IFIP TC 5 International Conference, CIIA 2015, Saida, Algeria, May 20-21, 2015, Proceedings 5, Springer, 2015, pp. 538–548.
- [6] H. Zarzour, M. Sellami, srCE: A Collaborative Editing of Scalable Semantic Stores on P2P Networks, *Int. J. Comput. Appl. Technol.* 48 (2013) 1–13. URL: <https://doi.org/10.1504/IJCAT.2013.055562>.
- [7] P. Nicolaescu, K. Jahns, M. Derntl, R. Klamma, Yjs: A Framework for Near Real-Time P2P Shared Editing on Arbitrary Data Types, in: Proceedings of the 15th International Conference on Engineering the Web in the Big Data Era - Volume 9114, ICWE 2015, Springer-Verlag, Berlin, Heidelberg, 2015, p. 675–678. URL: [https://doi.org/10.1007/978-3-319-19890-3\\_55](https://doi.org/10.1007/978-3-319-19890-3_55).
- [8] M. Kleppmann, A. R. Beresford, Automerge: Real-time data sync between edge devices, in: 1st UK Mobile, Wearable and Ubiquitous Systems Research Symposium (MobiUK 2018), 2018, pp. 101–105.
- [9] M. Kleppmann, A. Wiggins, P. van Hardenberg, M. McGranaghan, Local-First Software: You Own Your Data, in Spite of the Cloud, in: Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward! 2019, ACM, 2019, p. 154–178. URL: <https://doi.org/10.1145/3359591.3359737>.