# Causal Loop? Do It Yourself! (CL-DIY): An open-source and browser-based editor to share causal loop diagrams

Jiri Kaan, Ciska Ulug, Kristina Thompson, Annemarie Wagemakers, Spencer Moore
Health & Society, Wageningen University, The Netherlands

27 October 2026

## Summary

Causal loop diagrams (CLDs) are a qualitative system dynamics method. They show the variables in a system, the causal links between them and the polarity of each link, and they make reinforcing and balancing feedback loops visible. CLDs are often built together with stakeholders in group model building and other participatory settings. There, the diagram is a shared object that people reorganise, annotate and argue over. However, after a participatory session, the CLD usually ends up in specialist modelling software that participants cannot easily open, let alone change.

We therefore developed CL-DIY (Causal Loop? Do It Yourself!), an open-source, browser-based editor for building, exploring and refining CLDs. It lets a group revisit its diagram after the session: walk through it loop by loop, correct it themselves, and keep it as a record of what they decided and why. CL-DIY does this with a single file that anyone can open in a web browser, with no installation, no account and no server, so the data stays with its owners. In that file, participants can edit the diagram, group variables, trace and label feedback loops, and note the reason for each change. The result can be shared, edited further, archived alongside a publication and exchanged with other system dynamics tools. The most recent version of CL-DIY can always be found at https://github.com/Kajijoo/CL-DIY.

## Statement of need

CLDs make the feedback structure of a system explicit (Sterman 2000), and they are increasingly built together with stakeholders in group model building and other participatory settings (Hovmand 2014; Barbrook-Johnson and Penn 2022). Researchers who run participatory system-mapping sessions often finish with a CLD in a desktop modelling tool such as Vensim. That is a less accessible format for the next step: sending the map back to participants, letting them rearrange and annotate it on the spot, and recording which feedback loops the group identified. Proprietary tools need a licence and some training. Online mapping platforms need accounts and store the data on third-party servers, which can be a problem for sensitive community or health data. CL-DIY is for researchers and facilitators who want one file that they can easily e-mail, put in a shared drive or archive alongside a publication, and that anyone can open and edit and comment on in a web browser.

CL-DIY was developed in the Health in All Networks Simulator (HANS) project, which uses group model building workshops with members of two communities in Amsterdam, together with policy stakeholders, to explore social network interventions for health and well-being (Kaan et al. 2025). After the workshops, we found the current software insufficient for communication purposes, which was disappointing since a strength of CLDs is that they are a communication tool. We wanted to walk through the feedback loops one at a time, let participants explore the diagram themselves, and change it on the spot as they suggested corrections while keeping track of changes. CL-DIY was built for exactly that use.

## State of the field

Tools that support CLDs fall into three groups. (1) System dynamics modelling environments such as Vensim (Ventana Systems, n.d.) and Insight Maker (Fortmann-Roe 2014) support CLD sketching, but they are designed for stock-and-flow simulation, and most need a licence or an account. (2) General systems-mapping platforms such as Kumu (Kumu, n.d.) offer rich collaborative mapping, but they are hosted services. (3) Lightweight, open-source browser tools such as LOOPY (Case 2017) are excellent for teaching and exploring feedback, but they do not import existing models and do not record loop annotations or provenance. All of them do not allow for easy sharing and keeping track of edits that can be downloaded which is useful when revisiting the CLD and when writing reports/papers about the final CLD.

CL-DIY occupies a narrow niche between these groups. In CL-DIY, you can start a CLD completely from scratch. It also imports and exports models in the standard XMILE exchange format (OASIS 2015), and its output is a durable, offline, single file that carries the full edit history, with the reasons for each change. We built a new tool, rather than contributing to an existing one, because we wanted a simple, fit-for-purpose browser-based editor for this specific use: communicating and refining a CLD together with participants.

## Software design

CL-DIY turns a CLD into one self-contained, interactive HTML file that runs fully in the browser. Users can add, rename, merge and delete variables, draw causal links by dragging from one variable to another, set the polarity of each link, group variables into user-defined and colour-coded subsystems, and mark feedback loops, which are checked to be valid cycles, as reinforcing or balancing. Variables stay where the user places them; an optional force-directed layout can tidy the diagram on request. Edits can be undone and are recorded in a change log. By default, each edit asks for an optional comment on why it was made, so the log records the group's reasoning as well as its changes; the log can be downloaded as a CSV file. The annotated diagram, any selection of its feedback loops, or a single variable with its direct causes and effects can be exported as a PNG figure, with a legend of the subsystem colours.

Nothing is stored automatically, so the work must be saved to continue editing later. The full state (positions, subsystems, loops and the commented change log) is saved as a JSON file or as a new copy of the HTML editor with the diagram inside, and the causal structure can be exported as XMILE. Existing diagrams can be uploaded as JSON or as a Vensim-exported XMILE file, either in the browser or through a small Python command-line tool that builds the HTML file.

CL-DIY has two layers:

- **Python build step.** `sd.py` parses XMILE and reads the causal parents declared in each auxiliary's equation, so it also works for CLD-only exports whose equations are placeholders. It can optionally drop the duplicate copy that Vensim creates, with a `_0` suffix on every name, when two diagrams are merged into one file. `layout.py` computes an initial Fruchterman-Reingold force-directed layout (Fruchterman and Reingold 1991) with a fixed seed, so builds are reproducible. `render.py` serialises the graph into a JSON payload and inlines it, together with the fonts and the HTML, CSS and JavaScript templates, into one HTML document.
- **Browser application.** A dependency-free JavaScript/SVG application, embedded in the HTML, handles all editing. State is a plain JSON object of nodes, edges, subsystems, loops and a change log with comments. Undo uses snapshots of this state, and saving as JSON writes the same object out. The Python side reads it back in unchanged, so a diagram can go back and forth between the build step and the browser. Saving as HTML needs no Python: the page writes a copy of itself in which the current state replaces the embedded payload, so every saved diagram is itself a complete editor.

The main trade-off is that a self-contained file gives up real-time multi-user collaboration and server-side persistence. In return, CL-DIY needs no infrastructure, keeps data local and can be archived (for example on Zenodo) as an exact record of a modelling session.

A feedback loop is saved only if it is a real cycle in the current diagram: the user picks at least two variables in order, and there must be a direct causal link from each variable to the next and from the last back to

the first. Loops whose links are later deleted are removed, and loops in an uploaded file are checked again on import. Subsystems are a free-form grouping axis: a new diagram starts with four generic subsystems, which users can rename, recolour, add and delete to fit their own framework.

CL-DIY has clear limits. It does not simulate: it records the causal structure and the group's interpretation of it, not equations. CL-DIY is kept simple on purpose. XMILE import and export carry only the variables and causal links, so link polarity, positions, subsystems and loops are set in CL-DIY and kept in its own JSON and HTML files. It has been tested with diagrams of up to 80 variables; much larger diagrams are likely to become hard to read before they become slow.

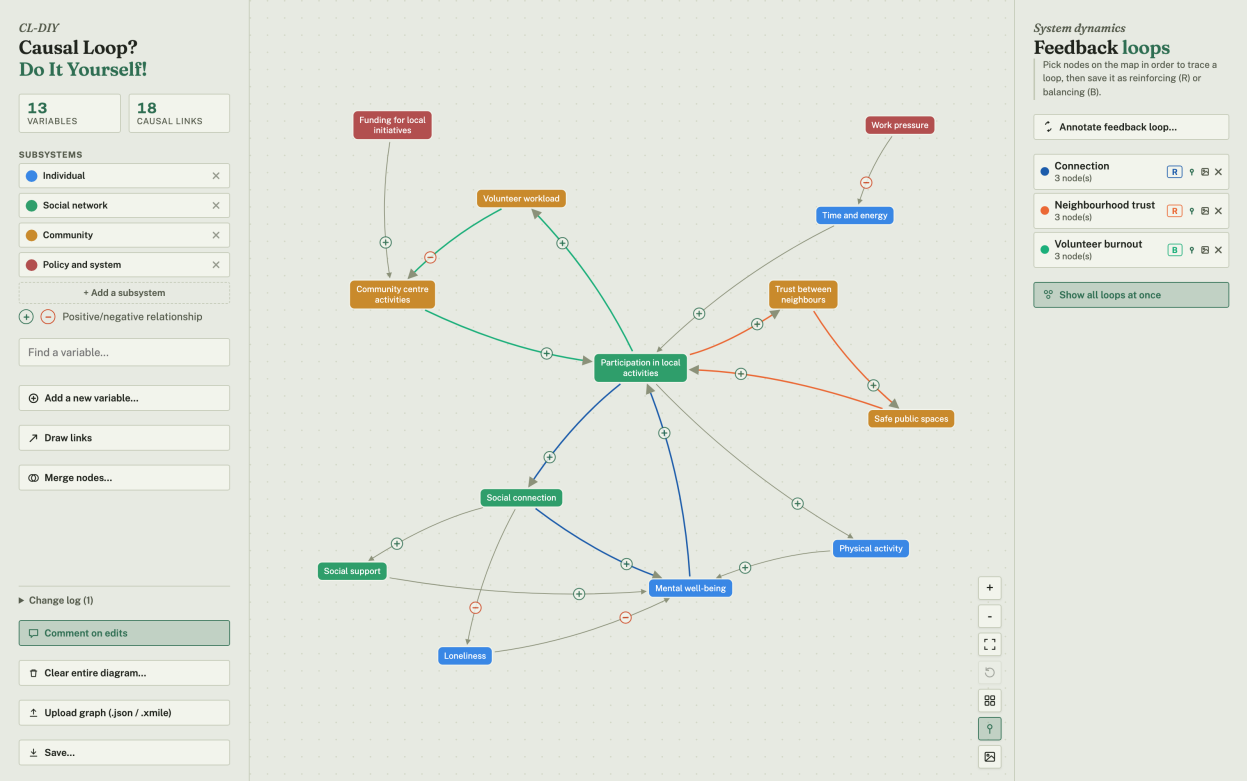



Figure 1: The CL-DIY interface showing a synthetic example CLD on social connection and health in a neighbourhood (`examples/example.json` in the repository). Variables are colour-coded by subsystem, each link carries its polarity, and three annotated feedback loops (two reinforcing, one balancing) are shown at once in their own colours.

## AI usage disclosure

Parts of the source code and documentation were written with the assistance of Anthropic's Claude Code, Opus 5.5. The authors designed the software, reviewed and tested all AI-assisted code, and wrote/edited this paper.

## Acknowledgements

CL-DIY is open source under the Apache License 2.0. The source code, a ready-to-use editor (`CL-DIY.html`) and an example diagram are available at https://github.com/Kajijoo/CL-DIY. Version 1.1.0, described in this paper, is archived on Zenodo (Kaan et al. 2026).

This work was supported by ZonMw - Preventie in transitie: Complexiteit omarmen en veerkracht versterken

(project number: 05550032110022).

## References

Barbrook-Johnson, Pete, and Alexandra S. Penn. 2022. *Systems Mapping: How to Build and Use Causal Models of Systems*. Palgrave Macmillan. https://doi.org/10.1007/978-3-031-01919-7.

Case, Nicky. 2017. "LOOPY: A Tool for Thinking in Systems." https://github.com/ncase/loopy.

Fortmann-Roe, Scott. 2014. "Insight Maker: A General-Purpose Tool for Web-Based Modeling & Simulation." *Simulation Modelling Practice and Theory* 47: 28–45. https://doi.org/10.1016/j.simpat.2014.03.013.

Fruchterman, Thomas M. J., and Edward M. Reingold. 1991. "Graph Drawing by Force-Directed Placement." *Software: Practice and Experience* 21 (11): 1129–64. https://doi.org/10.1002/spe.4380211102.

Hovmand, Peter S. 2014. *Community Based System Dynamics*. Springer. https://doi.org/10.1007/978-1-4614-8763-0.

Kaan, Jiri, Ciska Ulug, Kristina Thompson, Yara Khaluf, Annemarie Wagemakers, and Spencer Moore. 2025. "Health in All Networks Simulator: Mixed-Methods Protocol to Test Social Network Interventions for Resilience, Health and Well-Being of Adults in Amsterdam." *BMJ Open* 15 (April): e100703. https://doi.org/10.1136/bmjopen-2025-100703.

Kaan, Jiri, Ciska Ulug, Kristina Thompson, Annemarie Wagemakers, and Spencer Moore. 2026. *CL-DIY: Causal Loop? Do It Yourself!* V. 1.1.0. Zenodo, released October 26. https://doi.org/10.5281/zenodo.23662004.

Kumu. n.d. *Kumu*. Kumu Inc. Accessed October 19, 2026. https://kumu.io.

OASIS. 2015. *XML Interchange Language for System Dynamics (XMILE) Version 1.0*. OASIS. https://docs.oasis-open.org/xmile/xmile/v1.0/xmile-v1.0.html.

Sterman, John D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill.

Ventana Systems. n.d. *Vensim*. Ventana Systems, Inc. Accessed October 19, 2026. https://vensim.com.