

nev: A Visual Query Tracer and Builder for Declarative Logic Programming on Nemo

Julián Méndez¹, Lukas Gerlach², Tobias Wieland¹, Alex Ivliev², Markus Krötzsch², Raimund Dachsel¹

¹ Interactive Media Lab Dresden, TU Dresden ² Knowledge-Based Systems Group, TU Dresden

EuroVis 2026

The Eurographics Conference on Visualization



Motivation: Declarative Logic Programming

- Datalog-like languages use rules (if-then statements) to express what should be computed rather than **how**.
- Renewed interest for applications in system engineering, health-care data analysis, web data extraction, etc.
- Tools like Nemo extend Datalog with additional features (aggregation, negation, etc). Modern IDEs for Datalog extensions are very powerful and comprehensive.
- However, scarce resources exist to **learn** and **reason** with such languages! Does visualization help?



Examples Help

Example Nemo Program

```

1 % Shortest paths in weighted directed graph (i.e., each edge is assigned a cost)
2
3 e(1,2,7).e(1,4,2).e(1,7,15).e(2,3,17).e(2,9,3).e(3,5,9).e(3,8,14).e(4,2,6).e(4,6,1).
4 e(5,1,8).e(5,10,2).e(6,3,13).e(6,9,3).e(7,5,5).e(8,2,4).e(9,7,2).e(9,10,9).
5
6 totalCost(#sum(?cost, ?from, ?to)) :- e(?from, ?to, ?cost).
7
8 reachableIn(?from, ?to, ?cost) :- e(?from, ?to, ?cost).
9 reachableIn(1, ?to, ?cost1+?cost2) :-
10   reachableIn(1, ?mid, ?cost1),
11   e(?mid, ?to, ?cost2),
12   totalCost(?totalCost),
13   ?cost1 + ?cost2 < ?totalCost.
14
15 minDistancesFromNode1(?to, #min(?cost)) :- reachableIn(1, ?to, ?cost), ?to != 1.
    
```

main visual trace editor

The screenshot displays the nev interface. On the left is the 'explorer view' showing a list of Datalog rules. The main area is the 'main visual trace editor', which shows a tree structure of query execution. A central node is expanded to show a table of facts: 'reach...(1, ?mid, ?cost1)' with columns for ?mid and ?cost1, and rows of values (1, 49), (1, 52), (1, 54). Below this is another table: 'Table: reachableIn(1, ?mid, ?cost1)' with columns for ?mid and ?cost1, and rows of values (1, 49), (1, 52), (1, 54), (1, 66). On the right is the 'fact-tables view' showing a table: 'Table: e(?from, ?to, ?cost)' with columns for ?from, ?to, and ?cost, and rows of values (1, 2, 7), (1, 4, 2), (1, 7, 15), (2, 3, 17). The interface includes search, undo, redo, and query restriction controls.

explorer view

fact-tables view

- Nodes in the main view display the facts computed at that point of the trace as well as performance measures.
- The explorer view can be used to more easily **navigate** the main trace view (hover, click to pan).
- Fact tables can be pinned to the fact-tables view for paginated inspection and **comparison**.
- Trace editing can be disabled for incremental **exploration** and we provide options to undo/redo, search, adjust text-length, and edit query restrictions.

Contribution: Web-based Datalog Environment

- nev (Nemo Explain Visualizer) added to the development environment for Nemo (accessible through Nemo Web).
- Type 1 queries trace facts and result in **proof trees**. Type 2 queries take a **tree shape** and fill it with facts from every trace matching the given shape.
- **Visually editing** the trace triggers Type 2 queries: add/remove nodes, focus on rule, etc.
- The result? nev simultaneously supports the tasks of visual analysis, debugging, and query building.

Related Work:

- [1] DALIBO: Pev2. <https://github.com/dalibo/pev2>, 2025. Accessed: 2025-07-01
- [2] DUDAŠ M., LOHMANN S., SVÁTEK V., PAVLOV D.: Ontology visualization methods and tools: a survey of the state of the art. doi:10.1017/S0269888918000073
- [3] IVLIEV A., GERLACH L., MEUSEL S., STEINBERG J., KRÖTZSCH M.: Nemo: Your friendly and versatile rule reasoning toolkit. doi:10.24963/kr.2024/70
- [4] JORDAN H., SCHOLZ B., SUBOTIC P.: Soufflé: On synthesis of program analyzers. doi:10.1007/978-3-319-41540-6_23
- [5] KRÖTZSCH M.: Modern Datalog: Concepts, methods, applications. doi:10.4230/OASIS.RW.2024/2025.7

Implementation:

TypeScript: React + Vite
 Styling: Material UI, Data Grid, main trace uses d3-flextree
 T2 allows custom pagination and fact count, so only shown data is loaded
 History (undo/redo) uses the Memento Pattern
 Serverless cross-browser-tab messaging achieved with the Broadcast Channel API

Project Website:
imld.de/nev

