

ArcUno: A Topology-Sensing Tangible Toolkit for Lean Knowledge Graph Modeling

Filippos Taprantzis^{1,2}, Victor de Boer¹

¹Vrije Universiteit Amsterdam, the Netherlands

²University of Twente, the Netherlands

Abstract

ArcUno is a topology-sensing tangible toolkit for lean graph-based conceptual modeling. It lets users physically assemble nodes and directed edges on a desk while a computer-vision pipeline detects ArUco fiducials, reconstructs the topology, and serialises the result as a digital Knowledge Graph. The system combines graspable node and edge artifacts, automatic geometric calibration, real-time physical-digital synchronisation, and export to RDF/Turtle or JSON. This paper derives design constraints from prior work on external cognition and tangible interaction, then presents the design and implementation of a working prototype.

Keywords

Tangible user interfaces, conceptual modeling, knowledge graphs, ArUco fiducials, topology sensing

1. Introduction

Conceptual modeling is a foundational activity for understanding and designing complex systems. One of the representations that can result from conceptual modeling is a Knowledge Graph (KG) that represents entities and concepts as nodes and their relationships as edges within a directed, labeled graph[1]. However, the modeling process can often be cognitively demanding, as the networked structure can make it difficult to maintain an overview while simultaneously reasoning about details. Typically, KG construction is supported by digital tools, such as Protégé¹ or yEd², with screen, mouse, and keyboard being the primary interaction media. Literature on diagrams and external cognition suggests that the interface and representation form can change the cognitive work required for reasoning. Previous research has shown that diagrams can spatially index information and support perceptual inferences, with the benefits of external representations depending on how well the representation fits the task [2, 3]. Tangible user interfaces offer a promising extension to the conceptual modeling process. For instance, Tangible Bits explores how digital information can be coupled to real objects and physical space, with later frameworks extending this to include embodied experiences, physical representation of data, and space as additional design dimensions [4, 5, 6]. However, incorporating embodied experiences and tangibility in a modeling toolkit is not inherently beneficial and it is important to consider how the final representation influences the interaction and cognition [7]. In this paper, we use prior work on external representations and tangible interaction to motivate requirements for lean graph-based modeling, then describe the design and implementation of the **ArcUno** toolkit. The ArcUno prototype explores how physical node-edge assemblies can support the construction of small, inspectable KGs.

2. Background

Conceptual modeling is best understood as a process rather than a static artifact. Hoppenbrouwers et al. describe it as a structured process in which people share and align their representations, which frames modeling difficulty as something that emerges during construction rather than only in the finished

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

✉ f.v.taprantzis@student.utwente.nl (F. Taprantzis); v.de.boer@vu.nl (V. de Boer)



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

¹<https://protege.stanford.edu/>

²<https://www.yworks.com/products/yed>

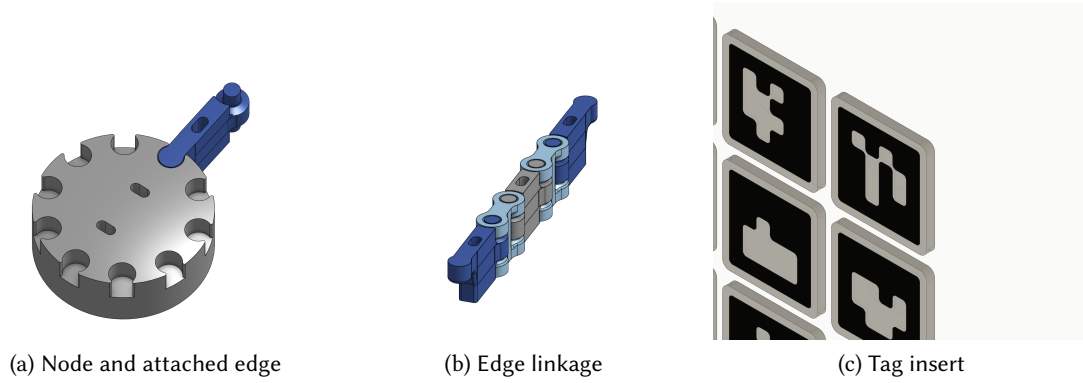


Figure 1: CAD renders of various physical components of ArcUno

diagram [8]. In this work, we target *lean* graph-based modeling which consists of constructing a small, inspectable network of concepts and relations that can later be encoded as a Knowledge Graph digitally.

For our prototype, we informed our requirements on previous work and research. Novak and Canas formalise concept maps as graphs of labelled nodes and relations, which motivates naming nodes and relation types explicitly rather than relying on proximity [9]. Oppl and Stary present a tabletop concept-mapping interface that supports externalisation during co-construction [10]. Furthermore, maintaining real-time consistency between physical and digital state is an established idea: ActiveCube senses the structure of connected cubes and keeps it consistent with a virtual representation [11], and Kwak et al. show that connections are easier to understand when they are visible in the manipulated objects rather than only on a central display [12]. These systems provide precedents for tangible graph assembly, tabletop concept mapping, and physical management of digital connections. ArcUno builds on these ideas in the narrower setting of lean KG construction, where the physical arrangement must remain inspectable while still producing a machine-readable graph.

3. Toolkit Design

3.1. Components

The ArcUno toolkit consists of 1) physical graph components; 2) an overhead capture setup; and 3) a host application for detecting and interpreting their arrangement. ArcUno’s physical components include *nodes* and *edges*. Fig. 1 shows selected elements of the design as CAD renders: a node with an attached edge linkage, an articulated edge component, and a fiducial tag used for sensing.

A node is a short cylinder (53 mm in diameter, 17 mm tall) with ten ports arranged around its circumference, each of which accepts the peg end of an edge. The peg/socket coupling follows the familiar wooden-train-track principle and engages over roughly the lower 7 mm of the component’s height, so that pieces snap together and pull apart with low effort. An edge is a 3D-printed articulated linkage of the same height, with four revolute joints that allow it to bend to reach between nodes placed at arbitrary positions and orientations. This design serves low-cost, reversible construction directly: a user can connect, disconnect, reposition, and revise parts of the model without treating any action as final, supporting the kind of exploratory, epistemic manipulation that benefits conceptual modeling.

Identity and meaning are carried by small (14 × 14 mm) ArUco *fiducial tags* that snap into obround sockets by friction fit. A node carries a single tag; an edge carries two, one at each end. The two-tag scheme is central to the approach, because it lets a single edge piece encode both *which* two nodes it joins and the *direction* of the relation between them: the ordering of the start and end tags is fixed when the edge is registered. This feature allows for modelling of *directed*, *labeled* graphs. The same socket interface also accepts human-readable labels, so the operator can attach printed concept names to nodes and printed relation names, which include an arrow glyph indicating direction, to edges. As a

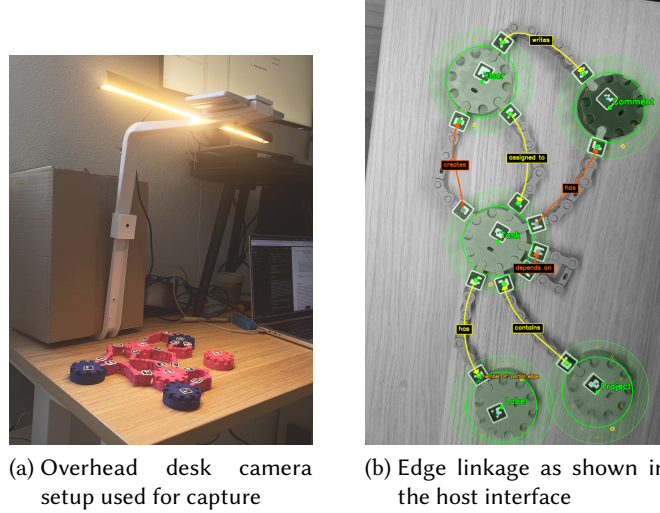


Figure 2: ArcUno in action. This figure shows (a) the setup with the overhead camera and (b) a screenshot of the host interface during topology inference

result, the relational structure is inspectable directly in the physical model rather than only on a screen.

The overhead capture setup consists of L-shaped brackets to hold a smartphone over a workspace. This allows continuous observation of the workspace and identification of the tags. The phone is connected to a machine running the host application to allow live image capture. (Fig. 2 (a))

The final component of the toolkit is the host application running on a consumer-grade laptop. This displays the inferred topology to the user and allows for export to KG data standards (Fig. 2 (b)).

3.2. Usage scenario

A user works with ArcUno in three phases. In the *calibration* phase, the operator places a known reference artifact under the camera so the system can recover the scale and tolerances of the working area (Section 4.3). In the *registration* phase, the operator introduces the node and edge pieces and binds each tag to its meaning: each node tag is assigned a concept, for example, “User”, and each edge is assigned a relation together with its direction, for example, “belongs to”. After registration, the system knows which tags denote nodes and which denote edges, and what each represents. In the *construction* phase, the user assembles and rearranges the model freely while the system senses the arrangement, committing observations under operator control. During construction, the host displays the inferred graph and a camera overlay so the operator can compare the sensed topology with the physical arrangement. Export is handled after inference, as described in Section 4.6.

4. Implementation

The source code and CAD models for ArcUno, as well as additional figures and documentation including a demo video, are available in the project repository: <https://github.com/robo-monk/ArcUno>.

4.1. Physical components and construction

The nodes, edges, tags, and labels are all 3D-printed. The ArUco tags are generated programmatically and printed in place using multi-material printing, so that the dark marker pattern and its light background are produced in a single print rather than applied as stickers. Each tag terminates in a pill-shaped plug that mates with the obround sockets on the nodes and edges; the friction fit holds the tag flat and flush while still allowing it to be swapped. Because the same socket geometry is shared by machine tags and human-readable labels, the two are interchangeable on any artifact.

4.2. Capture and transport

The *capture rig* is an L-shaped structure that holds a smartphone roughly 50–60 cm above the desk, looking straight down; a 3D-printed light bar, modelled on a monitor light bar, normalises illumination across environments and yields a usable working area of about 1.2×0.8 m. The phone serves as the camera. On the Android side, capture begins automatically once camera permission is granted and then streams full-resolution JPEG frames continuously. Frames are transported over a USB-tethered TCP connection rather than over Wi-Fi: the phone connects to a local port on itself, and `adb reverse` forwards that socket over the USB cable to a Python receiver on the host. Each frame is sent using a small custom binary protocol—an ARC1 header carrying a sequence number, timestamps, frame dimensions, payload length, and JPEG bytes. On the host, OpenCV detects the ArUco tags and estimates their pose.

4.3. Calibration

The calibration phase establishes the geometric parameters that the rest of the pipeline depends on, using a single straight calibration strip of five collinear ArUco tags that doubles as a ruler. The system examines every five-tag combination currently in view, fits a line through the tag centres, and selects the combination with the smallest residual-to-span ratio; the five chosen tags are then ordered along that line and assigned the roles A, B, C, D, and E. These roles encode three quantities. The central tag, C, represents a node tag. Its orientation, snapped to the nearest 90° relative to the strip axis, defines the reference orientation of a node tag, and the vector from C to the node centre, expressed in C's local frame, defines the offset from a node tag's centre to the geometric centre of its node. The outer pair, A and E, defines edge length. The mean of the A–B and D–E distances gives the expected separation between the two tags of a complete edge piece, which sets the tolerance used later to pair an edge's two endpoints. The sample variance of the tag positions across the captured calibration frames feeds an estimate of measurement noise that scales the radial, endpoint-acceptance, and edge-pairing tolerances, so that noisier capture conditions automatically relax the thresholds.

4.4. Registration

The second phase binds the tag identities to domain meaning. The operator presents the node pieces under the camera and assigns each node tag to a concept, then presents the edge pieces and assigns each edge to a relation. Because an edge carries two tags, its direction is captured at the same time, fixing which endpoint is the start and which is the end of the relation. By the end of registration the system holds a mapping from every tag to either a node concept or a directed edge relation.

4.5. Topology inference

During construction, the host reconstructs the graph from each committed frame. The detected tags are first classified as node or edge tags using the registration mapping. For each node tag, the node's geometric centre is recovered by applying the calibrated offset and reference orientation to the tag's measured pose. The two tags of each edge are paired using the calibrated edge-length separation, and each edge endpoint is then attached to the nearest node whose centre lies within the connection radius; the start/end ordering established at registration gives the relation its direction. The result is a directed, labelled graph in which nodes carry their assigned concepts and edges their assigned relations. Because the markers lie flat on the desk, they are detected robustly; brief hand occlusions are smoothed by the system over time to prevent flickering.

4.6. Output and synchronisation

The host maintains the current model as a directed, labelled graph whose nodes and edges correspond to the registered tags. The live graph view and camera overlay act as a real-time communication layer between the tangible arrangement and its digital representation, showing how physical changes are interpreted as graph structure during construction. The graph can be exported as RDF/Turtle or JSON.

5. Discussion

Initial user studies with Knowledge Engineering experts (n=6) show that ArcUno successfully can capture a variety of models produced by the participants. A mean score of 84.5 on the System Usability Scale indicates good usability with users reporting positively on the tangibility and the ease with which actions could be reversed, while reporting negatively on some false affordances of the tool (different colors were interpreted as carrying meaning). Future work includes design of additional sockets for (replaceable) human-readable labels, in addition to the current fiducials. Also, the toolkit can be extended to use semantics explicitly, communicating ie. logical inconsistencies to the user. The current ArcUno toolkit demonstrator provides an extensible framework and experimentation platform that can contribute to bridging the gap between the physical and digital in knowledge modelling.

Declaration on Generative AI use. For this work, the authors used OpenCode (with the open-source GLM-5 and GPT-5 series models), Claude Opus series models, and Vicad (a programmatic CAD tool developed during this project, used together with an LLM) in order to: assist with software development, structure the manuscript, check grammar and spelling and generate the CAD script that produced the fiducial markers. The authors take full responsibility for the the paper's content.

Ethics and Environmental footprint. This study raises no specific ethical concerns. Environmental impact was limited to the material and energy usage associated with the 3D printing process.

Acknowledgements. We thank Marcello A. Gómez-Maureira and the VU-UTwente Smart coalition.

References

- [1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, *ACM Computing Surveys (Csur)* 54 (2021) 1–37.
- [2] J. H. Larkin, H. A. Simon, Why a diagram is (sometimes) worth ten thousand words, *Cognitive Science* 11 (1987) 65–100.
- [3] M. Scaife, Y. Rogers, External cognition: How do graphical representations work?, *International Journal of Human-Computer Studies* 45 (1996) 185–213.
- [4] H. Ishii, B. Ullmer, Tangible bits: Towards seamless interfaces between people, bits and atoms, in: *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 1997, pp. 234–241.
- [5] E. Hornecker, J. Buur, Getting a grip on tangible interaction: A framework on physical space and social interaction, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 437–446.
- [6] S. Price, A representation approach to conceptualizing tangible learning environments, in: *Proc. of the 2nd International Conference on Tangible and Embedded Interaction*, 2008, pp. 151–158.
- [7] P. Marshall, Do tangible interfaces enhance learning?, in: *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, 2007, pp. 163–170.
- [8] S. J. B. A. Hoppenbrouwers, H. A. Proper, T. P. van der Weide, A fundamental view on the process of conceptual modeling, in: *Conceptual Modeling*, volume 3716 of *LNCS*, 2005, pp. 128–143.
- [9] J. D. Novak, A. J. Cañas, The Theory Underlying Concept Maps and How to Construct and Use Them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008.
- [10] S. Oppl, C. Stary, Effects of a tabletop interface on the co-construction of concept maps, in: *Human-Computer Interaction – INTERACT 2011*, volume 6948 of *LNCS*, 2011, pp. 443–460.
- [11] Y. Kitamura, Y. Itoh, T. Masaki, F. Kishino, Activecube: A bi-directional user interface using cubes, in: *Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, 2000, pp. 99–102.
- [12] M. Kwak, G. Niezen, B. J. J. van der Vlist, J. Hu, L. M. G. Feijs, Tangible interfaces to digital connections, centralized versus decentralized, in: *Transactions on Edutainment V*, volume 6530 of *LNCS*, 2011, pp. 132–146.