

OntoSketch: Towards Digital Sketching as a Tool for Creating and Extending Ontologies for Non-Experts

Marius Brade
Chair of Media Design,
Technische Universität Dresden,
Germany
marius.brade@tu-dresden.de

Florian Schneider,
Angelika Salmen
SAP Research, SAP AG
69190 Walldorf, Germany
florian.schneider@sap.com
angelika.salmen@sap.com

Rainer Groh
Chair of Media Design
Technische Universität Dresden
Germany
rainer.groh@tu-dresden.de

ABSTRACT

This paper introduces a sketch-based interaction concept called OntoSketch, enabling non-experts to create and extend formally represented knowledge. Contrary to current editors, OntoSketch is based on lightweight pen and paper-like interactions, including a visual vocabulary that supports non-experts in modeling knowledge about specific domain information. The visualization concept is based on previous work as well as insights from a user study with 20 participants. OntoSketch enables the user to work in three views with his data: created domain examples, related or emerging abstract concepts and a combination of both. This allows for reflection while creating formalized knowledge. An export function allows a non-expert to generate an ontology out of his sketch-based work, without the necessity to understand the concept of ontologies in detail. Our approach is demonstrated via a prototypical tablet-PC application.

Categories and Subject Descriptors

H.5.2. [Information Interfaces and Presentation]: User Interfaces – Graphical User Interfaces (GUI)

General Terms

Documentation, Human Factors, Management.

Keywords

Graphical User Interfaces, Knowledge Construction, Knowledge Representation, Linked Data Management, Non-Expert Users, Ontology, Reflection-in-Action, Semantic Data, Sketching Interface, Topic Networks, Visual Sensemaking.

1. INTRODUCTION

To support sharing and reuse of formally represented knowledge, Gruber [14] developed a mechanism for defining specifications that are portable between different representation systems. This mechanism consists of a common vocabulary representing shared domain knowledge. Such a vocabulary is called *ontology* and includes definitions of concepts, relations, functions, and further objects. This standardized system-independent form can be translated into specific representation languages (for more details see section 2.6) allowing researchers and other knowledge workers to share and reuse domain knowledge, while keeping the computational benefits of specialized implementations [14].

Especially in the web, which is characterized by interoperability of domains and services, the number of ontologies is constantly increasing [4, p.89]. Ontologies have also been introduced into the industries, however, with growing complexity of domains and processes creation and maintenance are cost intense. Current ontology tools require experts regarding knowledge modeling which in turn are dependent on experts providing the domain knowledge.

Our approach, called *OntoSketch*, seeks to simplify this process by providing a tool suitable for non-experts. In this paper we define *non-experts* as users being experts in specific domains (e.g. printing), but having no experience in using ontologies. OntoSketch supports visual representation, sketch-based creation and manipulation of domain knowledge. Non-experts can reflect, create and export formalized knowledge about their specific domain.

Our approach is motivated by a use case of the EU research project ComVantage [9], which addresses interorganizational web based collaboration. A set of companies are part of a virtual collaboration network in the fashion industry. Small and micro-companies enable customized products by combining different skill profiles. To enable a successful production process, well-defined specifications need to be implemented. An ontology is used to specify the distributed infrastructure for virtual enterprises based on formal definitions of their structure, processes, and products. Single enterprises joining and collaborating in the network need to describe their profiles using and extending this ontology. OntoSketch shall reduce and help dealing with complexity during these abstract modeling procedures without the need to fully understand ontologies. For specific cases a central expert in the collaboration network may be consulted to finalize a created or extended ontology. Our approach addresses especially small and medium sized companies, since they cannot afford costs and resources to build up the relevant expert knowledge.

OntoSketch is influenced by various disciplines, being described in section two. Section three provides details about our design decisions including results from a user study. Section four specifies the concept, and section five its implementation.

2. RELATED WORK

This section provides an overview about related approaches concerning OntoSketch. Areas of specific interest are sketching, sensemaking, external representations, and modeling ontologies.

2.1 Sketching

In human computer interaction considerable research has shown that sketching is the most immediate way of representation.

According to Johnson et al. [16] it can be seen as an iterative process including problem-framing and exploring possible solutions. They state: "Sketching allows people to visually represent ideas quickly, without prematurely committing to decisions. A sketch is not a contract: it is a proposal that can be modified, erased, built upon. The rough look of hand-made sketches suggests their provisional nature" [16 p. 3]. Tversky [30] analyzed sketching from the perspective of psychology. Results show they reveal a user's conception of a domain. Users new to a given topic extract structural information about spatial relations among elements. Experts further infer detailed functional information.

2.2 Digital Tools for Sketching and Writing

After the pioneer work on digital sketching tools from Sutherland [29] a great number of approaches exist regarding sketching with computational support. Commercial tools like the InkWriter [2] offer functions to support handwriting with real-time recognition, pen-gesture support, and complex text editing functions. This and other research approaches [5] distinguish between digitalization of text and drawings. Applications like InkKit [24] prepare informal sketches through abstraction for later formalization as diagrams.

To our knowledge there is no digital sketching support for modeling ontologies. Accordingly, we use sketching for the iterative process of problem-framing (e.g. a domain expert uses specific examples to find patterns and underlying concepts), as well as exploring possible solutions (e.g. describing and modifying abstract concepts and relations).

2.3 Sensemaking

Fischer et al. [12] emphasize the importance of structure during problem-framing and exploring solutions. They studied iterated knowledge maps users created while solving a collaborative sensemaking task. Furnas and Russell [13] define a *sensemaking* task to include a process involving collecting and organizing data in order to create sense. Thereby users often use a quite simple structure to solve a given problem. The authors state: "the process is sometimes ill-defined, iterative and complex: information retrieval, organization and task re-definition all interact in sometimes subtle ways". The sensemaking process is inherent to many information tasks and also applicable for creating and extending formalized knowledge about a domain.

Several studies focus on models of sensemaking employed by users [1, 12, 26]. Pirolli and Russell present a comprehensive overview about the topic [25]. Our approach follows the model proposed by Qu and Furnas [26] to support the users' work with modeling ontologies. It can be summed up into three essential stages:

- learning concepts/facts
- finding relationships
- forming categories and high-level connections

2.4 Techniques for External Representation

When studying situations, diagrams, illustrations, instructions and problems, people create representations helping to make sense of these. Kirsh [19] presents a comprehensive overview of the relations between interaction, external representation and sensemaking for further reading. Creating external representations can be done in various ways [10], e.g. concept maps, conceptual diagrams, semantic maps, topic maps [22], and mind maps being the most prominent ones. Eppler defines a mind map as "a

multicolored and imagecentred, radial diagram that represents semantic or other connections between portions of learned material hierarchically" [10]. The typical application context is personal note taking and reviewing with major advantage to provide a concise overview.

Inspired by these approaches, we aim at providing a reliable overview concerning the modeling task while at the same time enable the user to specify detailed information.

2.5 Tools for Sensemaking using External Representation

Digital tools supporting sensemaking tasks apply different kinds of external representations. A sketch-based representation tool is SketchViz [7]. It enables knowledge workers to dynamically build knowledge maps using lightweight pen and paper-like interactions on a tablet PC, and it recognizes objects, text, and relations as digital entities. A system employing a visualization inspired by nature is BrainDump [6]. Affordances and behavior of fluids and soap bubbles are adopted for easier information organizing during complex web search tasks. Objects can be connected to uniquely shaped groups with the possibility to relate these objects in different intensities via respective visualizations. Diagram based tools like Symbik [35] or iMapping [15] aim to combine the strengths of established diagramming techniques with interaction techniques like deep zooming and panning.

This paper offers a lightweight interactive method for visualization, as well as sketch-based creation and extension of ontologies via graphical entities. We address first sketches up to elaborated constructs. Paper like affordances and visualizations shall lower the cost of basic operations with a digital system often being overlooked in current sensemaking systems [26].

2.6 Modeling Ontologies

Prerequisite to support interactive visual modeling of ontologies is to understand the underlying mechanisms.

2.6.1 Naming Ontology Constructs

Contemporary ontologies share many structural similarities, regardless of the representing language. We define the following wording for ontology constructs to specifically address non-experts:

- *real elements* – also known as individuals, instances or objects
- *concepts* – classes that are also called set, type, sort, category, or kind
- *relations* – also called relationships or terms, they are ways in which concepts and real elements can be related to one another

In most languages there are many more constructs¹, making the work for non-experts very complex. By reducing the complexity of ontologies, presenting only a limited set of constructs to a user as listed above, we want to ease the modeling process.

2.6.2 Languages for Modeling Ontologies

Relations between real elements and the two ends of such a relation are usually referred to as triple, which is associated with a Unified Resource Identifier. This allows different data to be mixed, published, and shared across diverse applications [32]. The

¹ <http://www.w3.org/standards/semanticweb/ontology> last visited June 19th 2013.

so called *Resource Description Framework (RDF)* is a general-purpose language for representing information in the World Wide Web with the aim to extend its linking structure [32]. The most important thing RDF defines is a predicate called "rdf:type". This is used to say that things are of certain types. The *RDF Schema (RDFS)* [33] builds on the formal concept of RDF. It provides an additional abstraction layer for defining hierarchies in concepts and extended functionalities to describe the meaning of a relationship or a concept in human readable text. This is a schema. More vocabulary to describe relations and concepts is added by the Web Ontology Language (OWL), which builds on RDF and RDFS [31]. OWL adds semantics to the schema. This way it can be indicated that "If A isRelatedTo B" then this implies "B isRelatedTo A" for example. Further, users get the ability to say two things are the same, which means data from multiple sites can be joined. This is called *Linked Data* [3]. We use OWL in our approach.

2.7 Tools for Modeling Ontologies

Writing ontologies with a text editor is possible, but not very practicable. IsaViz [23] provides an approach for browsing and authoring models represented as graphs based on RDF. Users can work directly with a visual representation of a given ontology.

Most graphical tools like Protégé [27], OntoStudio [34], and Swoop [17] use a frame based layout and menu trees for adding items and edit their relations. Views for concepts, real elements and relations are separated from each other. Words representing concepts, real elements or relations need to be mapped to each other by clicking through dialogs, which can be cumbersome for the user. Ontologies may also be visually displayed using plug-ins such as OntoGraf [11] for Protégé. Concepts and real elements are represented as nodes of a graph and are connected by arrows. Relations are shown as an arrow with a dotted line. Each relation has its own color and a description or help text is available. Other approaches visualize ontologies using a spatial layout, for example a tree map [21] or in a hierarchical multi-perspective to enhance how people browse and explore complex information spaces [28]. Katifori et al. [18] and Lanzenberger et al. [20] give a detailed overview on current existing representations.

To sum up, ontologies are powerful, complex, multidimensional, and abstract specifications, but quite infeasible for non-experts as they are not familiar with the wording and functionalities. Systems and concepts addressing non-experts are currently missing. Therefore we aim to enable non-experts to model their domain knowledge without a deep understanding of ontologies.

3. DESIGN DECISIONS

In this section design decisions, based on insights from related approaches as well as from an initial user study regarding creation and interpretation of visual relations, are explained.

3.1 Use Case

The use case regarding OntoSketch is related to the EU research project ComVantage [9] as described in section 1. In this network companies provide customized products by combining their different skill profiles. To enable a successful collaborative production process, well-defined specifications need to be implemented. Ontologies specify the distributed infrastructure for the virtual collaboration network based on formal definitions regarding structure, processes, and products. When a new company joins the network, the collaboration based ontology

needs to be adapted, which is generally unfeasible for small and medium enterprises.

OntoSketch shall enable non-experts in a lightweight manner to extend existing ontologies with respect to their needs.

3.2 Constraints Regarding Ontologies

Unlike experts defining abstract concepts, we focus on real elements that are familiar from the user's world. Additionally, the complexity is decreased by reducing the functional scope to relevant processes for the non-expert (see section 2.6.1)

3.3 Requirements

The approach involves a set of requirements as listed below:

- [R1] Import of any ontology (in format of OWL or RDFS).
- [R2] Extension of this ontology, whereas deletion and change of imported parts are prohibited ("Linked Data principle" [3]).
- [R3] Available OWL constructs [31] in OntoSketch: owl:Class, rdfs:subClassOf, owl:ObjectProperty, owl:NamedIndividual.
- [R4] Constraint: The creation of new relations is restricted to real elements (otherwise domain and range restrictions limit the possibilities to work with real elements and the complexity increases unnecessarily for non-experts).
- [R5] If two concepts are related in the ontology and their instantiated real elements are being related by the user as well, the system proposes a selection of possible types of relations.
- [R6] Export of the extended ontology: All created (sketched) items are exported as a new ontology with references to the items from the originally imported ontology.

OntoSketch is built upon SketchViz [7], providing all functions regarding the support of digital sketch-based sensemaking.

3.4 User Study Regarding Visualization

Most ontology visualizations are graph based visualizations due to the relations between the knowledge items (described in section 2.7). Yet, there are various ways to create and interpret visual relations. Therefore we conducted a preliminary user study to identify the most suitable ones for non-experts.

3.4.1 Hypothesis

In current diagram techniques used in computer science and related fields (e.g. UML or Entity Relationship) relations are represented by arrows, pointing from sub to main concept. Our first hypothesis is that the directions of these arrows are not intuitive. Non-experts interpret the arrows pointing from main to sub concept.

In addition we wanted to study alternative forms of visualizing relations with our second hypothesis being that arrows work best to visualize directed relations.

3.4.2 Method

The user study consisted of two parts, focusing 1) on methods for sketching relations, and 2) on the interpretation of a prearranged set of visual relations.

Since the participants had diverse expertise backgrounds, the ComVantage specific use case was substituted by a common example reflecting family relationships.

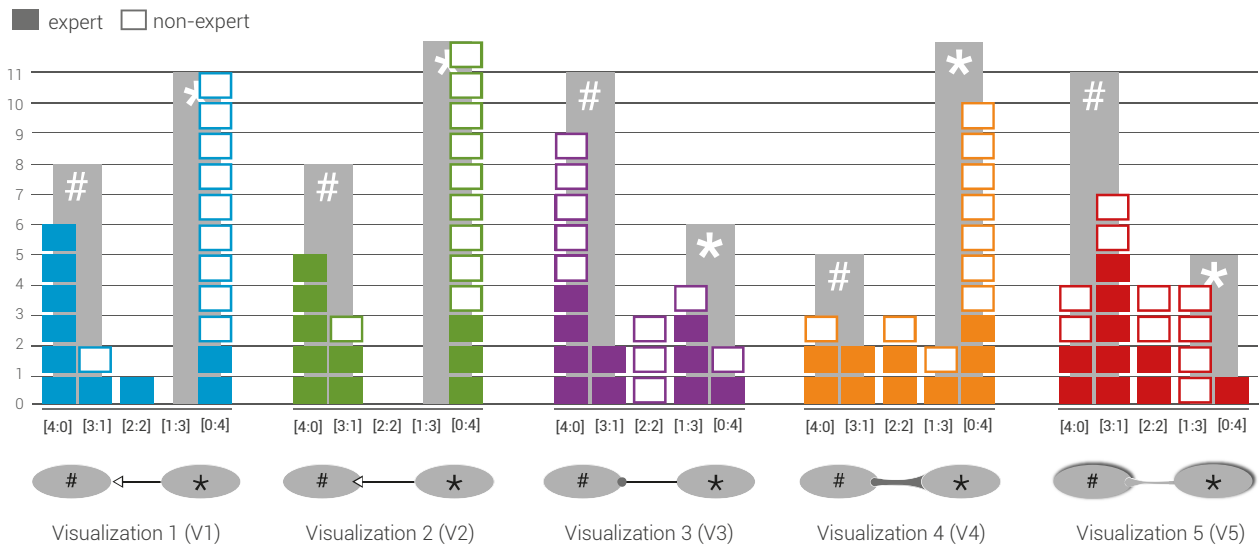


Figure 1: Results for each type of relation between concepts. V1, V2 performed best for non-experts, followed by V4.

After a short introduction, participants were given a handout with prepared tasks and a questionnaire. Each participant worked autonomously and had as much time as needed to complete the tasks. During the user study participants were observed and the observer took notes. Participants were assured that there is no wrong answer in sketching or interpreting given relations, they should pick the most suitable one.

3.4.3 Participants

20 voluntary participants (8 female, 12 male) between 19 and 75 years of age took part in the user study. They were split in two groups of ten: one group had no knowledge in computer science or in modeling abstract concepts, whereas the other group consisted of computer science graduate students, thus being skilled in defining abstract models.

3.4.4 Part 1: Sketching Methods and Artifacts

During this part of the study participants used pen and paper and were asked to bring a set of words (child, person, mother, son, father and daughter) in a hierarchical order and visualize inheritance where applicable.

3.4.5 Results Part 1

Figure 2 shows a set of example sketches. All sketches from both target groups reflected spatial orientation either from top to bottom, or from left to right expressing hierarchical order.

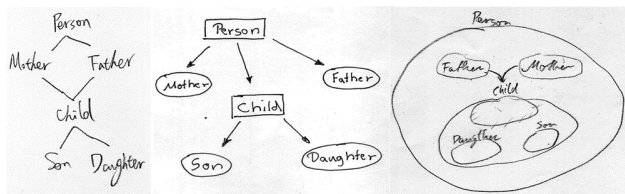


Figure 2: Set of sketches participants used to describe relations.

Most participants expressed un-specified (non-hierarchical) relations by simple strokes (30% experts, 70% non-experts). Experts drew more directional arrows than non-experts (40% experts, 20% non-experts). 15% of all participants (2 experts, 1

non-expert) used spatial diagrams (like VENN diagrams). One participant created small icons next to the word. Expert sketches showed a strong similarity to UML, Entity Relationship or Flow-Chart diagrams.

3.4.6 Part 2: Interpretation of Visual Relations

During this part of the study participants were given 20 tasks (example task see Figure 3) comprising of five different types of visualizations (see bottom of each diagram of Figure 1). Each depicted in four orientations (top to bottom, right to left, bottom to top and left to right). In every task participants should identify main and sub item within a predefined relation (example see Figure 3). The handout consisted of single sided pages with one task per page, presented in a mixed order. Participants were not allowed to go back to a previous page.

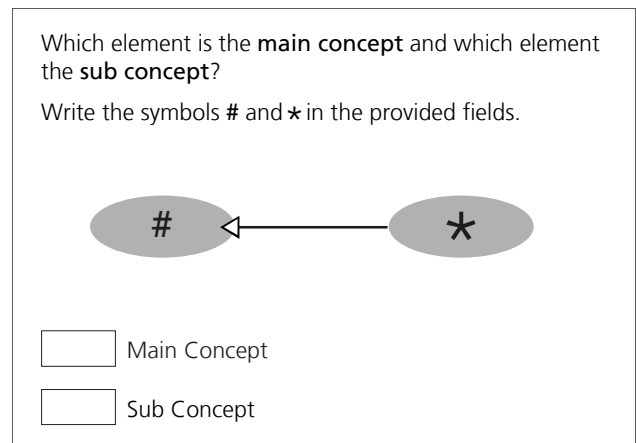


Figure 3: example task to interpret a visual relation between two concepts named as # and *.

3.4.7 Evaluation

The results were analyzed for every type of relation. For each participant five possible results were written in the notation [to_main:from_main], to_main and from_main are represented by numbers from 0 to 4 indicating how many orientations (top to

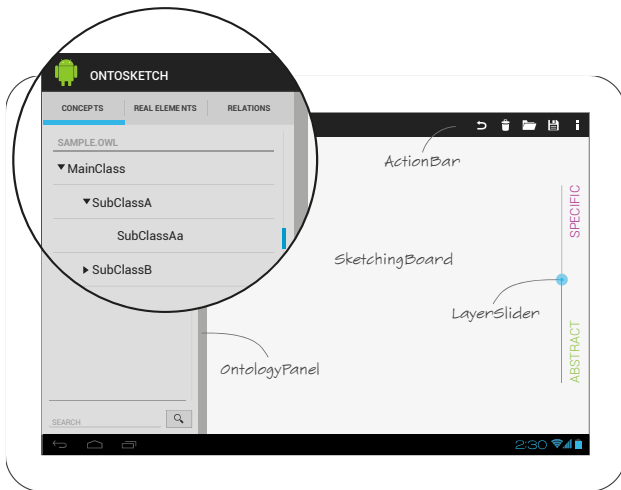


Figure 4: User interface of OntoSketch based on SketchViz [7]

bottom, right to left, bottom to top and left to right) were interpreted the same way:

- [4:0] when # was interpreted as main concept or [0:4] when * was interpreted as main concept, constant result for all four orientations, no deviation
- [3:1] or [1:3] when including one deviation respectively
- [2:2] when no clear difference in interpretation exists

Counting the number of results [4:0] and [3:1] or [0:4] and [1:3] over all participants allows to see a tendency for each type of relation (see Figure 1, grey bars).

3.4.8 Results Part 2

Figure 1 shows the results of part 2 of the user study, where filled rectangles symbolize experts, empty ones non-experts. Each type of relation is shown at the bottom including a label with a number (e.g. V1). Experts and non-experts are arranged by conditions (e.g. [4:0]) for each type of relation.

The interpretation of arrows (see Figure 1, V1 and V2) by experts and non-experts is opposing, which confirms our first hypothesis. Experts interpret these relations from sub to main concept (bottom up) and non-experts the other way round from main to sub concept (top down).

Confirming our second hypothesis the specified relation types showed no indicators for further visual forms working better than arrows. Concerning the expert group, both arrow visualizations performed best in condition [4:0] (see Figure 1, V1 and V2), followed by V5. V3 and V4 showed no clear difference in interpretation. Regarding the non-expert group V1 and V2 clearly performed best in the condition [0:4] (see Figure 1, V1 and V2) followed by V4. V3 and V5 did not reveal such a clear difference.

3.4.9 Discussion

Expert sketches from first part of the study showed a strong similarity to diagrams such as UML which meets our expectations since this group is experienced with abstract modeling tools. The second part of the study confirmed our hypothesis, that non-experts interpret arrows contrary to experts. Therefore, in OntoSketch relations are modeled from main to sub concept.

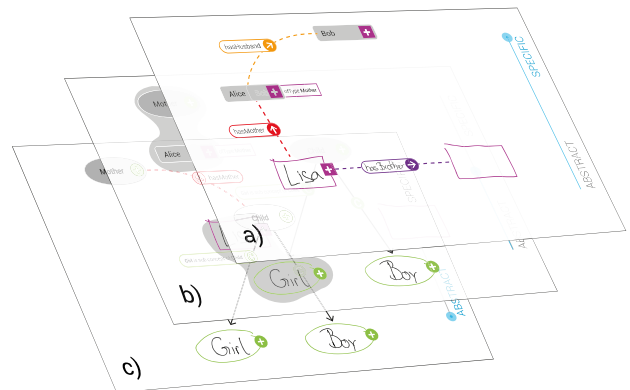


Figure 5: Layer concept: a) real elements, b) all elements except relations, c) abstract concepts.

An interesting question for future work addresses the different results of V4 with experts (distinct) and non-experts (diverse) (see Figure 1, V4). Exploring further different types of visual relations will benefit the validity of the results.

4. CONCEPT OF ONTOSKETCH

This section describes the concept of OntoSketch, based on the design decisions defined in the previous section. Where applicable, links to the requirements are given (in format: [RX]).

4.1 User Interface and Workflow

The following section explains relevant parts of the user interface in addition to a brief workflow.

4.1.1 SketchingBoard

Building upon the existing prototypical Android application SketchViz [7] enables panning and zooming, as well as free sketching with handwriting and gesture recognition. Users can construct a representation of their mental models quite similar to pen and paper. The working area for creating and extending an ontology is called SketchingBoard (see Figure 4).

4.1.2 Action bar

The ActionBar (see Figure 4) holds the functionalities for importing an existing ontology [R1] for extension [R2] as well as exporting such extended ontologies [R6]. Moreover, the ActionBar comprises the standard actions undo, delete, save a session to continue work later, open a session, and settings. The latter include colors, left hand / right hand usage of OntologyPanel, and file export type.

4.1.3 OntologyPanel

The OntologyPanel displays the imported ontology (see Figure 4), including the tabs *Concepts* (a tree visualization shows concepts and sub-concepts), *Reals Elements*, and *Relations* as well as a search function. The user can choose any of the elements in the OntologyPanel and drag them onto the SketchingBoard to operate on [R3]. To support the process of creation, imported objects appear formalized while new items are displayed as sketched elements. The OntologyPanel can be minimized to enlarge the working area.

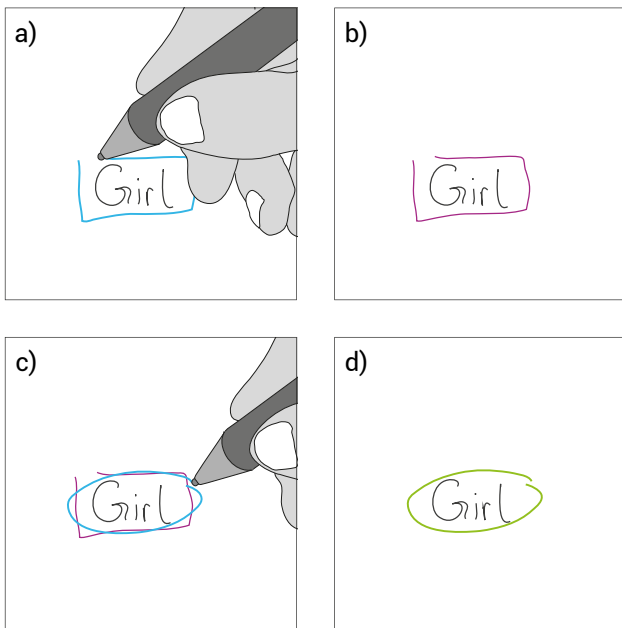


Figure 6: Items can be transformed by overwriting.

4.1.4 LayerSlider

A layer concept distinguishes between real elements and abstract domain concepts. The LayerSlider provides a smooth visual transition between the layers, focusing on either specific elements, abstract concepts or a merged view. This allows for different perspectives while working with real world elements and emerging abstract concepts (see Figure 4 and Figure 5). To avoid visual clutter the merged view only displays concepts and real elements without relations.

4.2 Interaction and Visualization

The main contribution of OntoSketch is the combination of formal elements with sketched information from doodling to handwriting. Initially sketched information can be transformed into formalized information using a specific visual vocabulary and be reused.

4.2.1 Visual Vocabulary

We developed a simple visual vocabulary to ensure a clear distinction between concepts, real elements and relations (cf. Figure 7 and Figure 8):

- Color distinction: concepts are green, real elements are magenta, and relations between the same types are colored respectively. Relations between concepts and real elements are visualized by a gray shape in the background.
- Form distinction: concepts are represented by round to oval shapes, real elements by rectangles, and relations by arrows, dotted lines, or groups with a common outline.
- Optical distinction between formalized (imported / reused items) and sketched items: Formalized elements have a gray background and labels are displayed in printed font (see Figure 8, c, d). Self-drawn objects remain sketched, even if the type is changed by overwriting (see Figure 6).

4.2.2 Interaction Regarding an Ontology

Sketching round or rectangular shapes creates concepts or real elements. They may hold words or sketches or remain empty,

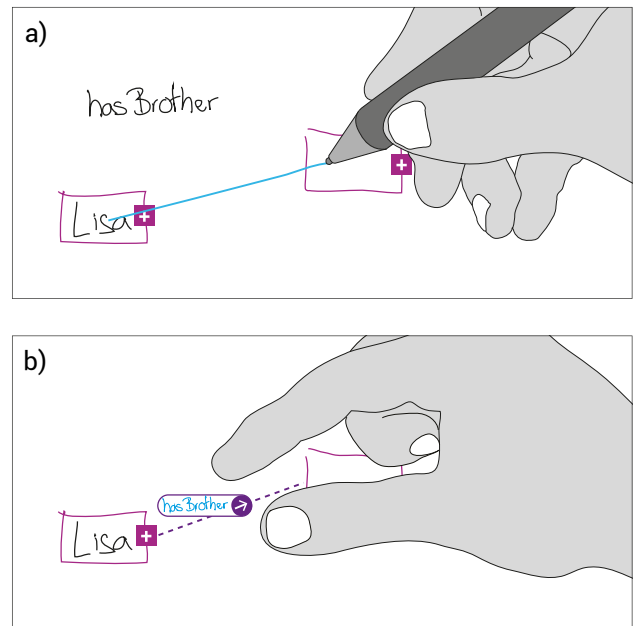


Figure 7: Example for creating relations: (a) drawing a stroke to create a relation, (b) assign text and sketches.

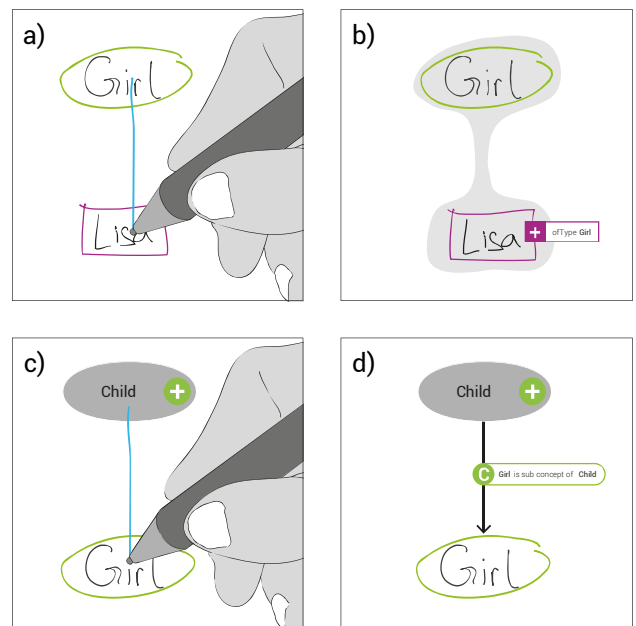


Figure 8: Strokes create instantiations (a, b) or relations, e.g. sub- or super-concept (c, d), dependent on the type of connected elements.

functioning as placeholders. Interaction with items remains consistent, no matter if they are formalized, sketched, or placeholders.

Figure 7 shows an example use case: (a) in a first step the user writes down that Lisa has a brother. Then the user wants to structure this information and creates two rectangles, which instantly are recognized as real elements by the system and given the respective color. To support the process of sensemaking, items

may remain empty. Drawing a line between the two rectangles (b) formally defines the relationship which is further specified by dragging the text “has Brother” onto the line.

Moreover, OntoSketch supports the following relations [R4] between sketched and imported elements (see Figure 8): concept to real element (instantiation), concept to concept (sub- or super-concept) and real element to real element (see Figure 7). The latter can either be new relations not existing in the original ontology or predefined relations with assigned properties if the real elements are instantiated [R5]. Furthermore, a filtering function can be used to look up connections from a selected ontology item in the ontology panel.

4.2.3 Assistance

OntoSketch offers assistance while the user works with it. Icons on active items and relations offer further explanations on demand (e.g. see plus-sign on items in Figure 7 and Figure 8). Based on predefined relations between concepts in the ontology OntoSketch suggests derived relations for linking real elements. Furthermore, notifications are provided after successful interactions, e.g. word recognition, pattern matching when creating objects, linking of objects, and error messages.

5. IMPLEMENTATION

The implementation of OntoSketch is based on SketchViz [7], providing a zoomable user interface, handwriting, and gesture recognition. The prototype is implemented on a Samsung Galaxy Tab 10.1N WiFi GT-P7511 running Android 4.1 (minimum requirement for OntoSketch is Android SDK 3.0). Based on the Android version² of the Jena framework [8] which supports RDF export, we created our own implementation to realize OWL export as well. The application in use is shown in Figure 9.



Figure 9: OntoSketch implemented on an Android based tablet-PC enabling pen and multi-touch interaction.

6. CONCLUSION

This paper introduces a sketch-based ontology editor named OntoSketch. It supports non-experts in modeling domain knowledge using a visual vocabulary. This vocabulary is based on ontologies including concepts, relations, and real elements.

OntoSketch allows non-experts to create and extend ontologies. Based on real elements out of their specific domain users can find reoccurring patterns and define abstract concepts describing these. Existing ontologies can be imported, explored and adapted respectively. While being in the process of creation, self-defined

entities and relations appear sketched indicating the transient state. Final models can be exported as incremental ontology, thus being able to be imported again as formalized representations. This way OntoSketch enables non-experts to create ontologies without having to fully understand and study this complex technique. To our knowledge OntoSketch is the first approach trying to simplify modeling formal knowledge with graphical means.

The visualization of relations in OntoSketch is based on the results of a user study with twenty participants, revealing that arrows are interpreted in a reversed way by non-experts as opposed to in modeling techniques such as UML diagrams.

7. FUTURE WORK

OntoSketch envisions combining freedom and flexibility in sketching with formalization of structured knowledge. Finding appropriate visualizations and interactions for this field is challenging and requires further research. A next step will be to evaluate OntoSketch with an appropriate use case.

8. ACKNOWLEDGMENTS

The authors would like to thank Jan Hladik, Steffen Buzin and Tobias Münch for the very helpful discussions. On behalf of Marius Brade, this research project (contract no. 080951799) is funded by the European Social Fund and the Free State of Saxony as well as SAP Research. On behalf of Angelika Salmen, the research leading to these results was partly funded by the European Community's Seventh Framework Programme under grant agreement no. FP7-284928 ComVantage.

9. REFERENCES

1. Abraham, A., Petre, M. and Sharp, H. Information seeking: Sensemaking and interactions. *CHI 2008 Workshop on Sensemaking*.
2. aha! software cooperation (1993). aha! InkWriter Handbook. California.
3. Berners-Lee, T. Linked Data - Design Issues. 2009. <http://www.w3.org/DesignIssues/LinkedData.html> last visited April 4th 2013.
4. Bizer, C. (2009). The emerging web of linked data. *Intelligent Systems*, IEEE, 24(5), 87–92.
5. Blagojevic, R.V., Plimmer, B.E., Grundy, J.C., Wang, Y. 'Building Digital Ink Recognizers using Data Mining: Distinguishing Between Text and Shapes in Hand Drawn Diagrams', IEA-AIE 2010, Spain, July - July 2010.
6. Brade, M., Brändel, C. and Groh, R. 2012. BrainDump: taking fluid interaction literally. *In Proceedings of the AVI '12*, ACM, New York, NY, USA, 790-791. DOI=<http://doi.acm.org/10.1145/2254556.2254716>
7. Brade, M., Brändel, C., Salmen, A., and Groh, R. SketchViz: a sketching interface for domain comprehension tasks illustrated by an industrial network use case. *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies - i-KNOW '12*, ACM Press (2012).
8. Carroll, J., Dickinson, I., and Dollin, C. 2004. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (WWW Alt. '04)*. ACM, New York, NY, USA, 74-83. DOI=<http://doi.acm.org/10.1145/1013367.1013381>

² <http://code.google.com/p/androjena/> (last visited June 19th 2013)

9. Christl, C., Hladik, J., Graube, M., Willfort, R. and Urbas L. Using mobile technology for inter-organizational collaboration and end-customer integration. *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies - i-KNOW '13*, ACM Press (2013).
10. Eppler, M. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information Visualization 2006 Volume 5*, Palgrave Macmillan Ltd. 2006, 202 -210, DOI:10.1057/palgrave.ivs.9500131
11. Falconer, S. OntoGraf - Protege Wiki. 2010. <http://protegewiki.stanford.edu/wiki/OntoGraf> last visited June 10th 2013.
12. Fisher, K., Counts, S. and Kittur, A. 2012. Distributed sensemaking: improving sensemaking by leveraging the efforts of previous users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 247-256. DOI=[10.1145/2207676.2207711](https://doi.org/10.1145/2207676.2207711)
13. Furnas G.W. and Russell D. M. 2005. Making sense of sensemaking. In CHI '05 extended abstracts on Human factors in computing systems (CHI EA '05). ACM, New York, NY, USA, 2115-2116. DOI=[10.1145/1056808.1057113](https://doi.org/10.1145/1056808.1057113)
14. Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Journal of Knowledge acquisition*. Vol. 5, 2 April (1993), 199–220. DOI= [10.1006/knac.1993.1008](https://doi.org/10.1006/knac.1993.1008)
15. Haller, H. and Abecker, A. Designing a Knowledge Mapping Tool for Knowledge Workers. In R. Setchi, I. Jordanov, R. Howlett and L. Jain, eds., *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer Berlin Heidelberg, 2010, 660–669.
16. Johnson, G., Gross, M.D., Hong, J., and Yi-Luen Do, E. Computational Support for Sketching in Design: A Review. *Foundations and Trends® in Human-Computer Interaction 2*, 1 (2007), 1–93.
17. Kalyanpur, A., Parsia, B., & Sirin, E. (2006). Swoop: A Web Ontology Editing Browser. *Web Semantics: Science, Services and Agents on the World Wide Web, Volume 4, Issue 2*, June 2006, Pages 144-153.
18. Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulou, E. Ontology visualization methods - a survey. *ACM Comput. Surv.* 39, 4, Article 10 (November 2007). DOI=[http://doi.acm.org/10.1145/1287620.1287621](https://doi.org/10.1145/1287620.1287621).
19. Kirsh, D. Interaction, external representation and sense making. In Niels Taatgen and Hedderik van Rijn, editors. *Proceedings of the Thirty First Annual Conference of the Cognitive Science Society*, pages 1103--1108, Austin, TX, 2009.
20. Lanzenberger, M., Sampson, J. and Rester M. (2010). Ontology Visualization: Tools and Techniques for Visual Representation of Semi-Structured Meta-Data, *Journal of Universal Computer Science*, vol. 16, no. 7 (2010), 1036-1054
21. Nazemi, K., Breyer, M., & Hornung, C. (2009). SeMap: A Concept for the Visualization of Semantics as Maps. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, et al. (Eds.), *Universal Access in Human-Computer Interaction. Applications and Services (Vol. 5616, pp. 83–91)*. Berlin, Heidelberg: Springer Berlin Heidelberg.
22. Park, J. and Hunting, S. XML Topic Maps: Creating and Using Topic Maps for the Web. *Addison-Wesley Longman Publishing Co., Inc.* 2002. Boston, MA, USA
23. Pietriga, E.: IsaViz: A Visual Authoring Tool for RDF. <http://www.w3.org/2001/11/IsaViz/>. 2007, last visited June 10th 2013.
24. Plimmer, B., and Freeman, I. A toolkit approach to sketched diagram recognition. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1 (BCS-HCI '07), Vol. 1*. British Computer Society, Swinton, UK (2007), 205-213.
25. Pirolli, P., Russell, D.M. Introduction to this Special Issue on Sensemaking. *Human-Computer Interaction, Vol. 26, Iss. 1-2*, 2011. DOI: 10.1080/07370024.2011.556557
26. Qu, Y. and Furnas, G.W. 2005. Sources of structure in sensemaking. In *CHI '05 extended abstracts on Human factors in computing systems (CHI EA '05)*. ACM, New York, NY, USA, 1989-1992. DOI=<http://doi.acm.org/10.1145/1056808.1057074>
27. Stanford Center for Biomedical Informatics Research. Protégé. <http://protege.stanford.edu> last visited June 19th 2013.
28. Storey, M.-A., Noy, N.F., Musen, M., Best, C., Ferguson, R., and Ernst, N. Jambalaya. *Proceedings of the 7th international conference on Intelligent user interfaces - IUI '02*, ACM Press (2002), 239.
29. Sutherland, Ivan E. 1964. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop (DAC '64)*. ACM, New York, NY, USA, 6.329-6.346. DOI=[10.1145/800265.810742](https://doi.org/10.1145/800265.810742)
30. Tversky, B. What do sketches say about thinking. *AAAI spring symposium on sketch understanding*, (2002), 148–151.
31. W3C, World Wide Web Consortium. Web Ontology Language OWL / W3C Semantic Web Activity. 2007. <http://www.w3.org/2004/OWL/> last visited April 4th 2013.
32. W3C, World Wide Web Consortium. RDF - Semantic Web Standards. 2004. <http://www.w3.org/RDF/> last visited April 8th 2013.
33. W3C, World Wide Web Consortium. RDF Vocabulary Description Language 1.0: RDF Schema. 2004. <http://www.w3.org/TR/rdf-schema/> last visited June 10th 2013.
34. Weiten, M. (2009). OntoSTUDIO ® as a Ontology Engineering Ontology Engineering Tools : State of the Art. *Semantic Knowledge Management*, 51–60.
35. Werner, H., Latzina, M., and Brade, M. Symbik—A new medium for collaborative knowledge-intensive work. *Proc. of the International Conference on Education, Informatics, and Cybernetics. International Institute of Informatics and Systemics*, FL, URL: http://www.iiis.org/CDs2011/CD2011IDI/ICEIC_2011/Papers/Pdf/EI924DD.pdf.