Software Galaxies: Displaying Coding Activities using a Galaxy Metaphor

Daniel Atzberger
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

Willy Scheibel
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

Daniel Limberger
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

Jürgen Döllner
Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam, Germany

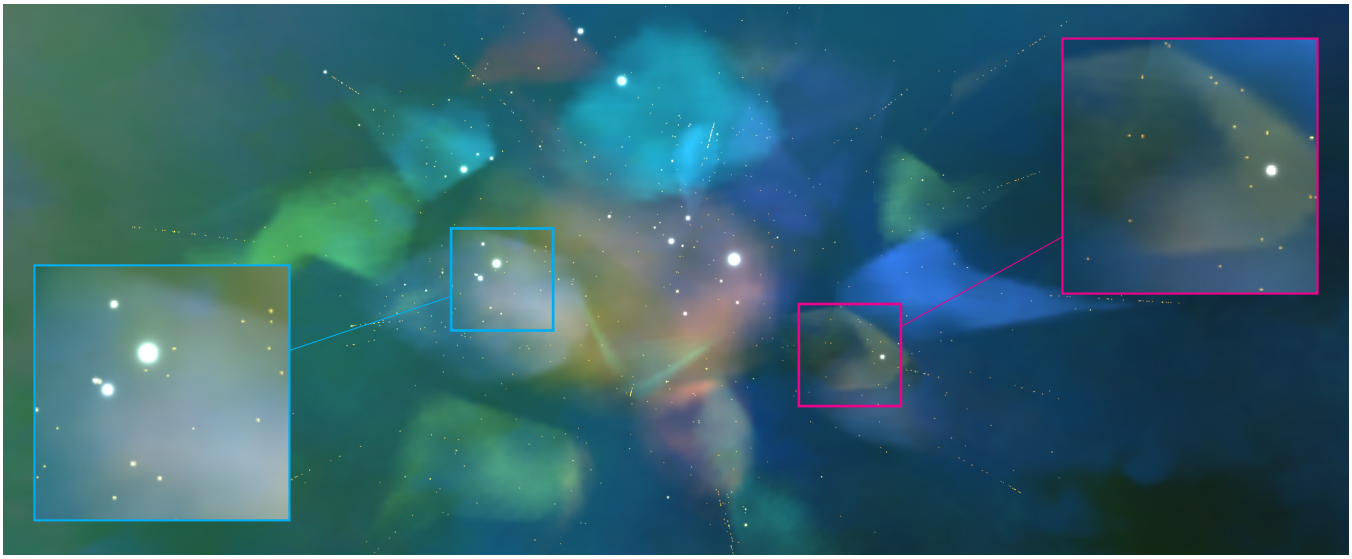

Figure 1: Nebula metaphor applied to the *Software Galaxy*: The left box highlights how documents are displayed as stars of varying intensities, the right box highlights how clusters of documents are visually grouped, using enclosing volumetric fog.

ABSTRACT

Software visualization uses metaphors to depict software system and software development data that usually has no inherent gestalt. The choice of a fitting metaphor for visual display is researched broadly, but deriving a layout based on similarity is still challenging. We present a novel approach to 3D software visualization called *Software Galaxy*. Our layout is based on applying Latent Dirichlet Allocation on source code documents. We utilize a metaphor inspired from astronomy for depicting software metrics for single documents and clusters of documents. Our first experiments indicate that a 3D visualization capturing semantic relatedness can be beneficial for standard program comprehension tasks.

VINCI '21, September 6–8, 2021, Potsdam, Germany

ACM ISBN 978-1-4503-8647-0/21/09.
https://doi.org/10.1145/3481549.3481573

CCS CONCEPTS

• **Human-centered computing** → **Visual Analytics**.

KEYWORDS

Software Visualization, Visualization Metaphor, Topic Modeling

ACM Reference Format:
Daniel Atzberger, Willy Scheibel, Daniel Limberger, and Jürgen Döllner. 2021. Software Galaxies: Displaying Coding Activities using a Galaxy Metaphor. In *The 14th International Symposium on Visual Information Communication and Interaction (VINCI '21), September 6–8, 2021, Potsdam, Germany.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3481549.3481573

1 INTRODUCTION

Software metrics provide a possibility for monitoring large software systems through mining software repositories and measuring aspects of complexity and quality of single source code files. This can be useful for managing maintenance tasks. As this is still a human task, software visualization is used to depict the state or the evolution of software through mapping metrics to the visual variables of 2D, 2.5D, or 3D viusalization techniques, e.g., Software Cities [10],

or Gource [3]. The layout of such a visualization technique is determined by the hierarchical structure of a project, which is helpful for navigating through a system. However in many program comprehension tasks the question of detecting files that implement a similar concept arises. This motivates other approaches, which try to place the points based on their semantic relatedness [1, 7, 9].

This work extends the approach for 2D "semantic layouts" to the third spatial dimension. Our use of visual variables is inspired from the astronomical metaphor proposed by Lanza [8], i.e., we display each file as a star with their own visual variables in space, thus resulting in a galaxy. We further extend the metaphor by depicting clusters through volumetric nebulae. Summarizing, this paper contributes a novel approach to depict software coding activities in a 3D virtual environment with star depictions and superimposed star clusters using a 3D layout based on semantic similarity.

2 SOFTWARE GALAXY METAPHOR

We extract the similarity directly from the source code. Through the choice of identifier names and additional comments, developers encode semantics and domain knowledge directly in the source code, which can therefore be seen as a medium of communication between developers. This motivates the use of text clustering techniques from the Natural Language Processing (NLP) domain to source code. One of the most widely used class of NLP-techniques for clustering source code are Topic Models [4]. After applying various preprocessing steps to the vocabulary of the source code files, i.e., (1) stopword removal, (2) removal of keywords of the programming language, (3) identifier splitting, and (4) lemmatization, each document is stored as a Bag-of-Words vector thus neglecting the ordering of the words. Latent Dirichlet Allocation (LDA) applied on the documents leads to a description of each document as a distribution over topics, which are in turn distributions over the vocabulary [2]. Measuring the distance between topics via the Jensen-Shannon distance, we project the topic-word-vectors onto the three-dimensional space using Multidimensional Scaling [5]. The locations of the documents are determined by a linear combination of the topic vectors according to their document-topic-distribution [1].

As visual mapping, we construct a 3D virtual scene where each document is located at its position in the layout. In order to create stylistic stars, we apply point spread functions (PSF), in particular approximations of the Airy function. We use the two basic attributes size and color as independent visual variables of the stars. The size is used as input for the PSF and determines its radius. A mapping that has shown its usefulness in our experiments was a mapping of the Lines-of-Code to size and the number of authors modifying a single file over time to color thus displaying the coding activities [6].

With appropriate application of LDA and a dimension reduction, the proximity of points reflects their semantic similarity. We emphasize this similarity by adding a cluster visualization that visually groups documents that implement a common semantic. These clusters are depicted as semi-transparent volumes using a nebula metaphor. The volume of a nebula is derived by alpha shapes applied to the cluster points and rendered by ray-tracing a low resolution 3D texture (one per nebula). The nebula color is an additional visual variable that can depict the star density within a cluster.

Our prototype is an extension to a web-based, interactive 3D scatterplot framework implemented webgl-operate, WebGL, and TypeScript [12]. The point rendering is adapted to use PSFs and a volume rendering was added as well. The available point labeling and interaction techniques are provided by the framework.

3 CONCLUSIONS

We proposed a 3D software visualization approach based on a layout representing the semantic structure and a galaxy metaphor for displaying the dynamic of the development process. We plan to evaluate more topic modeling and dimension reduction techniques on various datasets of different size to find a best practice for generating semantic layouts [11]. Another step is to expand the idea of the star metaphor and improve the existing rendering quality.

ACKNOWLEDGEMENTS

This work is part of the "Software-DNA" project, which is funded by the European Regional Development Fund (ERDF or EFRE in German) and the State of Brandenburg (ILB). This work is part of the KMU project "KnowhowAnalyzer" (Förderkennzeichen 01IS20088B), which is funded by the German Ministry for Education and Research (Bundesministerium für Bildung und Forschung).

REFERENCES

[1] Daniel Atzberger, Tim Cech, Merlin de la Haye, Maximilian Söchting, Willy Scheibel, Daniel Limberger, and Jürgen Döllner. 2021. Software Forest: A Visualization of Semantic Similarities in Source Code using a Tree Metaphor. In *Proc. 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Volume 3 (IVAPP '21)*. SciTePress, 112–122. https://doi.org/10.5220/0010267601120122

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022. https://doi.org/10.5555/944919.944937

[3] Andrew H. Caudwell. 2010. Gource: Visualizing Software Version Control History. In *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion (OOPSLA '10)*. ACM, 73–74. https://doi.org/10.1145/1869542.1869554

[4] Tse-Hsun Chen, Stephen W. Thomas, and Ahmed E. Hassan. 2016. A survey on the use of topic models when mining software repositories. *Empirical Software Engineering* 21, 5 (2016), 1843–1919. https://doi.org/10.1007/s10664-015-9402-8

[5] Michael A. A. Cox and Trevor F. Cox. 2008. Multidimensional scaling. In *Handbook of Data Visualization*. Springer, 315–347.

[6] Tobias Knöschke. 2020. *Design and Implementation of 3D Visualizations for Topic Maps and their Application in Software Analytics.* Master's thesis. Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam.

[7] Adrian Kuhn, Peter Loretan, and Oscar Nierstrasz. 2008. Consistent Layout for Thematic Software Maps. In *Proc. 15th Working Conference on Reverse Engineering (WCRE '08)*. IEEE, 209–218. https://doi.org/10.1109/WCRE.2008.45

[8] Michele Lanza. 2001. The Evolution Matrix: Recovering Software Evolution Using Software Visualization Techniques. In *Proc. 4th International Workshop on Principles of Software Evolution (IWPSE '01)*. ACM, 37–42. https://doi.org/10.1145/602461.602467

[9] André Skupin. 2004. The world of geography: Visualizing a knowledge domain with cartographic means. *Proceedings of the National Academy of Sciences* 101, suppl 1 (2004), 5274–5278. https://doi.org/10.1073/pnas.0307654100

[10] Frank Steinbrückner and Claus Lewerentz. 2013. Understanding software evolution with software cities. *Information Visualization* 12, 2 (2013), 200–216. https://doi.org/10.1177/1473871612438785

[11] Eduardo F. Vernier, Joao L. D. Comba, and Alexandru C. Telea. 2021. Guided Stable Dynamic Projections. *Computer Graphics Forum* 40, 3 (2021), 87–98. https://doi.org/10.1111/cgf.14291

[12] Lukas Wagner, Willy Scheibel, Daniel Limberger, Matthias Trapp, and Jürgen Döllner. 2020. A Framework for Interactive Exploration of Clusters in Massive Data using 3D Scatter Plots and WebGL. In *Proc. 25th International Conference on 3D Web Technology (Web3D '20)*. ACM, 31:1–2. https://doi.org/10.1145/3424616.3424730