Linecept: An Early Prototype of a Timeline-Based Design Coordination Tool

David Tamas Kutas, Aditya Nair, Prerna Singh, Emily Kan
 {dkutas,adityan1,prerns1,kane1}@uci.edu
 Department of Informatics
 University of California, Irvine
 Irvine, CA

Janet Burge
 jburge@coloradocollege.edu
 Department of Mathematics and
 Computer Science
 Colorado College
 Colorado Springs, CO

André van der Hoek
 andre@ics.uci.edu
 Department of Informatics
 University of California, Irvine
 Irvine, CA

ABSTRACT

In software design, the various stakeholders generate large numbers of heterogeneous artifacts. These artifacts are often developed in, and managed by, different tools. In this paper, we present our initial prototype of Linecept, a tool that helps stakeholders organize, find, and view disparate design artifacts by organizing them on a timeline that presents a single unified view of the artifacts and who created them. We have used Linecept to retrospectively capture design artifacts for its own creation and in a software design class.

CCS CONCEPTS

• **Software and its engineering** → **Software design engineering**; • **Human-centered computing** → **Interaction design process and methods**; *Visualization systems and tools*; • **Applied computing** → *Computer-managed instruction*.

KEYWORDS

Software design, coordination, collaboration, document management

ACM Reference Format:

David Tamas Kutas, Aditya Nair, Prerna Singh, Emily Kan, Janet Burge, and André van der Hoek. 2020. Linecept: An Early Prototype of a Timeline-Based Design Coordination Tool. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20)*, May 23–29, 2020, Seoul, Republic of Korea. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3387940.3392228>

1 INTRODUCTION

The products and by-products of software design come in many forms: sketches, diagrams, graphics, presentations, formal documentation, and more. These development *artifacts* are generated and accessed throughout the duration of the project [3, 4].

Research in Software Engineering (SE) and SE environments has produced a great many tools, for example hyperlinking and hypermedia, to address this problem [2, 8, 9]. The reality, however,

is that stakeholders gravitate towards using tools like Google Drive, Dropbox, Confluence, Jira, and even e-mail to share and manage artifacts. This, however, carries with it significant problems, since the only organization provided is through the file structure and naming conventions. It can be difficult to obtain an overall view of project progress, even with well-chosen file, and directory names.

In this paper, we describe a preliminary prototype for Linecept, an experimental tool that uses a timeline view to organize disparate software design artifacts by grouping them into *phases*. This allows designers to capture their design process at a high level along with the artifacts, when they are produced, and by whom. Our particular intent is to hereby promote continuous capture of the design history of a project, and provide access to this history in a convenient manner.

2 MOTIVATING SCENARIO

A globally distributed team is working on a new web-based inventory management system for a manufacturing company. The stakeholders have expertise in different domains, so they have decided to use Google Drive to manage their product artifacts because all team members have at least some familiarity with it. Sarah, the business architect, kicks the project off by creating a folder in Google Drive called "Inventory Website Project". She wants the project participants to be able to add folders in the drive so she shares the folder with all project members and gives them write access.

The first task is to determine the requirements. Sarah creates a sub-folder called "Business Development." She decides to create a presentation using Google Slides and names it "Business Architecture Requirements." The default access for documents in a shared folder is the same as for the folder. She decides to limit write access to the customer representative, Vihaan, because they are the only two on the team who are working on this document. This means she has to manually change the access given to the rest of the team. Sarah and Vihaan communicate through a variety of mediums to complete this document (e-mail, comments, messaging, etc.) through which Vihaan describes the client's goals and Sarah turns these into requirements.

After the requirements have been completed, Sarah informs the rest of the team. John, the leader of the back-end team, works on the database schema and UML diagrams. The schema is managed in GitHub and the UML diagrams are created in a drawing tool hosted on his own computer and stored locally on his machine. John has some questions about the business goals. Google Drive shows that Sarah was last person to edit the "Business Architecture

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSEW'20, May 23–29, 2020, Seoul, Republic of Korea

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-7963-2/20/05...\$15.00

<https://doi.org/10.1145/3387940.3392228>

Requirements" document so John e-mails her some questions. Sarah receives the questions the next day (she is working in a different time-zone than John) and she forwards the questions on to Vihaan, since he was the person making these decisions. Eventually, the answers get back to John.

Sakura, the leader of the front-end team, has also been told about the new requirements. She works on visualization ideas with her team. She creates a folder in Google Drive, called "Front-End Design" and as they create artifacts, she uploads them to that folder.

After a couple of weeks, Vihaan checks in to see how the project is progressing. He knows that John has been working because he received some questions, but he cannot find any documents on Google Drive so he does not know what is happening. He e-mails John, who then creates a folder called "Back-end Design" and uploads the UML diagrams. John e-mails Vihaan to tell him that the UML diagrams are up on Google Drive and asks him for his GitHub user-id so he can give him access to the schema. Vihaan is not a GitHub user so he does not have a GitHub user-id. John and Vihaan discuss getting Vihaan set up on GitHub so he can view the documents using the GitHub web interface, but decide that, since the schema will be part of the larger code base, it would be easier for Vihaan to find it if the files were also uploaded to Google Drive so Vihaan does not have to hunt through the code to look for them.

Vihaan also browses through Sakura's folder containing the front-end artifacts. He sees that she has uploaded some designs for the landing page but he does not know what she and her team are currently working on. Again, he uses e-mail to contact Sakura for a status update.

There are ways that this scenario is idealized. All teammates choose descriptive names for files and folders. It only involves four stakeholders, all of whom know how to reach each other via e-mail. Even so, the scenario highlights two potential document management problems—a lack of visibility into author responsibilities for team-written documents and project artifacts requiring different tools for access. It also demonstrates that document management systems such as Google Drive only show work in process for documents being developed in the system—they do not show what is being worked on off-line or planned for the future.

Another drawback of using Google Drive is that it captures the history of individual documents (when they were uploaded, when they were modified) but does not give an overview of the overall project timeline (what tasks were started when, what roles the participants played). The link between product and process is only captured if the participants are careful to design their naming scheme. Tasks that are planned and not started, or are started but are being done on tools outside the management platform, are not visible. If there had been more than four participants and a larger number of documents, these problems would have been compounded.

3 APPROACH

Our goal is to provide an easy way to organize and access project artifacts that offers an overview of how the project is progressing by:

- Providing access to all artifacts through a single display, independent of the artifact file format;

- Providing a flexible way to organize/group these items;
- Displaying meta-information about project artifacts to show when each was created and completed as well as who was involved, and how;
- Allowing information about tasks in progress to be captured where artifacts can be attached when complete; and
- Capturing the design history.

To meet these objectives, we used a *timeline view* based on a Gantt chart, a common representation used in project planning. The timeline groups the artifacts into phases with start and end times. An artifact also has a start and end time displayed on the timeline. Stakeholders can be associated with an artifact and given *roles* that indicate their involvement.

4 LINECEPT

In this section we describe how the motivating scenario would play out using Linecept, a preliminary prototype of a tool for design coordination.

4.1 Dashboard

In this alternative scenario, Sarah logs into Linecept and uses the *Dashboard*, shown in Figure 1, to create a new project, "Inventory Website Project" to add the project stakeholders, who will all by default have read access to the uploaded artifacts. Projects with their stakeholders are shown as *Project Cards* on the dashboard.

Figure 1: The Linecept dashboard and the Inventory Website Project's project card example highlighted and zoomed.

Figure 2: The Inventory Website Project's timeline with the Business Development, Back-end Design, and Front-end Design phases. Note that The "Filter and Search Function Design" and "Individual Page Design" Artifact Placeholders span into the future, visualized by crossing through the vertical time axis.

4.2 Timeline

When Sarah clicks on the Project Card on the dashboard, she is taken to the Inventory Website Project's *timeline* view (see Figure 2). This timeline, which can be scrolled vertically and horizontally as well as zoomed in and out, is used to group the project artifacts.

Sarah and Vihaan are working on the requirements in Google Slides. They are ready to share the results with the rest of the team. They want to capture the fact that this was the initial phase of the development process, so they create a *Phase* in Linecept, called "Business Development". Phases are containers that group artifacts and display them on the timeline. Sarah gives the phase a duration that corresponds to their time working on the document. Next, she adds the "Business Architecture Requirements" presentation to Linecept. Sarah enters the start and end times, selects the phase the artifact belongs to, and uploads the document that she has downloaded from Google Drive. She adds herself and Vihaan as *collaborators* for the document and specifies her role as "reviewer" and his role as "goal architect" (indicating that he is responsible for goal definition).

John visits Linecept to review the requirements. John reviews the "Business Architecture Requirements" artifact and has some questions about the project goals. When he looks at the artifact dialog window in Linecept, shown in Figure 3, he sees that the contributors were Sarah and Vihaan and that Vihaan was responsible for the project goals. John contacts Vihaan with his questions.

John also creates a new phase in Linecept, called "Back-end Design." His two main tasks are to create the database schema and the UML diagrams. He has not created these yet, but he uses Linecept to create *artifact placeholders* with a start and anticipated end time to indicate what he is working on and when he expects the work to be completed.

Sakura logs into Linecept and also reviews the requirements posted in the Business Development phase. She creates a "Front-end Design" phase to hold the documents that she is working on. She creates three artifact placeholders for her first three tasks—"Landing Page Ideation," "Filter and Search Function Design," and "Individual Product Page Design." She starts with "Landing Page Ideation" and uploads a photograph of a white-board mock-up created by her team.

Figure 3: Artifact review dialog window.

Vihaan logs into the system and checks to see how the project is progressing. He notices that John has some artifacts that, according to the timeline view, should be complete, but have not yet been uploaded. Vihaan e-mails John to ask him to upload these artifacts, which he does. Vihaan also can see that Sakura has some tasks in progress. The deadline for these has not yet passed so he decides to check on them later.

5 PRELIMINARY CASE STUDIES

Linecept has been used in two preliminary case studies. First, we used Linecept to retrospectively capture all the artifacts created during its development. We captured four months worth of artifacts. This included nine development phases with 73 artifacts and three empty artifact placeholders; 70 percent of the uploaded files described visual information, such as UML diagrams, software flows, and layouts. The other 30 percent were code snippets that the design team felt were important enough to capture and upload. The team is currently working on a new version of Linecept, and is using Linecept to refer back to these previous work products.

We also used Linecept in a Software Design undergraduate course at the University of California, Irvine. The class had roughly 260 students who used Linecept for three design studio projects, one of which required Linecept use and the other two made it optional. Students used Linecept to create 153 projects and uploaded 2,054 files. The goal of the course was to teach students about "good" high-level approaches to design (e.g., identifying stakeholders early, creating alternatives, iteration). The TAs and instructors used Linecept to review the design history for each project team and examine the intermediate artifacts in detail to assess the process used by the student design teams.

6 RELATED WORK

As mentioned previously, various commercial tools exist to share and manage project artifacts, such as Google Drive, Dropbox, and GitHub. Additionally, a rich body of research on project timeline visualizations, collaboration tools, knowledge management, and design history capture and use has emerged over time. Here we will focus on design history.

Each design artifact captures a snapshot of the design at the moment in time when it was created. Capturing the addition or evolution of artifacts captures the design history. For source code, this information can be extracted from version control systems. The Code Time Machine [1] works with a versioning system to allow users to explore the commit history for software files. For documents, some editing platforms, such as Google Docs, keep track of the revision history. This information is not always that easy to analyze. Chronicle captures the workflow history for a single document and allows the history to be re-played [6]. GEM-NI [10] can be used to explore and generate design alternatives using the design history.

Linecept captures the design history on a timeline. Other tools that follow this approach are the Classroom Bridge [5], which captures versions of documents created during team projects, and Designer's Outpost [7], which captures design notes (the electronic version of sticky notes). Linecept allows an additional level of organization within the timeline by grouping the documents in phases, where a phase could correspond to the design phase or capture other meaningful groupings, such as subsystems.

7 CONCLUSION

Our goal for this initial version of Linecept was to demonstrate how project artifacts could be organized and accessed in a tool that could also be used to show project participants how the project is progressing. We have used our tool to retrospectively capture

design data from the tool itself and also in a software design course with over 260 students.

Studying how Linecept was used through observation, artifact analysis, and user surveys, has revealed limitations of the tool that we will be addressing in future work. In particular, we want to make it easier to create phases and upload documents and to integrate Linecept with external tools, such as Google Drive. We also will be adding a notification feature to alert users when documents have been changed and an artifact's timeline "ends" with no document uploaded. More broadly, we want to build on our initial prototype to turn Linecept into a full-fledged design coordination tool: by providing active assistance for developers working on their design activities in the now, Linecept implicitly builds a history that can be consulted and revisited as needed.

ACKNOWLEDGEMENTS

The authors would like to thank the undergraduates who either contributed to the design and development of Linecept or were participants in the class in which it was deployed.

REFERENCES

- [1] E. Aghajani, A. Mocci, G. Bavota, and M. Lanza. The code time machine. In *Proceedings of the 25th International Conference on Program Comprehension, ICPC '17*, page 356–359. IEEE Press, 2017.
- [2] K. M. Anderson, R. N. Taylor, and E. J. Whitehead. Chimera: Hypermedia for heterogeneous software development environments. *ACM Trans. Inf. Syst.*, 18(3):211–245, July 2000.
- [3] M. Book and A. van der Hoek. Sketching with a purpose: Moving from supporting modeling to supporting software engineering activities. In *2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 93–96, May 2018.
- [4] M. Cherubini, G. Venolia, R. DeLine, and A. J. Ko. Let's go to the whiteboard: How and why software developers use drawings. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, page 557–566, New York, NY, USA, 2007. Association for Computing Machinery.
- [5] C. H. Ganoe, J. P. Somervell, D. C. Neale, P. L. Isenhour, J. M. Carroll, M. B. Rosson, and D. S. McCrickard. Classroom bridge: Using collaborative public and desktop timelines to support activity awareness. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology, UIST '03*, page 21–30, New York, NY, USA, 2003. Association for Computing Machinery.
- [6] T. Grossman, J. Matejka, and G. Fitzmaurice. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology, UIST '10*, page 143–152, New York, NY, USA, 2010. Association for Computing Machinery.
- [7] S. R. Klemmer, M. Thomsen, E. Phelps-Goodman, R. Lee, and J. A. Landay. Where do web sites come from? capturing and interacting with design history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '02*, page 1–8, New York, NY, USA, 2002. Association for Computing Machinery.
- [8] N. Streitz, J. Haake, J. Hannemann, A. Lemke, W. Schuler, H. Schütt, and M. Thüring. Sepia: A cooperative hypermedia authoring environment. In *Proceedings of the ACM Conference on Hypertext, ECHT '92*, page 11–22, New York, NY, USA, 1993. Association for Computing Machinery.
- [9] W. Wang and J. M. Haake. Flexible coordination with cooperative hypertext. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space—Structure in Hypermedia Systems: Links, Objects, Time and Space—Structure in Hypermedia Systems, HYPERTEXT '98*, page 245–255, New York, NY, USA, 1998. Association for Computing Machinery.
- [10] L. Zaman, W. Stuerzlinger, C. Neugebauer, R. Woodbury, M. Elkhaldi, N. Shireen, and M. Terry. Gem-ni: A system for creating and managing alternatives in generative design. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, page 1201–1210, New York, NY, USA, 2015. Association for Computing Machinery.