SYNKIT:THE DEFINITIVE TOOLKIT FOR CREATING COLLABORATIVE SOFTWARE

Joel Shajan
Dept.of Computer Science and Engineering
St.Joseph's College of Engineering and Technology
Palai,Kottayam,Kerala

Varghese Martin
Dept.of Computer Science and Engineering
St.Joseph's College of Engineering and Technology
Palai,Kottayam,Kerala

Varun K V
Dept.of Computer Science and Engineering
St.Joseph's College of Engineering and Technology
Palai,Kottayam,Kerala

Vishnu Nair P
Dept.of Computer Science and Engineering
St.Joseph's College of Engineering and Technology
Palai,Kottayam,Kerala

Prof. Mereen Thomas
Dept.of Computer Science and Engineering
St.Joseph's College of Engineering and Technology
Palai,Kottayam,Kerala

***Abstract*—We present Synkit a collaborative software which is easy to implement yet robust toolkit to rapidly convert any pre-existing software into multi functional environments. Synkit is lightweight and provide all the necessary functionalities needed for action synchronization and management for multiuser environment. It also provide support for group communication and chat management which means you can communicate with each other and put down your thoughts and suggestions within the software itself,hence saving one's valuable time. It is available as third party service/development kit that can be easily integrated into existing and new software cheaply and effectively.**

I. Introduction

As remote work culture increases in popularity, it is essential to have software that enables effective collaboration. Synkit aims to create a simple way of developing collaborative software to meet the demands of the increasingly online workforce.It eliminates the need for developers to provide complex networking and data consistency controls.Instead, Synkit is flexible and adaptable, able to accommodate changing team needs and individual preferences.The overarching goal of Synkit is to help people working on common goals achieve them through real-time collaborative editing, as well as integrated audio and chat features.Any changes or updates made by one user are synchronized in real time so that all other users can immediately view and incorporate them.By providing these capabilities, Synkit enables people to work together productively and creatively even when they are not in the same physical place.Teams can tap into diverse perspectives, rapid feedback loops, and round-the-clock productivity regardless of geographic barriers.Collaborative software like Synkit will only become more crucial as remote work continues to grow in popularity and necessity.With Synkit, we aim to create a collaborative software solution that is simple to set up yet powerful enough to meet the demands of modern teams and workflows.

II. Objective and Scope

- To research, design and develop a software that helps implement collaborative environments in software with minimal development time.
- To improve over existing tools that offer similar features and benefits to add value to the individual or company using the service.
- To provide Audio and Chat facilities for better communication and management of any work or a project.
- To help Multiusers to attain common goals by working on a single project.

III. Literature Survey

A. Fluid Computing

Fluid computing is an emerging paradigm in computer sci ence that emphasizes the fluidity and adaptability of software systems, data, and computing resources. Unlike traditional computing models that aim to develop static software systems and applications, fluid computing promotes the continuous change and reconfiguration of software to address evolving requirements and environments. This literature survey summa rizes key concepts and research directions in fluid computing based on an analysis of relevant publications. Several key concepts underlie fluid computing, including software dynamism, perpetual beta, and meta-design. Software dynamism refers to the ability of software systems to dynam ically reconfigure themselves in response to changes without service interruption. The perpetual beta concept suggests that software should continuously evolve and experiment in re sponse to feedback and real-world experience. Meta-design emphasizes

designing software systems with self-modification capabilities that can dynamically change their own structure and behavior. Some of the promising research directions in fluid comput ing include dynamic software updating, adaptive software sys tems, and autonomic computing. Dynamic software updating studies how to update software components at runtime. Adap tive software systems can dynamically adapt their behavior and structure to address new requirements and environments. Autonomic computing aims to develop self-managing com puter systems that can optimize their operations with minimal human input. In conclusion, fluid computing is an emerging paradigm with promising concepts and research directions focused on developing software systems that can dynamically adapt to change. Fluid software systems promise to be more robust, dependable, and long-lasting than traditional static software systems. This literature survey summarizes the key concepts and research directions in fluid computing based on an analysis of publications from reputable peer-reviewed sources.

B. Importance of Collaboration in Today's World

Collaborative software is crucial in today's fast-paced and globally connected world. When software is designed to be collaborative, it allows people from all over the world to pool their diverse knowledge, skills, and resources, which would not be possible otherwise. Collaborative software fosters creativity, innovation, and higher quality work by incorporating feedback and input from multiple individuals with varied backgrounds and perspectives. Overall, collaborative software leads to better solutions and outcomes than software designed for individual use, especially for complex problems that require interdisciplinary thinking. In today's world, these complex problems are increasingly common and collaboration is key to solving them. Collabo rative software provides the capabilities required for people to connect and work together effectively on these challenging issues. As such, collaborative software plays an essential role in today's world by enabling the sharing of ideas, rapid feedback, and the synthesis of diverse perspectives that drives innovation and progress. Without these collaboration mechanisms, we would not be able to solve many of the world's most pressing problems or take advantage of opportunities that emerge from globalization.

C. How fluid computing helps improve collaborative experience

Fluid computing enables collaborative software systems to dynamically adapt to the changing needs and environments of users and teams. As teams grow and evolve, fluid software can reconfigure itself to accommodate new members and ways of working. It can also optimize team processes by monitoring how teams interact and identifying opportunities for improvement. By making software adaptive and responsive to the dynamics of collaboration, fluid computing creates a better experience for users and helps teams work together more effectively.

D. Existing Solutions

[1] LiteDoc: Make Collaborative Editing Fast, Scalable, and Robust: Learnt LiteDoc's locking mechanism, which limits who can edit a component at any given time. This supports less features, but can handle editing conflicts.With LiteDoc,an alternative strategy to address this issue by offering a more straightforward semantic model,collaborative editing quick, scalable, and reliable.Formally demonstrate that LiteDoc achieves deterministic guarantees of correctness, which is more significant. One user may only write in one section at a time when using LiteDoc, which divides the shared document into various sections. By doing this, all conflicts that result from having several writers send messages to the same address are eliminated. Additionally, this mechanism eliminates the need to set up time-consuming modules for OT, diff-sync, and rollbacks in case of conflicts. Although LiteDoc offers fewer features than general collaborative editors like Google docs, it is still customary (and polite) to refrain from writing to the same location at the same time when several people are collaborating.

[2] CollabEd: A Platform for Collaboratizing Existing Editors: A platform called CollabEd makes it simple to collaborate on linear editing systems. The modular architecture, which is geared towards enabling developers to create CollabEd plugins for already-existing editors without having to provide or comprehend the networking and data consistency controls necessary for real-time, synchronous collaborative editing. This article presents the open-source CollabEd application, which has been ported to work with several well-known programming editors (including Eclipse, NetBeans, jEdit, etc.) and a drawing programme (DrawSWF). Optimistic replication datamodel (MSET), which preserves local-site editing response time without the use of operational transformation, serves as the foundation for the CollabEd platform. The capability to record collaboration sessions for later playback with user-specific editing statistics is one of the features we'll highlight.

[3] A social collaborative virtual environment for software development: Learnt about SCI – Social Collaborative IDE a development environment that combines social networking with collaborative development, integrating development tools and information sharing between team members.This study introduces the SCI (Social Collaborative IDE), a social development environment that combines the ideas of social networks and collaborative development environments. SCI combines presence and activity awareness data with tools for group programme development. The information on activity awareness enables project and session sharing as well as behaviour awareness among developers. The collaboration tools offer a variety of options for synchronous and asynchronous teamwork as well as information sharing. SCI offers a framework for managing team member invitations that are still pending.

[4] Hyper-linked communications: WebRTC enabled asynchronous collaboration: The concept of hyper-linked communications applies many of the hypermedia principles that are frequently used in web content to the field of communications.Voice and video calls with integrated communication content can be coordinated, organised, and navigated. No other medium compares to voice and image for the ability to convey emotion. Conference calls can benefit even more from hypermedia concepts. The design, implementation, and evaluation of a hyper-media communication platform that utilises WebRTC technology and is targeted at the web platform are discussed. With support for multiple media types, a collaborative text editor, time annotations, instant messaging, and a method to superimpose hyper-media to video, our prototype offers a hyper-linked, non-linear, multi-party video conferencing platform.

[5] Remote collaboration across heterogeneous large interactive spaces: Collaborative virtual environments, with a focus on how to accurately reflect each user's real environment in a virtual environment. A current initiative that attempts to support remote collaborative involvement across diverse large interactive venues is also introduced. In order to facilitate collaborative interaction with extremely huge datasets and computations in the context of scientific data analyses, design, engineering, decisionsupport, education, and training,launched a significant initiative named DIGISCOPE. There are ten interconnected huge interactive spaces that make up DIGISCOPE. These spaces feature motion trackers, 3D displays, large wall-sized monitors, and virtual reality systems. In the framework of DIGISCOPE, they wish to develop user interfaces that take into account collaboration as a key component of the system. Users should be able to engage with shared contents while also being able to comprehend what other users are seeing or doing and how they can interact. For distant collaboration across diverse large interactive areas, it is essential to understand what other users are doing right now.

[6] Does Distance Still Matter? Revisiting Collaborative Distributed Software Design: Studied the design activities of both co-located and distributed professional software designers. Learnt that distance is a major factor and must consider extra non-technical aspects.The majority of software development activities require supporting distributed collaboration, which is a requirement of global software engineering. However, physical separation makes collaboration difficult. Technologies for collaboration and communication have made significant strides in recent years. Therefore, investigate whether these developments allow for efficient remote collaboration. In order to achieve this, the design processes used by both locally based and remotely based professional software designers. Based on observations,the findings from the review of questionnaires and video recordings of design sessions.

[7] Team Lab: a collaborative environment for teamwork: Learnt Team lab, a client-server application for software development teams which supports concurrent programming activity and sharing of a central code repository integrated with a collaborative virtual environment that provides tools for communication and other functions.The tools supporting software development and those supporting communication and other social activities are currently two separate sets of tools. In addition to making work easier, integrating the two functionalities would enable teams to record both formal and informal project-related information and use it to build, preserve, and provide access to a corporate memory. This might result in new, more effective work procedures.A MUM tool called Team Lab is a collection of software development tools for Smalltalk programmers. Its integration into MUM transforms it into an intriguing new type of environment for software developers, despite being similar in many ways to environments like ENVY or StORE. As a MUM tool, Team Lab operates based on events. Users can subscribe to any event that Team Lab objects comprehend, and subscribers are notified as soon as the event takes place.

IV. Conclusion

In Conclusion, Synkit offers a straightforward and user-friendly set of tools to build out a flexible collaborative platform that can scale to support numerous users, greatly reducing development time and experience. Capable of managing multiple rooms with multiple users each at once. To make work easier and advance a shared objective, provide audio and chat management. Able to quickly integrate Synkit functionalities into current systems. It keeps track of all user actions as a consistency and fail safe measure. Synchronizes all users' actions in a specific room. All incoming action sequences are gathered into a single, efficient pipeline. Also updated are all user states according to the unified pipeline. well-optimized and adaptable to a variety of platforms and applications.

References

[1] S. Kumar, H. Pan, R. Wang and L. Tseng, "LiteDoc: Make Collaborative Editing Fast, Scalable, and Robust," 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, TX, USA, 2020, pp. 1-6, doi: 10.1109/PerComWorkshops48775.2020.9156221.

[2] K. G. Granville and T. J. Hickey, "CollabEd: A Platform for Collaboratizing Existing Editors," 2009 International Conference on Mobile, Hybrid, and On-line Learning, Cancun, Mexico, 2009, pp. 90-96, doi: 10.1109/eLmL.2009.20.

[3] H. Bani-Salameh, C. Jeffery and J. Al-Gharaibeh, "A Social Collaborative virtual environment for software development," 2010 International Symposium on Collaborative Technologies and Systems, Chicago, IL, USA, 2010, pp. 46-55, doi: 10.1109/CTS.2010.5478525.

[4] H. Rocha and R. L. Pereira, "Hyper-linked communications: WebRTC enabled asynchronous collaboration," 2017 IEEE International Conference on Communications (ICC), Paris, France, 2017, pp. 1-7, doi: 10.1109/ICC.2017.7996702.

[5] C. Fleury, N. Férey, J. -M. Vézien and P. Bourdot, "Remote collaboration across heterogeneous large interactive spaces," 2015 IEEE Second VR International Workshop on Collaborative Virtual Environments (3DCVE), Arles, France, 2015, pp. 9-10, doi: 10.1109/3DCVE.2015.7153591.

[6] Guang Yang and I. Tomek, ”Team Lab: a collaborative environment for teamwork,” Proceedings Sixth International Workshop on Groupware. CRIWG 2000, Madeira, Portugal, 2000, pp. 142-145, doi: 10.1109/CRIWG.2000.885168.
[7] (no date) WebRTC. Available at: https://webrtc.org/ (Accessed: October 12, 2022).
[8] Rust (no date) Rust Programming Language. Available at: https://www.rust-lang.org/ (Accessed: October 12, 2022).
[9] The Collaborative Interface Design Tool. (no date) Figma. Available at: https://www.figma.com/ (Accessed: October 12, 2022).