外文翻译译文

实时协作 (RTC) 在这两个领域都有悠久的历史 研究和工程领域。它正在获得显着 在 COVID-19 大流行期间流行。对于地理上分散的团队，团队成员通常必须拥有用于会议的远程协作工具 和演示目的。此外，考虑到分布式和快速增长的数据量和快速 现代网络浏览器的发展，有必要提供一个有组织的基于网络的群体感知平台来支持 大量用户之间的协作。在 RTC 平台中，不仅多媒体的工件是共享的，而且是共享的。 对工件的控制是广播和同步的。

一些开创性的工作，如 GroupSketch [1]， VideoWhiteboard [2] 和 Liveboard [3] 通过捕获用户的绘图实现了协作多媒体系统 并将它们投影到远程屏幕上。基于相同的 技术，各种屏幕共享产品，例如Zoom， 开发了 Cisco WebEx 和 Google Hangouts/Meet 并且现在被广泛使用。近期，随着高速发展 现代网络技术，大量基于网络的协作 开发了包括 Collabode [4]、RichReview++ [5] 和 Tele Board [6] 在内的工具。

据我们所知，尽管已经为多种用途开发了各种基于 Web 的群件工具，例如 作为白板绘图和文档编辑，没有系统可以 集成这些功能以适应通用多媒体协作。金等人。 [7]做出了宝贵的贡献 在这个方向上，他们提出了 MVC 架构 用于无处不在的协作。他们的工具仍然只处理 静态媒体，如白板图纸和图像，没有 有效地处理具有动态的其他类型的媒体 视频和网页等内容。 此外，对于目前流行的屏幕共享产品， 在特定会话中，媒体的内容和 对媒体的操纵是通过捕获 连续显示，导致网络消耗大 带宽。这对地理上的大 团队分散，网络质量不稳定。相比之下，我们建议将演示文稿的内容拆分为 静态媒体资源和动态动作。静态媒体 资源，例如视频或 PDF 文档，可以是 事先传送给与会者；和动态事件 发生在会话中，例如使视频静音，并广播和 即时同步。因此，协作会话是 组织为静态材料和动态的组合 封装在事件驱动的消息流中的事件。 通过这些消息，我们还实现了协作事件的精确记录和回放，使我们的工作与众不同 来自传统的协作平台。

为了弥补上述先前工作中的研究空白， 我们提出了一个上下文感知的基于网络的协作多媒体系统——CWcollab。具体来说，我们的贡献是：

1) 通用协作多媒体系统。

• 支持一般多媒体。不仅是静态媒体（PDF 文档、图像等），还包括动态媒体 媒体（视频、网页等），在 CWcollab。这一显着特点使我们的 以前工作的工作，如表 I 所示。

• 支持一般事件。会话中的所有操作 被捕获为事件，在 飞。这也表明系统对其他人开放 插件模式中的可能扩展。开发者 可以根据我们的制服在不同类型的媒体上添加对各种协作事件的支持 界面，在第 III-A 节中演示。

• 支持一般环境。我们的系统是 完全基于网络。用户可以通过以下方式访问系统 各种平台，包括台式机、平板电脑和 移动设备，只要支持 Web 浏览器。这显着消除了复杂性 在不同平台上的设置。

2) 一种对象优先的上下文感知方法，用于捕获和重放媒体动作以实现丰富的功能 网络带宽低。

• 对象优先的媒体控制。据我们所知，大多数基于 Web 的协作工具都是 基于位置或基于比例，这意味着 媒体控制通过绝对或相对位置同步。不过很多网站 已经应用了响应式网页设计，其中 用户界面在具有各种屏幕尺寸的各种设备上自动调整。这构成 捕捉和回放事件的挑战 传统的基于位置的方法。相比之下， 在 CWcollab 中，媒体控件与媒体相关 对象。我们提出了一个对象优先的混合 同步方法，代表每个动作 在简单的消息中，在第 III-B 节中讨论。

• 丰富的功能。由于每个媒体控件都由一个简单的消息表示，CWcollab 还支持 丰富的交互功能，用于协作和 演示，例如材料准备，实时 同步和精确重播会话。

• 网络带宽低。对象优先方法的使用也意味着 CWcollab 具有非常低的带宽使用率，与 当前的视频会议产品，例如 Zoom 和谷歌环聊。在我们的研究中，数百 千字节足以将事件存储在 会话，与使用的数百兆字节相比 在屏幕共享工具中。

本文的其余部分在第二部分介绍了相关工作，在第三部分讨论了架构设计和实现，并在第四部分评估了我们的平台，包括与 Google Hangouts 的比较。 第五节总结了本文。

RTC 通常被认为是计算机支持的协作工作的一个子域。以前，人们通过视频/音频通话实现基本的远程协作。虽然很明显，这种形式的交流只能允许非常初步的合作。之后，研究人员开发了更复杂的协作应用程序。一些开创性的工作包括 GroupSketch [1]、VideoWhite board [2] 和 Liveboard [3]。在这些初步系统中，用户和他们的图纸被摄像头捕捉、传输并投射到远程屏幕上。这些系统的一个问题是仅传输屏幕截图，因此它们缺乏在地理上分散的人们之间提供交互的灵活性。随后，研究人员开发了更精细的原生桌面协作应用程序。例如，布斯等人。 [8] 提出了一种“mighty mouse”多屏协同工具，通过 VNC 协议提供平滑的鼠标跨平台移动。这种系统的一个关键问题是它们通常需要在不同平台上进行复杂的设置，因为它们需要在这些平台上具有不同的实现。

外文翻译原文

原文出处：C. Tang， B. Wang， C. Y. R. Chen and H. Wu， “CWcollab： A Context-Aware Web-Based Collaborative Multimedia System”， ICC 2021 - IEEE International Conference on Communications， 2021:1-6.

Real-Time Collaboration (RTC) has a long history in both research and engineering domains. It is gaining signifificant popularity during the COVID-19 pandemic period. For geographically dispersed teams, it is usually essential for teammates to have a remote collaboration tool for conferencing and presentation purposes. Additionally, considering the distributed and rapidly increasing volume of data and expeditious development of modern web browsers, it is necessary to provide an organized web-based group-aware platform to support collaboration among a large number of users. In RTC platforms, not only are the artifacts of multimedia shared but also the controls on the artifacts are broadcast and synchronized.

Some of the pioneering work, such as GroupSketch [1], VideoWhiteboard [2], and Liveboard [3], implemented collaborative multimedia systems by capturing users’ drawings and projecting them to remote screens. Based on the same technology, various screen sharing products, such as Zoom, Cisco WebEx, and Google Hangouts/Meet, were developed and are now widely used. Recently, with the rapid development of modern web technologies, lots of web-based collaboration tools, including Collabode [4], RichReview++ [5], and TeleBoard [6], were developed.

To our knowledge, even though various web-based groupware tools have been developed for versatile purposes such as whiteboard drawing and document editing, no systems can integrate these functionalities to suit general-purpose multimedia collaboration. Kim et al. [7] made a valuable contribution in this direction that they proposed an MVC architecture for ubiquitous collaboration. Their tool still only handled static media like whiteboard drawings and images, without effectively working on other types of media with dynamic contents such as videos and web pages.

Furthermore, for current popular screen sharing products, in a specifific session, both the contents of the media and manipulations on the media are transmitted by capturing the display continuously, leading to large consumption of network bandwidth. This poses challenges for large geographically dispersed teams with the unstable network quality. By contrast, we propose to split the contents of a presentation into static media resources and dynamic actions. The static media resources, for example, a video or a PDF document, can be transmitted to attendees beforehand; and the dynamic events occurring in a session such as muting a video are broadcast and synchronized on the flfly. As a result, a collaboration session is organized as the combination of static materials and dynamic events encapsulated in an event-driven stream of messages. With these messages, we also implemented the precise recording and replay of collaboration events, differentiating our work from traditional collaboration platforms.

To cover the research gaps in prior work mentioned above, we propose a context-aware web-based collaborative multimedia system-CWcollab. Specififically, our contributions are:

1) A general-purpose collaborative multimedia system.

• Support for general multimedia. Not only static media (PDF documents, images, etc.) but also dynamic media (videos, web pages, etc.), is supported in CWcollab. This notable feature differentiates our work from prior work, as illustrated in Table I.

• Support for general events. All actions in a session are captured as events, sent and handled on the flfly. This also indicates the system is open to other possible extensions in a plugin pattern. A developer can add supports for various events of collaboration on different kinds of media following our uniform interface, demonstrated in Section III-A.

• Support for general environment. Our system is totally web-based. Users can access the system from various platforms, including desktops, tablets, and mobile devices, as long as web browsers are supported. This signifificantly eliminates the complexity of setup on different platforms.

2) An object-prioritized context-aware approach to capture and replay media actions for rich functionalities with low network bandwidth.

• Object-prioritized media controls. To our knowledge, most web-based collaboration tools are position-based or proportion-based, which implies that media controls are synchronized through absolute or relative positions. However, many websites have applied the responsive web design, where the user interface is automatically adjusted on various devices with various screen sizes. This poses challenges for capturing and replaying events with the traditional position-based approach. By contrast, in CWcollab, media controls are related to media objects. We propose an object-prioritized hybrid synchronization approach, representing each action in simple messages, discussed in Section III-B.

• Rich functionalities. As each media control is represented by a simple message, CWcollab also supports rich interactive functionalities for collaboration and presentation such as material preparation, real-time synchronization, and precise replay of a session.

• Low network bandwidth. The usage of an objectprioritized approach also implies that CWcollab has a very low bandwidth usage, compared with current video conferencing products such as Zoom and Google Hangouts. In our study, hundreds of kilobytes can be enough to store the events in a session, compared with hundreds of megabytes used in screen sharing tools.

The remainder of this paper introduces the related work in Section II, discusses the architectural design and implementation in Section III, and evaluates our platform including a comparison with Google Hangouts in Section IV. Section V concludes the paper.

RTC is usually considered as a subdomain of Computer Supported Cooperative Work. Previously, people achieved basic remote collaboration by means of video/audio calls. While it is obvious that this form of communication can only allow very preliminary cooperation. Afterward, researchers developed more complicated collaborative applications. Some of the pioneering work includes GroupSketch [1], VideoWhiteboard [2], and Liveboard [3]. In these preliminary systems, users and their drawings are captured by cameras, transmitted and projected to remote screens. An issue of these systems is that only screenshots are transmitted, thus they lack the flflexibility to provide interaction among people geographically dispersed. Subsequently, researchers developed more elaborate native desktop collaborative applications. For instance, Booth et al. [8] proposed a “mighty mouse” multi-screen collaboration tool, which provides a smooth mouse movement cross-platform, via VNC protocol. A crucial issue of this kind of systems is that they usually require complicated setups on different platforms because they need to have distinct implementations on these platforms.