HotPi: Open-Source Collaborative Patient Documentation

Tim Jungnickel
TU Berlin, Germany
Chair for Complex and Distributed IT Systems
tim.jungnickel@tu-berlin.de

Juan Cabello
TU Berlin, Germany
Chair for Complex and Distributed IT Systems
juan.e.cabellobilbao@campus.tu-berlin.de

Klemens Raile
Charité Universitätsmedizin Berlin, Germany
Department of Pediatrics
klemens.raile@charite.de

Abstract

In hospitals and clinics a lot of paper based documentation can be observed. Since paper based solutions are prone to information loss and data inconsistencies, the development of mobile applications for real-time collaboration is promising. In this work, we identify the major challenges for fitting application designs to medical environments. We present an open-source prototype that can be reused and customized based on the clinic's needs.

Author Keywords

real-time collaboration; operational transformation; patient documentation

ACM Classification Keywords

H.5.3 [Group and Organization Interfaces]: Computer-supported cooperative work

Introduction

Hospitals and clinics typically use complex enterprise software, like hospital information systems (HIS), to run their business operations. This software covers a broad variety of tasks, from treatment documentation to billing and resource planning. We observe many different software systems that are closely interconnected. These systems often lack a seamless integration, which results in frustration on the side of the medical and administrative staff [4].

CSCW 2017, February 25–March 1, 2017, Portland, OR, USA.
ACM ISBN 978-1-4503-4688-7/17/02.
http://dx.doi.org/10.1145/3022198.3026342

At the very end of this software chain are doctors and other clinicians which would rather focus on the treatment of patients than the processing of the business operations. At the Charité, the largest university hospital in Europe, we have seen many paper based workarounds. The medical staff use printed copies of shared Word documents to organize treatment documentation as well as the daily schedule. When a patient is discharged from the clinic, the annotations on the printed copies are collected for a report that is processed by the HIS. These paper based solutions are still quite common in many hospitals, even though such workarounds create major issues, e.g. information loss or inconsistent data. Moreover, modern real-time distributed collaboration, as can be seen in Google Docs, is not supported. Since these collaborative systems work well for editing text documents and spreadsheets, we explore their feasibility in clinics and hospital environments.

We note that solutions like Electronic Health Records are already part of the ongoing innovation in hospital infrastructures. We propose a solution that can run alongside with the existing infrastructure and offers a adaptable application design for mobile devices, aiming to provide an alternative to paper based documentation. We identify major requirements for real-time collaboration software in clinic environments and present an open-source prototype to show the feasibility of our approach.

Requirements

The development of suitable software for the described purpose goes along with strict requirements of usability and compatibility with law. We identify the most important requirements to be the following:

> **Always Available:** Even in unstable network environments, the doctors must be able to read, modify and add information.
>
> **Privacy Protection:** Sensitive information must not be disclosed to unauthorized parties.
>
> **Cross-platform:** The system must support various platforms, especially mobile devices, such as tablets.

The first requirement raises a fundamental challenge in distributed systems research. According to the CAP Theorem [2], we cannot guarantee a consistent view of the data alongside with partitioned networks and the availability requirement. Hence, we cannot avoid that doctors see different information on their mobile devices if network failures are present. This seems to be very critical in a hospital environment. However, with the *classical* paper based solutions, where every doctor annotates a personal copy of the patient's file, inconsistencies are widely accepted. To automatically solve possible inconsistencies and to provide a high responsiveness if a working network connection is present, a consistency control system is needed.

The second requirement leads to strong encryption of the stored data and the communication. Depending on the regulations of the country, no external service provider like Google Docs can be used, since sensitive data must remain inside the hospital's infrastructure.

The third requirement ensures that existing devices, such as PCs, smartphone and tablets can be used without the need for a specific hardware.

Application Design

From the stated requirements, we derive an application design, based on the latest available open-source technologies. The third requirement implies the development of a web-based application. In contrast to a native mobile application, a web-based application runs on various operating systems without further adaptation. Together with the

MERN

The MERN stack is a software bundle that is based on the programming language JavaScript and comprises four building blocks of which the name derives: (1) MongoDB for the database, (2) Express.js for the web server application, (3) React.js with Redux for the client application and (4) Node.js for the server platform. Website: http://mern.io

Operational Transformation

OT is a mechanism that enables real-time collaboration for shared documents. The basic idea is, that two or more collaborators can edit their local digital copy of a document at any time and all changes are automatically exchanged and merged. Popular programming libraries that can be used to develop collaborative applications are ShareJS and the Google Realtime API.

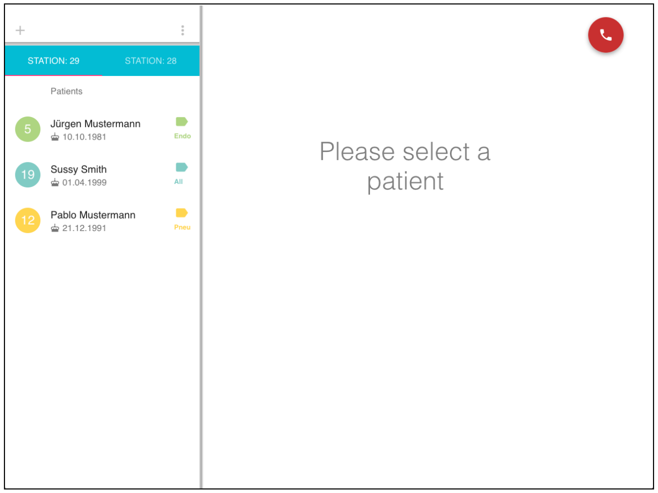
first requirement, a single-page application is required. The single-page application runs, once loaded, completely autonomously in the browser. We suggest the use of a modern open-source software stack like MERN. This suggestion is in accordance with the second requirement, since no external software service is needed. Note that we describe the suggested technologies in the sidebar.

If one doctor updates a patient's record, the update is immediately present on the doctor's device and the information will be propagated to the other devices as soon as possible. If the network connection is unavailable, the updates will be queued. To regain a consistent state after patient information have been updated, we suggest the use of a widely accepted consistency control system like Operational Transformation (OT) [5, 1]. These systems have been developed within the CSCW community. OT is the underlying technology of the leading online collaboration tools like Google Docs.

Prototype

Based on the stated application design, we implemented a prototype that aims to be an alternative to outdated paper based solutions, which are still presently used in many hospitals. The open-source implementation is in an early stage and publicly available on github: https://github.com/hotpi/frontend

As shown in Fig. 1 and 2, our prototype features a comfortable way to browse between patients and to annotate the patients' information with different notes. We structured the layout based on the paper document structure we have found at the hospital. Hence doctors can add, delete and modify notes for the anamnesis, the treatment documentation (or history) and the upcoming tasks. Particular informa-



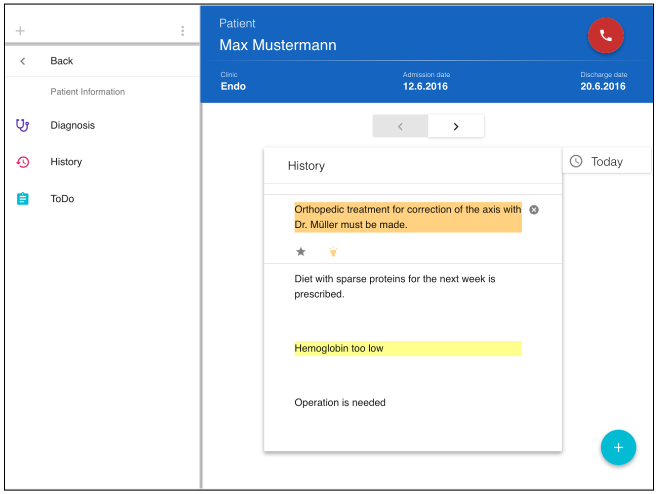
5 Jürgen Mustermann 10.10.1981 Endo

19 Sussy Smith 01.04.1999 All

12 Pablo Mustermann 21.12.1991 Pneu

Please select a patient

Figure 1: Patient selection view of the prototype.



Patient: Max Mustermann

Clinic: Endo

Admission date: 12.6.2016

Discharge date: 20.6.2016

History

Orthopedic treatment for correction of the axis with Dr. Müller must be made.

Diet with sparse proteins for the next week is prescribed.

Hemoglobin too low

Operation is needed

Figure 2: Active notes of a patient treatment.

Prototype source code
and further information:

https://github.com/hotpi/frontend

Technical Contributions

Since there is no open-source OT library that offers an offline mode, we decided to implement the OT mechanism without the use of an existing programming library. Our prototype is, to the best of our knowledge, the first open-source OT application that offers an offline mode and is based on a hierarchical JSON data model. We base our implementation on the mechanism we presented in [3].

tion in the notes can be highlighted with different colors as well as different priorities can be chosen.

Since the underlying technology of our prototype is Operational Transformation, the prototype supports both real-time collaboration and offline mode. Hence, if a network connection is available, multiple doctors can edit the notes simultaneously. If no network connection is available, the application remains fully functional and the information is updated when the device is online again.

We note that the prototype follows the information structure we have seen at the Charité. Other hospitals are likely to use a different structure to process the patients' information. Therefore, we have chosen an open-source license that allows an easy adaptation of the prototype.

Evaluation and Future Work

In order to ensure correct behavior of the application, we simulated a collaborative editing of the patients' information. Therefore, we used an authentic patients' information document from the hospital and divided all given information on the sheet into two groups. Two independent persons in separate rooms received the task to add the content of the assigned group to the prototype. As a result, both test persons collaboratively managed to transfer the content of the paper sheet into the prototype without any problems.

Additionally, we introduced the prototype to a small group of doctors at the hospital. The first feedback was very good and all of them stated that the usage of such software would improve their daily work routine. Future work includes test runs at the hospital. Currently, we are work on the improvement of the robustness of the software, as we cannot risk loss of data when real patients' information is processed.

Conclusion

The documentation of the treatment of patients in hospitals and clinics requires software that reflects the workflow of the clinicians. The technical challenges of replacing paper based solutions, which are still present in many hospitals, are complex and heavily related to distributed systems research. We introduced an open-source prototype that addresses these issues. With the presented application design, we prepared the development of customized solutions for other hospitals and clinics. Ultimately, we believe that by introducing fitting application design approaches, we increase the variety and the usefulness of collaborative software in clinic environments.

REFERENCES

1. C. A. Ellis and S. J. Gibbs. 1989. Concurrency Control in Groupware Systems. *SIGMOD Rec.* (1989).
2. Seth Gilbert and Nancy Lynch. 2002. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News* (2002).
3. Tim Jungnickel and Tobias Herb. 2016. Simultaneous Editing of JSON Objects via Operational Transformation. In *ACM Symposium on Applied Computing (SAC '16)*.
4. D.A. Ludwick and John Doucette. 2009. Adopting electronic medical records in primary care: Lessons learned from health information systems implementation experience in seven countries. *International Journal of Medical Informatics* (2009).
5. Chengzheng Sun and Clarence Ellis. 1998. Operational Transformation in Real-time Group Editors: Issues, Algorithms, and Achievements. In *Computer Supported Cooperative Work (CSCW '98)*.