J. Austerjost, M. Bargholz, M. Porr, D. Geier, T. Becker, D. Marquard, P. Lindner, T. Scheper and S. Beutel

A flexible IT infrastructure for the integration of smartglasses into the brewing laboratory as a digital support for standard analysis workflows

Standard operating procedures (SOPs) are an often-used medium for the reproducible step-by-step execution of complex laboratory operations. Even in today's era of digitalization, most SOPs are still paper-based. This might cause disorder within the lab and often distracts the experimenter from the actual experiment. Furthermore, most of the experimental documentation nowadays is still done manually by transferring experimental data and results in paper-based laboratory journals, which is quite sensitive to errors as well as time-consuming. We developed a digital laboratory infrastructure that enables interactive hands-free experiment guidance and instrument control via smart safety goggles, also called smartglasses. Instructions for an SOP and experimental data can be displayed directly into the experimenter's field of view and triggered by voice commands. Experimental data and working steps can be processed and documented instantly. The developed laboratory infrastructure allows for the flexible integration of laboratory instruments and SOPs. Different SOPs commonly used for the spectrophotometric analysis of beer (MEBAK methods) were implemented showing the applicability of the established system in the brewing laboratory. Furthermore, a benchmark study of the system was performed. The developed solution enables a faster experiment execution and analysis than conventional paper-based SOPs and is a first step towards the integration of smart safety goggles for the support of laboratory workflows. This might pave the way to easier process documentation and an increase of laboratory workflow efficiency.

Descriptors: smartglasses, laboratory automation, wearables, assisted reality

1 Introduction

More than 50 years have passed since the first head-mounted displays (HMDs) were introduced. Pioneers like Ivan Sutherland were the first that conceived and technically implemented the idea of a head-worn device that displays graphical content into the user's field of view in the late 1960s [1]. The device created by Sutherland and his colleagues was called "The Sword of Damocles", whose name was inspired by the necessity to mount it to the ceiling due to its massive weight and its bulky head tracking mechanism. Miniaturization of electronics and optics have evolved during the decades since, and so have head-mounted displays [2]. Smartglasses size has ranged from backpack solutions in the 1980s to smartglasses that now nearly have the dimensions of regular glasses [3].

After Google rolled out their smartglasses model "Google Glass" in 2013, smartglasses technology became available to a broader audience. Other manufacturers followed Google's example and released models with similar technical characteristics. Although most of the current smartglasses models were not successful in the consumer field, many applications emerged within professional areas. Applications of smartglasses have been evaluated in healthcare, logistics, and manufacturing as a documentation respectively commissioning tool or as a support during workflows [4–7].

Recently, the idea of giving tasks in the laboratory a digital surplus has caught the interest of researchers in different scientific disciplines. Integration efforts extend beyond smartglasses to virtual assistants and other smart devices such as tablets and smartphones that are subject to current research in which their potential as a supporting tool in laboratory settings is evaluated [8, 9]. Smartglasses provide crucial advantages over handheld devices. These advantages include hands-free operation by voice commands and the direct display of information in the user's field of

https://doi.org/10.23763/BrSc18-20austerjost

Authors

Jonas Austerjost, Institute of Technical Chemistry, Leibniz University Hannover, Hannover, Germany, Institute of Brewing and Beverage Technology, Technical University of Munich, Freising, Germany; Malte Bargholz, Marc Porr, Daniel Marquard, Patrick Lindner, Thomas Scheper, Sascha Beutel, Institute of Technical Chemistry, Leibniz University Hannover, Hannover, Germany; Dominik Geier, Thomas Becker, Institute of Brewing and Beverage Technology, Technical University of Munich, Freising, Germany; corresponding author: beutel@iftc.uni-hannover.de

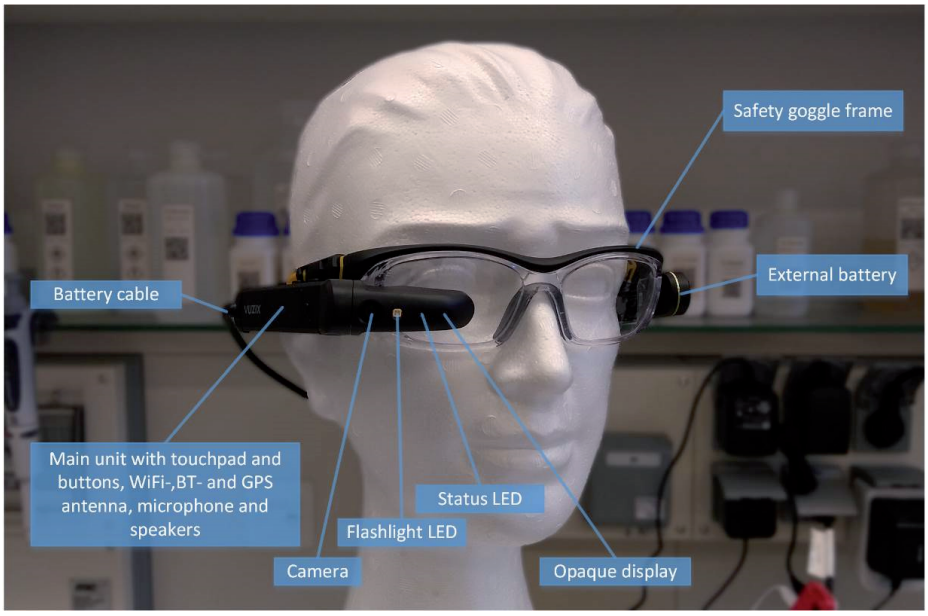

Fig. 1 Smart safety goggles used for the evaluation of the developed digital laboratory architecture (Smartglasses Vuzix M300, Vuzix Corp., USA)

view. Furthermore, safety goggles are mandatory in most laboratories – so why not give them an additional digital benefit? A handful of researchers already seized the idea of bringing smartglasses into the laboratory.

Cortazar et al. used a Google Glass device for the quantification of chlorophyll in plant leaves, while *Feng* et al. developed a Google Glass application for the processing of immunochromatographic assays [10, 11]. Both applications take advantage of a client-server model, where the internal smartglasses camera captures images. Subsequently, the image data are transmitted to an external server for processing. The results of the processing can then be retrieved by the smartglasses and shown in the user's field of view. This approach has the advantage of conserving the battery and processing power of the smartglasses by sourcing out the hardware demanding processing task to a more powerful server unit. A similar concept was worked out by *Austerjost* et al. who used smartglasses and smartphones for the determination of colony-forming units on agar plates [12]. Other applications within the laboratory include the use of smartglasses in physics education as a tool for better experiment perception [13].

The use of smart safety goggles within the laboratory offers the potential to revolutionize experimental work. Not only can built-in sensors (e.g. microphone and camera) be used to collect experimental data, but also external sensors or instruments can be connected using the internal Bluetooth or WiFi capabilities (see Fig. 1). We developed a digital laboratory architecture that allows the easy creation of standard operating procedures (SOPs) that can be retrieved and displayed by smartglasses that act as the front-end. These SOPs can be triggered and navigated hands-free using voice commands. Furthermore, laboratory instruments can be integrated into the back-end architecture, allowing the control of these devices and the display of instrument data into the experimenter's field of view. Measured experimental data and workflow data are saved to a database. Therefore, experimental results can be traced back to their source at a later date. The evaluation of this setup has been carried out using different SPOs covering the spectrophotometric analysis of beer [14]. A benchmark of the developed system was performed by comparing the execution time of a paper-based SOP with the execution time of a smartglasses-based SOP.

2 Materials and methods

2.1 Back-end application

The developed back-end application is a JavaScript application running on NodeJS v8, a cross-platform native JavaScript interpreter (https://nodejs.org/en/). It provides both a JSON/HTTP API (Application Programming Interface) for retrieving and creating protocols and a separate WebSocket API that manages all instrument measurements that are triggered via the backend.

The JSON API stack consists of the express HTTP server (http://expressjs.com/de/), a database backend provided by the popular ORM (Object Relational Mapper) sequelize.js (http://docs.sequelizejs.com/) and custom controller code, which translates HTTP requests to database actions. Database access is also shared with the WebSocket stack, which is powered by socket.io (https://socket.io/), adding reliability and data consistency features in comparison to raw WebSockets, and a custom plugin system providing the interface with laboratory devices. SOPs are implemented using the YAML (http://yaml.org/) format. A custom human-readable syntax was developed that fits the needs of standard laboratory workflows.

A single plugin interfacing with a photometer (ThermoFisher Scientific GENESYS™ 10S UV/VIS) was implemented, which provides functions for setting the wavelength, choosing a turret position, blanking the spectrophotometer and measuring an absorbance value. The photometer was integrated using its integrated USB-RS232 interface. It was directly connected to the backend server via a USB-A/USB-B cable. Port commands were provided by the manufacturer (ThermoFisher Scientific Corp., USA).

The system runs on a Dell OptiPlex 3050 Micro (Dell Technologies Inc., USA) with an Intel Core i3-7100T, 16 GB of DDR4 RAM and a 500 GB SSD. Linux Ubuntu 16.04 is installed as an operating system.

2.2 Front-end applications

Two clients have been developed for communicating with the backend application: A command-line tool for inserting and managing SOPs and an Android application for displaying and executing SOPs.

The command-line tool is a JavaScript application running on NodeJS (https://nodejs.org/en/). It uses the axios (https://github.com/axios/axios) library for HTTP communication and supports inserting, deleting and listing protocols that are available via the backend. Protocols are specified in a simplified YAML syntax that is

mapped to the expected JSON syntax before inserting them via the provided backend HTTP/JSON API. It also automatically uploads associated images to the backend before inserting the protocol.

The Android application was implemented using Android Studio (Alphabet Inc., USA) and the Kotlin programming language (https://kotlinlang.org/).

Internally the Android application uses a reactive-programming approach powered by RxJava (https://github.com/ReactiveX/RxJava) and uses square's moshi (https://github.com/square/moshi) and retrofit (https://github.com/square/retrofit) for JSON decoding and encoding. It also uses square's Picasso (https://github.com/square/picasso) library for downloading and displaying images and socket.io-java-client (https://github.com/socketio/socket.io-client-java) library for interfacing with the WebSocket API provided by the backend. MxParser (http://mathparser.org/) and MathView (https://github.com/Nishant-Pathak/MathView) are used for calculating and rendering mathematical equations. All SOP slides for the individual working steps were prepared using Microsoft Visio 2016 and saved as .png-files with a resolution of 640 × 360 pixels. Different SOPs for the analysis of beer were visualized and corresponding SOP YAML files were created.

The functionality of the Android client application was evaluated using smartglasses Vuzix M300 (Vuzix Corp., USA), which uses specific voice commands to trigger the progress of the SOPs, and a Samsung Galaxy Tab S2 T719N (Samsung Corporate Group, South Korea), which uses graphical touch buttons to move forward and backward through the SOPs.

2.3 Beer analysis assays

Four different MEBAK (Central European Commission for Brewing Analysis) analyses have been transferred into interactive smartglasses-based illustrated SOPs. They have been tested for function and can be carried out using the proposed hard- and software solutions that are freely available. The methods include the spectrophotometric determination of beer color (MEBAK method 2.12.2), the spectrophotometric determination of bittering units in beer (MEBAK method 2.17.1), the spectrophotometric determination of total carbohydrates in beer (MEBAK method 2.6.4.1.1) and the spectrophotometric determination of free nitrogen in beer (MEBAK method 2.6.4.1.1) [14]. Corresponding YAML SOPs are made freely available on GitHub [15]. For detailed procedures and used material see supplemental material.

2.4 System benchmark and user study

To benchmark and give a first insight about the performance of the developed system, 12 graduate students in the field of biotechnology and chemistry (age 23 to 29) were divided into two groups. The first group of six performed MEBAK method 2.12.2 (spectrophotometric determination of beer color) using a conventional paper-based SOP (see supplemental material for details) and the second group of six performed the same experiment with the help of a smartglasses-based SOP. Both groups performed the experiment using the same laboratory working place and instruments (see supplemental material). Experiments were performed in a conventional laboratory surrounding with active exhaust ventilation (50–55 dB; equal to the sound level of a normal conversation). Test persons that executed the paper-based SOP received a calculator and a pen. Neither of the participants had previous knowledge about the performed experiment or the workflow. All participants performed the experiment for the first time to exclude learning effects. The laboratory training status of the test persons was expected to be similar, due to a similar practical education. Prior to the experiment, the group that executed the smartglasses-based SOP received a one-minute introduction on which voice commands to use to operate the application. In addition, the test persons were given time to adjust the smartglasses to their eyes. The experiment execution times were measured using a conventional smartphone timer. The timer was started after turning over the paper-based SOP or using the "select" command within the smartglasses SOP application respectively. The timer has been stopped after the test persons finished the paper-based SOP by saying "finished", or used the command "end protocol" within the smartglasses-based SOP respectively.

A user study was performed with the previously mentioned 12 graduate students. All users performed MEBAK method 2.12.2 (spectrophotometric determination of beer color) using the voice-controlled smartglasses application. Afterwards, they were asked to fill out a survey that included questions about themselves, their opinion about the usability of the utilized application and their impression of the technology (see supplemental material for the questionnaire and detailed results).

3 Results and discussion

3.1 Established server environment

The established server environment that handles all inquiries from clients provides both a JSON/HTTP API (Application Programming Interface) for retrieving and creating SOPs and a separate WebSocket API that manages all instrument measurements that are triggered via the backend (see Fig. 2).

This separation is due to the duality of workflows; they contain static data (e.g. the instructions for each step of the SOP or an equation), that can be fetched beforehand, as well as dynamic data (e.g. measurement results) that can only be made accessible after a certain progression in the SOP was reached.

Instrument integration into the established setup was done via the native USB-RS-232 interface of the spectrophotometer and a wired USB connection to the backend server. A plugin structure was worked out to enable flexible device integration. This means a new plugin needs to be created for every new device that should be integrated. Plugins contain information about the device interface, the communication protocol and specific command strings that are mapped to a keyword within the SOP syntax to trigger the command (see chapter 3.3). This approach allows the use of a wide range of common communication standards including MQTT, Modbus, and TCP/IP (see GitHub repository for the used RS-232 JavaScript plugin structure) [15–18]. Wireless communication standards like Bluetooth of WiFi can also be implemented (see

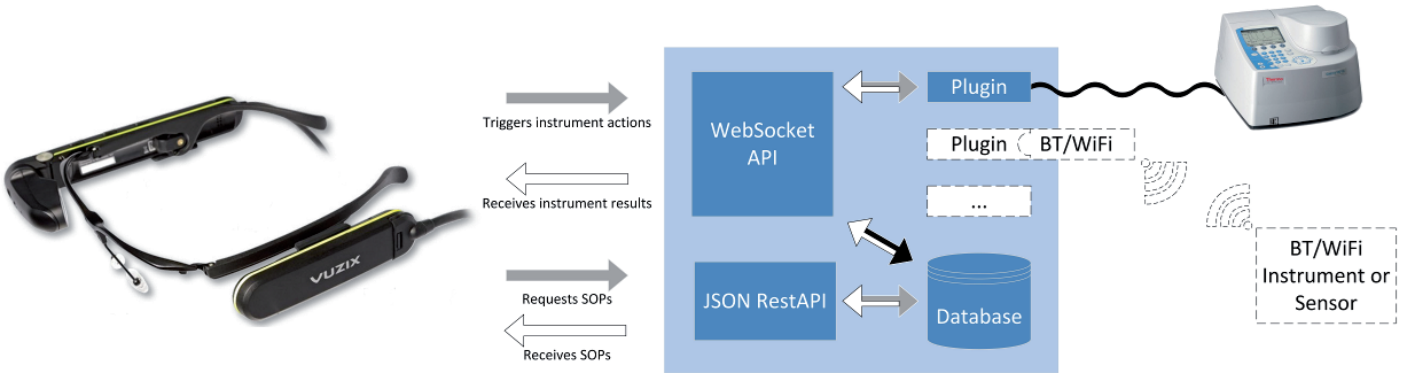

Fig. 2 **Architecture and data flow of the established setup. Gray arrows indicate data requests and white arrows indicate the return of the requested data. The black arrow specifies the storage of instrument and process data in the database. White boxes with dashed lines state flexible plugins that can be implemented into the architecture**

Fig. 2) if appropriate hardware standards are met on both the back end and instrument site of the setup.

With every start of the developed backend application, the WebSocket stack loads all plugin JavaScript files in a predetermined directory and registers their name and provided function with a custom plugin manager. When a client wants to trigger a measurement it sends a request containing a valid action ID in a custom WebSocket channel. The backend then checks if this request is valid and synchronously returns an associated unique result ID to the client, therefore allowing the client to issue multiple requests in parallel. It then asynchronously performs the measurement by calling the corresponding function on the loaded plugin and creates a database entry (IOResult) for the result containing the same unique ID that was returned to the client. The client receives the result value and ID asynchronously via a WebSocket channel and then updates its interface. We chose an asynchronous approach instead of a polling approach to improve application and backend performance and scalability. Although the workflow data is linked to an automatically generated ID that is traceable within the relational database, the registration of a custom ID might be desirable. Depending on the technology that is used to trace samples within the laboratory or company, different solutions are implementable. Either the previous scanning of sample QR or barcodes or the recording of voice and subsequent speech-to-text processing are options to store individual sample IDs [19, 20]. Additionally, the IDs in the database can be modified using standard SQL (Structured Query Language) operations. The provided SQL API allows for the integration of the database into third party software like LIMS and ERP (entertprise resource planning) systems that enable the

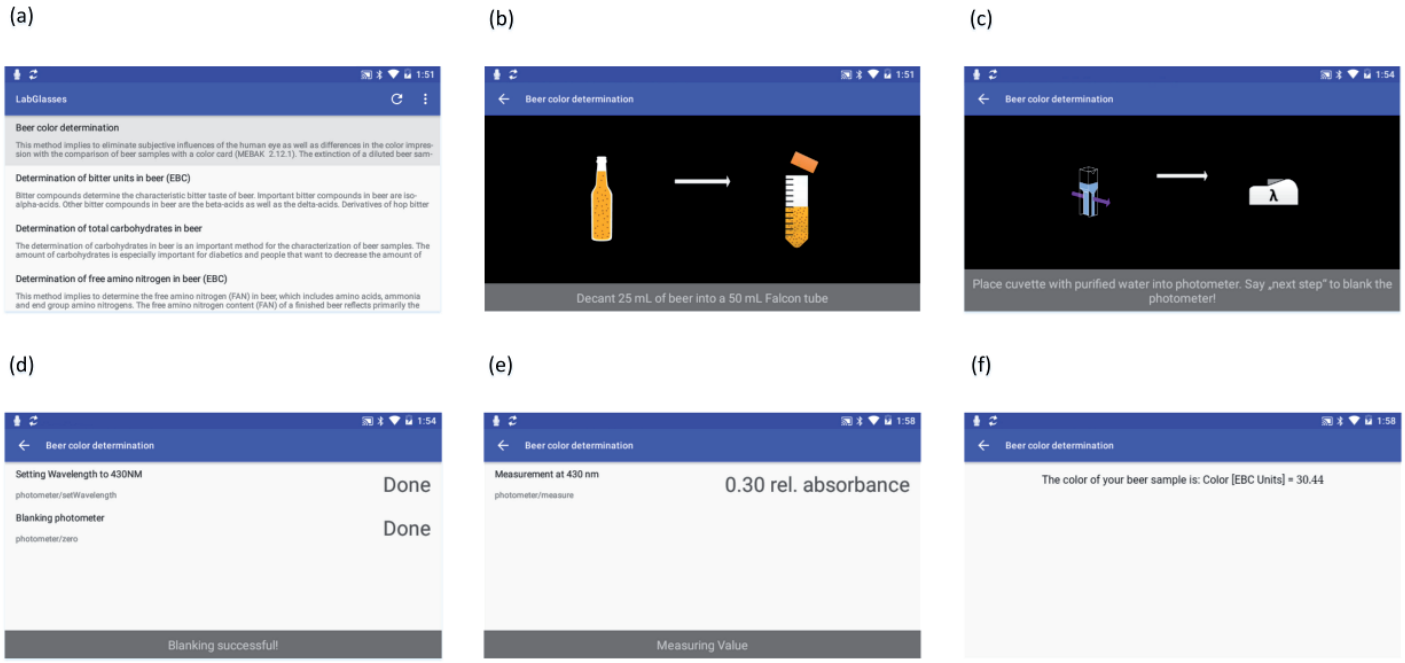

Fig. 3 **Exemplary screens showing the progress of the SOP for beer color determination displayed into the experimenter's field of view. (a) Master screen that is shown after opening the app and that lists the currently available SOPs and a short description (all implemented SOPs are shown); (b) Ordinary instruction that is not mapped to an action (instruction text is shown in the gray box under the instruction pictograms); (c) Ordinary instruction that precedes an instruction that triggers an action; (d) An instruction that triggers an action and shows the successful execution (setting of a wavelength and blanking); (e) Display of a successful measurement and the corresponding relative absorbance units; (f) Display of the experimental results using an equation that is defined within the SOP**

Table 1 Implemented voice commands that allow the navigation within the developed client Android application

Voice command	Triggered action
"Up"	Navigates up in the master detail screen
"Down"	Navigates down in the master detail screen
"Select"	Selects and starts an SOP within the master detail screen
"Next Step"	Triggers the next step of the SOP
"Previous Step"	Goes back to the previous step of the SOP
"Next Result"	Navigates forward in the result menu in multipath SOPs
"Previous Result"	Navigates backward in the result menu in multi-path SOPs
"End protocol"	Finishes the protocol after the last experimental instruction and goes back to the master detail screen
"Go home"	Ends the application

creation of experimental reports and the monitoring of experimental progresses [21].

3.2. Front-end application for smartglasses

The developed application is running on Vuzix M300 smartglasses (Vuzix Corp., USA) as well as all other Android devices above API Level 23 (restricted due to Vuzix development tools). It features a Master-Detail application flow (see Fig. 3, screen a) allowing the user to first select a protocol and then guides the user through the instructions, even allowing voice navigation on the Vuzix M300 smartglasses (see Table 1 for implemented voice commands). Instructions are displayed according to their type, i.e. timer instructions display a specified countdown, equation instructions display a mathematical equation that can contain previous measurement results and specific calculated results, and normal instructions are rendered with an associated image plus a text overlay containing the instruction text (see Fig. 3, screen 2). Additionally, if measurements (actions) are associated with the instruction, a list of pending measurements is displayed, automatically executed and finally displayed to the user (see Fig. 3, screens 4 and 5). The application prevents the progression of the SOP if a step is not finished, such as a countdown has not run out yet or a measurement failed. If a multi-path instruction is detected the user is prompted to select a result that matches his experimental progress. The correct following instructions are then displayed to the user and irrelevant instructions are hidden.

SOP pictograms were custom-made for all experimental steps and integrated as .png-files. A standard graphic format, which allows the use of nearly every image editing program.

The evaluation of the application was done with test users that are used to performing laboratory experiments and have a similar professional background. The main focus was to reach a high comprehensibility of the SOPs and an easy handling of the client app to achieve an independent performance of an experiment. It turned out that the used pictograms are comprehensible and the voice commands could be processed with high accuracy. All test persons could perform the implemented SOPs completely self-reliant (see 3.4).

3.3 SOP creation and syntax

To ensure the creation of new SOPs by untrained users without any programming skills, a human-readable, easy to understand markup language format was chosen. The YAML format (YAML Ain't Markup Language) allows the straightforward construction of SOPs, where a list of arrays forms a specific SOP step.

Each SOP has a unique ID, by which it can be identified and retrieved by the Android client-application, as well as a name and

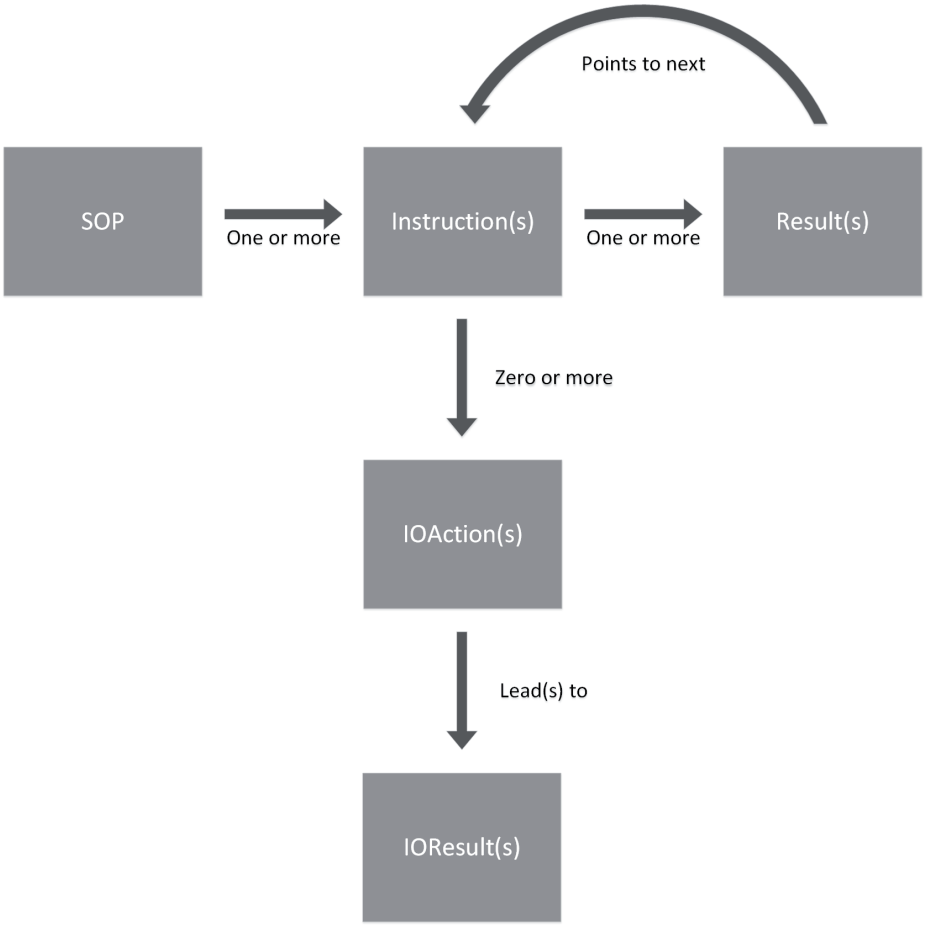

Fig. 4 Flowchart of an SOP. An SOP consists of one or more instructions. An instruction leads to one or more results (depending if it is a multipath experiment or not). An instruction can contain zero or more IOActions. IOActions are instrument actions that are triggered at a defined instruction. These IOActions lead to corresponding IOResults (e.g. a successful measurement or the adjustment of instrument parameters)

```
name: Beer color determination
description: This method implies to eliminate subjective influences of the human eye as well as differences in the color
impression with the comparison of beer samples with a color card (MEBAK  2.12.1). The extinction of a diluted beer sample is
measured in a 10-mm-cuvette at a wavelength of 430 nm. Color giving melanoides and melanoidines that are formed in the
brewing process absorb at 430 nm. The color in EBC units can be calculated by using an appropriate factor.
imageBasePath: beercolor/ # a path relative to the global image base path
instructions: # we will refer to instruction as steps and vice versa for the length of the example
    - Step 1 : # this is the instruction identifier which is later used for target this instruction with a result
        description: Beer color determination.\n Say “next step” to advance to the next work instruction and „previous
       step“ to go to the previously shown work instruction!
        results:
            - add a description for a result here: Step 2
    - Step 2 :
        description: Put on Gloves!
        results:
            - add a description for a result here: Step 3
    - Step 3 :
        description: Decant 25 mL of beer into a 50 mL Falcon tube
        results:
            - add a description for a result here: Step 4
        ...

    - Step 10 :
        description: Place cuvette with purified water into photometer. Say „next step“ to blank the photometer!
        results:
            - add a description for a result here: Step 11
    - Step 11 :
        description: Blanking successful!
        actions: # actions describe anything that happens with an external device
            - plugin: photometer
              action: setWavelength
              wavelength: 430
              humanReadableName: Setting Wavelength to 430NM
              unit: null
            - plugin: photometer # the name of the corresponding plugin in labor-api (required)
              action: zero # each plugin supports certain actions (required)
              humanReadableName: Blanking photometer
              unit: null
        results:
            - add a description for a result here: Step 12
    - Step 12 :
        description: Remove previously inserted cuvette and place cuvette with beer sample into photometer
        results:
            - add a description for a result here: Step 13
    - Step 13 :
        description: Say „next step“ to measure!
        results:
            - add a description for a result here: Step 14
    - Step 14 :
        description: Measuring Value
        actions:
            - plugin: photometer
              action: measure
              humanReadableName: Measurement at 430 nm
              equationIdentifier: a
              unit: rel. absorbance
        results:
            - add a description for a result here: Step 15
    - Step 15 :
        description: The color of your beer sample is: \n\nColor [EBC Units]:
        equation: a*25*4
        results:
            - add a description for a result here: Step 16
    - Step 16 :
        description: Dispose all samples. \n The experiment has finished!\n Say \"end protocol\" to exit the SOP!
        results:
            - experiment is finished: null # if you specify null as the next instruction, the labor-api-cli will know that
              this is your last instruction.
```

Fig. 5 **Structure of an SOP using the suggested YAML structure. The SOP describes the spectrophotometric determination of beer color (MEBAK 2.12.2, see chapter 2.3.1). The three dots after step 3 represent steps 4–9, which are ordinary instructions any instrument actions involved. Gray highlighted text shows the name of the SOP and a description of the process that is shown in the master screen (see Fig. 3 a), and the source directory of the image data. Italic and bold text marks descriptions that are shown in the smartglasses' display as instruction text under the individual step image. They can also contain equations (see step 15) that process previously recorded data (so-called IOResults). Underlined text represents instrument actions (so-called IOActions), which portray device events (e.g. measurements or adjustments). They can be linked to an equation identifier, which enables the processing of instrument data that is recorded during the action in a later step (see step 15)**

a description. This unique ID is generated when the protocol is initially uploaded with the developed command-line tool. In addition, each SOP contains an array of instruction structures, where each instruction represents a step in the source SOP (see Fig. 4). An instruction contains a unique ID, a description, an optional image, one or more results, zero or more IOActions and optionally an equation or a timer sequence (for incubation or waiting sequences) (see Fig. 5). Each result contains a description, an optional image and a target instruction identified by its unique ID. Instructions and results from a circular progression structure allowing it to effortlessly represent multi-path SOPs, where experiment progression depends on intermediary experimental results (e.g. specific color or turbidity changes that require different experimental paths) (see Fig. 4).

IOActions contain a unique ID, a plugin name, a plugin function name, optional plugin arguments, a human-readable name, an optional unit, and an optional equation identifier. They represent external laboratory instrument actions associated with this instruction. Plugin name and plugin action names map to plugins provided by the custom plugin system through the WebSocket API (see Fig. 2). The equation identifier can be used to reuse the IOResult, which was retrieved in the associated instruction in later instructions that might have an equation associated with them (e.g. a calculation at the end of the experiment). Specifying an equation on an instruction changes the type of the instruction to display a calculated mathematical expression instead of a description and image. Similarly, specifying timer duration on an instruction changes the type of the instruction so that the client will display a countdown. Pictograms are associated with the instruction step by naming the specific pictograms after the step (e.g. the pictogram for step 2 would be implemented as 2.png). If no image data is available, only the text of the step description is shown in the Android client app.

The four above mentioned MEBAK analyses (see chapter 2.3.) have been implemented into the setup. Detailed illustrated SOPs were worked out and have been tested (see online repository for details) [15].

Although the first step to a user-friendly creation of SOPs for the developed laboratory environment is done, future efforts include the establishment of an “SOP wizard”. This tool shall make the creation of SOPs even easier, by providing the user a graphical user interface (GUI) to enter the descriptions, actions, and images.

3.4 System benchmark and user study

A benchmark of the system was performed by letting two groups of test users perform the SOP for the spectrophotometric determination of beer color (see supplemental material for details). One group used a paper-based SOP (see supplemental material) and the other group performed the experiment using the smartglasses-based SOP [15]. A total of 12 test users (6 female and 6 male users) were used for the first benchmark of the system (see 2.4 for more details). Users that are not familiar with the analysis method were chosen as test persons, a method previously reported to benchmark assisted industrial assembly workflows [22]. A benchmarking study including method experienced laboratory technicians might have led to a different outcome, though, and is intended for future studies.

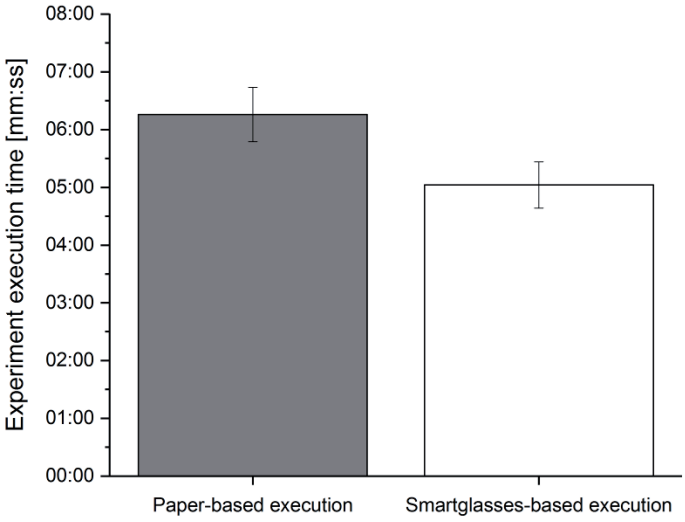

Fig. 6 Average experiment execution times of the paper-based SOP execution of the spectrophotometric determination of beer color (MEBAK method 2.12.2) (gray column) and the smartglasses-based SOP execution of the same experiment (white column)

The average execution time of the paper-based SOP was determined as 6:16 min ± 28 s (median: 6:14 min), whereas the smartglasses-based execution time of the experiment could be stated as 5:03 min ± 24 s (median: 5:01 min; see Fig. 6).

This results in an average process efficiency boost of 19.4 % by using the presented infrastructure. Due to the automated spectrophotometer actions and the automated calculation of the experimental result, time could be saved. It has to be stated that the performed experiment only involves three instrument actions and only one measurement and calculation. More complex smartglasses-based SOPs could even increase the workflow efficiency compared to a paper-based SOP execution. Because all test persons performed the experiment for the first time, the developed system might as well be an interesting tool to reduce initial training phases for experimental workflows.

After performing the paper-based SOP, the first group of test users was given the chance to perform the previously made experiment using the smartglasses. Afterwards, all users participated in a survey to rate the overall usability of the system and to state their opinion about smartglasses technology in the laboratory (see supplemental material for details). Two of the study participants had previously worn smartglasses, for demonstration purposes only, though. The overall usability of the system received an average rating of 8.75 ± 0.75 on a scale from 1 (bad) to 10 (very good; median: 9), which states a user-friendly and intuitive application design. Furthermore, 4 out of 12 users stated that they have privacy concerns about using smartglasses as a support for their laboratory work. Yet, all participants declared that they think smartglasses have the potential to increase the efficiency of laboratory work. 25 % of the participants think that it takes a decade until the presented technology is widely available in laboratory environments, whereas 50 % think the technology will be already present in 3 to 5 years. All but one participant of the survey reported that they sometimes use their personal smartphone as a support during laboratory work (e.g. to look things up, or to calculate dilutions, etc.). This indicates

that smart devices are already part of the daily laboratory routine. New solutions like smartglasses combine smartphone features with a hands-free operating mode. Once this technology is more widespread it has not only the chance to change the work efficiency in laboratories but also other professional environments [23].

5 Conclusion

We established a flexible digital laboratory infrastructure that enables the bidirectional communication between smart devices and laboratory instruments. Additionally, we were able to show the applicability of smartglasses as a voice-controlled supporting tool for analysis workflows in the brewing laboratory. We proposed an easy to understand and human-readable SOP syntax for an uncomplicated integration of SOPs into the developed infrastructure. Our benchmarks revealed that the established solution could save time compared to a conventional paper-based experiment execution and documentation. Furthermore, we could show that the developed smartglasses application is user-friendly and features an easy operation.

The described architecture has the potential to significantly ease the work burden of performing and documenting standard experimental processes since all device parameters and measurement results are automatically saved within a designated database. The integration of the system into commercially available electronic laboratory notebooks or laboratory information and management sytsems can be realized using the integrated SQL database API, which enables the automated generation of a process report. Also the communication with common ERP systems can be performed to monitor the progression of workflows. This could lead the way to data documentation being in accordance with good manufacturing and good laboratory practice standards. Additionally, the proposed solution might be of interest for education and training purposes since an easy to handle application for the step-by-step experiment performance is provided.

Smartglasses just recently emerged in science and other professional fields. To unleash their full potential within the laboratory and other environments, a holistic digitization approach must be pursued. The IT infrastructure, as well as the instrument infrastructure, needs to be set up or updated. To simplify integration work, instrument manufacturers need to move together to agree upon consistent instrument communication standards. This results in less documentation work, fewer documentation errors and inevitable efficiency boosts. Although the performed process and hardware evaluation can overall be described as positive, future work focuses on further reliability tests, the scale-up of the instrument setup and the determination of the technology acceptance of test persons from a more diverse field of professional backgrounds [24]. Another important step that needs to be performed to use the proposed system within the industry is the validation of the architecture and all of its components [25, 26]. Furthermore, controversial issues regarding the technology, including the potential violation of personal rights due to the integrated camera system and the possible negative health effects resulting from long-term usage of wireless technologies, need to be discussed at length [27–29]. We decided to upload the project code to GitHub, to enable other researchers to profit from our findings and to evaluate the architecture in their laboratory surroundings and to customize the software to fit their needs[15].

Funding

This work was financially supported by the Bavarian Ministry of Economic Affairs and Media, Energy and Technology within the Information and Communications Technology program (Grant number: IUK-1504-0006//IUK470/001).

Acknowledgments

The authors thank Ethan Overfelt for proofreading the manuscript and all test users for their participation.

Conflicts of Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

6 References

1. Sutherland, I.E.: A head-mounted three dimensional display, Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I), ACM Press, New York, New York, USA, 1968, p. 757.
2. Furht, B.: Handbook of Augmented Reality, Springer New York, New York, NY, 2011.
3. Mann, S.: Wearable computing: A first step toward personal imaging, Computer, **30** (1997), no. 2, pp. 25-32.
4. Muensterer, O.J.; Lacher, M.; Zoeller, C.; Bronstein, M. and Kübler, J.: Google Glass in pediatric surgery: An exploratory study, International Journal of Surgery, **12** (2014), no. 4, pp. 281-289.
5. Liebert, C.A.; Zayed, M.A.; Aalami, O.; Tran, J. and Lau, J.N.: Novel Use of Google Glass for Procedural Wireless Vital Sign Monitoring, Surgical Innovation, **23** (2016), no. 4, pp. 366-373.
6. Mitrasinovic, S.; Camacho, E.; Trivedi, N.; Logan, J.; Campbell, C.; Zilinyi, R.; Lieber, B.; Bruce, E.; Taylor, B.; Martineau, D.; Dumont, E.L.P.; Appelboom, G. and Connolly, E.S.: Clinical and surgical applications of smart glasses, Technology and Health Care, **23** (2015), no. 4, pp. 381-401.
7. Farrell, W.A.: Learning Becomes Doing: Applying Augmented and Virtual Reality To Improve Performance, International Society for Performance Improvement, **57** (2018), no. 4, pp. 19-28.
8. Young, H.A.: Scientific Apps are here (and more will be coming), Cytokine, **59** (2012), no. 1, pp. 1-2.
9. Austerjost, J.; Porr, M.; Riedel, N.; Geier, D.; Becker, T.; Scheper, T.; Marquard, D.; Lindner, P. and Beutel, S.: Introducing a Virtual Assistant to the Lab: A Voice User Interface for the Intuitive Control of Laboratory Instruments, SLAS Technology, **23** (2018), no. 5, pp. 476-482.
10. Cortazar, B.; Koydemir, H.C.; Tseng, D.; Feng, S. and Ozcan, A.: Quantification of plant chlorophyll content using Google Glass, Lab Chip, **15** (2015), no. 7, pp. 1708-1716.
11. Feng, S.; Caire, R.; Cortazar, B.; Turan, M.; Wong, A. and Ozcan, A.: Immunochromatographic diagnostic test analysis using google glass, ACS Nano, **8** (2014), no. 3, pp. 3069-3079.

12. Austerjost, J.; Marquard, D.; Raddatz, L.; Geier, D.; Becker, T.; Scheper, T.; Lindner, P. and Beutel, S.: A smart device application for the automated determination of E. coli colonies on agar plates, Engineering in Life Sciences, **17** (2017), no. 8.
13. Strzys, M.P.; Kapp, S.; Thees, M.; Klein, P.; Lukowicz, P.; Knierim, P.; Schmidt, A. and Kuhn, J.: Physics holo.lab learning experience: Using Smartglasses for Augmented Reality labwork to foster the concepts of heat conduction, European Journal of Physics, **39** (2017), no. 3, pp. 1-12.
14. Jacob, F.: Wort, beer, beer-based beverages : collection of brewing analysis methods of the Mitteleuropäische Brautechnische Analysenkommission, Selbstverl. der MEBAK, 2013.
15. Austerjost, J. and Bargholz, M.: LabGlasses repository on GitHub, https://github.com/tciluh?tab=repositories, accessed 24 July 2018.
16. Hunkeler, U.; Truong, H.L. and Stanford-Clark, A.: MQTT-S – A publish/subscribe protocol for Wireless Sensor Networks, 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08), (2008), pp. 791-798.
17. Fovino, I.N.; Carcano, A.; Masera, M. and Trom-Betta, A.: Design and implementation of a secure Modbus protocol, IFIP Advances in Information and Communication Technology, Springer, Berlin, Heidelberg, 2009, pp. 83-96.
18. Forouzan, B.A.; A./Fegan, B. and Chung, S.: TCP/IP protocol suite, McGraw-Hill, 2003.
19. Tarjan, L.; Šenk, I.; Tegeltija, S.; Stankovski, S. and Ostojic, G.: A readability analysis for QR code application in a traceability system, Computers and Electronics in Agriculture, **109** (2014), pp. 1-11.
20. Schalkwyk, J.; Beeferman, D.; Beaufays, F.; Byrne, B.; Chelba, C.; Cohen, M.; Kamvar, M. and Strope, B.: "Your Word is my Command": Google Search by Voice: A Case Study, Advances in Speech Recognition, Springer US, Boston, MA, 2010, pp. 61-90.
21. Saf'yanov, A.S.; Tereshchenko, A.G.; Tereshchenko, V.A.; Yanin, A.M.; Yunak, A.L. and Tereshchenko, O. V.: Information interaction between LIMS and corporate information system elements at enterprise, Automation and Remote Control, **73** (2012), no. 4, pp. 760-764.
22. Funk, M.; Kosch, T.; Greenwald, S.W. and Schmidt, A.: A benchmark for interactive augmented reality instructions for assembly tasks, Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia – MUM '15, (2015), pp. 253-257.
23. Kim, S.; Nussbaum, M.A. and Gabbard, J.L.: Augmented Reality "Smart Glasses" in the Workplace: Industry Perspectives and Challenges for Worker Safety and Health, IIE Transactions on Occupational Ergonomics and Human Factors, **4** (2016), no. 4, pp. 253-258.
24. Rauschnabel, P.A. and Ro, Y.K.: Augmented reality smart glasses: an investigation of technology acceptance drivers, International Journal of Technology Marketing, **11** (2016), no. 2, p. 123.
25. Hoffmann, A.; Kähny-Simonius, J.; Plattner, M.; Schmidli-Vckovski, V. and Kronseder, C.: Computer system validation: An overview of official requirements and standards, Pharmaceutica Acta Helvetiae, **72** (1998), no. 6, pp. 317-325.
26. López, O.: 21 CFR Part 11 – Complete Guide to International Computer Validation Compliance for the Pharmaceutical Industry, CRC Press, 2004.
27. Mann, S.: Eye Am a Camera : Surveillance and Sousveillane, Time, http://techland.time.com/2012/11/02/eye-am-a-camera-surveillance-and-sousveillance-in-the-glassage/, accessed 23 July 2018.
28. World Health Organization: Electromagnetic fields and public health: mobile phones, http://www.who.int/en/news-room/fact-sheets/detail/electromagnetic-fields-and-public-health-mobile-phones, accessed 23 July 2018.
29. Baan, R.; Grosse, Y.; Lauby-Secretan, B.; El Ghissassi, F.; Bouvard, V.; Benbrahim-Tallaa, L.; Guha, N.; Islami, F.; Galichet, L. and Straif, K.: Carcinogenicity of radiofrequency electromagnetic fields, The Lancet Oncology, **12** (2011), no. 7, pp. 624-626.

Received 25 October 2018, accepted 2 February 2019