Tracebook: A Dynamic Checklist Support System

Shan Nan†§, Pieter Van Gorp*, Hendrikus H.M. Korsten†*, Richard Vdovjak‡, Uzay Kaymak*, Xudong Lu§ and Huilong Duan§
*School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
†Department of Anesthesiology and Intensive-Care, Catharina Ziekenhuis Eindhoven, The Netherlands
‡Philips Research, Eindhoven, The Netherlands
§Department of Biomedical Engineering, Zhejiang University, Hangzhou, P.R. China

Abstract—It has recently been demonstrated that checklists can enable significant improvements to patient safety. However, their clinical acceptance is significantly lower than expected. This is due to the lack of good support systems. Specifically, support systems are too static: this holds for paper-based support as well as for electronic systems that digitize paper-based support naively. Both approaches are independent from clinical process and clinical context. In this paper, we propose a process-oriented and context-aware dynamic checklist support system: Tracebook. This system supports the execution of complex clinical processes and rules involving data from Electronic Medical Record systems. Workflow activities and forms are specific to individual patients based on clinical rules and they are dispatched to the right user automatically based on a process model. Besides describing the Tracebook functionality in general, this paper demonstrates the support system specifically on an example application that we are preparing for a controlled clinical evaluation. At last we discuss the limitations of Tracebook.

I. INTRODUCTION

Checklists, sometimes referred to as “safety checklists” or “medical checklists”, have in recent years gained support as vehicles for disseminating evidence-based best practices in clinical medicine, with the aim of improving patient safety [1], [2]. In clinical practice, they are completed in a well-defined process by a responsive team. Usually, team members verbally confirm and discuss all the tasks listed on checklists [3]. Sometimes they are handed to physicians and support staff working on support procedures [4].

Although two independent studies have reported how surgical safety checklist enabled significant reduction in mortality and morbidity [1], [2], practitioners are still reluctant to adopt these checklist in their daily practice due to difficulties while using these checklists [4], [5]. If a good support system could be developed and implemented, it would ease the use of checklist and therefore increase the adoption. However, there is a lack of such a system.

The existing support systems are known to have the following limitations which cause adoption barrier: the lack of reminder mechanisms and the excessive length of checklists [4]. Reminders and alerts have been used widely in clinical decision support systems (CDSSs) [6]. However, alerts regardless of context cause alert fatigue and drain users’ motivation. Moreover, teamwork is not well supported in current checklist support systems.

In this paper, we propose Tracebook, a dynamic checklist support system aiming to support the implementation of checklists in a process-oriented, context-aware way. Tracebook works as an assistant to dispatch checklists to the right person according to clinical workflow and filter the content of checklists based on clinical rules. Moreover, Tracebook bridges participants who work in different phases and have little communication by presenting history checklists in a comprehensive process model. Additionally, by the name Tracebook, we stress the idea of teamwork. Tracebook intends to make clinical processes more traceable and accountable (e.g., by tracking and showing who was responsible for already completed activities). Also, similar to the non-medical social platforms (e.g., Facebook), Tracebook can make a (care) organization more open to its clients (as well as to colleagues from another organizational unit) so that they can know each other, by means of picture logs.

The remainder of this paper is structured as follows: Section II reviews current checklist support systems’ functions and adoption barriers. Section III describes the features of a dynamic checklist support system and walks the reader through these features with an example. Section IV describes two other digital checklist support systems and discusses the major differences. Section V discusses the limitations of this research. Finally in section VI, we give a conclusion to this paper.

II. BACKGROUND

Among all the popular checklists, three of them are well studied and tested, which are the World Health Organization Surgical Safety Checklist (WHO SSC) [1], the SURgical Pa-tient Safety System (SURPASS) [2] and the Surgical-Crisis Checklist [7]. Both paper-based and digitized support systems have been applied to these checklists. In the following subsection, we make a brief review of these checklist support systems’ functionalities and limitations. Then, we discuss the reasons of these limitations.
A. Functionalities of checklist support systems

WHO SSC has been implemented in paper-based support systems. They usually consist of one piece of paper with three blocks. In each block, checkable items that belong to one stage are listed. In each stage, a nurse, an anesthesiologist and a surgeon should stand together to confirm verbally whether the listed tasks have been performed and to document the checked results. The surgical process can only proceed after every item has been checked.

A paper-based support system for SURPASS was firstly introduced in 2008. 18 pages of checkable items are organized per process step. Some of them are optional based on whether or not the patient will receive local anesthesia. Some items are optional based on specific patient conditions. There are also areas to make notes or remarks related to the checks. To improve the usage of the checklist and streamline the checklist process, in 2011, AMC rolled out SURPASS Digital, a web-based checklist support system integrated into the AMC’s hospital information systems. SURPASS Digital provides static web pages of checkable items. The 18 pages of checklist are presented in a sequential order in the system to illustrate the peri-operative process. Users can review history checks in the system [8].

Surgical Crisis Checklist also depends on a paper-based support system. The system relies on color coding and numbering for organizing the checkable items of 12 anticipated emergency situations. When a crisis happens, the OR staff can find the correct checklist more conveniently than without this organization technique. The checklists essentially have turned evidence-based best practices into actionable steps for dealing with critical scenarios. People are encouraged to confirm and discuss each step verbally. Signatures of the participants are not needed for Surgical Crisis Checklist.

B. Insufficiencies

Though WHO SSC is wide-spread, the adherence to WHO SSC is reported low. Based on different statistical methods, the completion and compliance of WHO SSC are reported differently. A multi-center study [5] indicates that the completion rate and compliance rate are 61% and 90.2% respectively. Another single-site study [9] shows that the completion rate is 85% and the compliance rate is 69%. Both of these studies expected a 100% completion and compliance. Barriers are identified in these studies, which are redundancy in the checkable items, poor communication, time consumption, inappropriate timing, identification of the role and responsibility of staff as well as users’ attitude [5].

While implementing SURPASS in AMC, 65% of the surgeons completed checklist almost always, and only 35% of the anesthesiologists did so. Surgical staff was interviewed to find out the reasons. The most frequent reasons were “forgotten” with 66%, “logistics” with 45%, “lack of time” with 34% as well as “motivation” and “others” with 11%. Integration into hospital information system, providing electronic checklist and making checklist shorter were suggested to improve the compliance of SURPASS [4].

The Surgical Crisis Checklist has been evaluated in a simulation-based study. Participants of the evaluation claim that they would like to use this checklist in their work. However, the implementation in real clinical situations has not been reported yet [7].

C. Static nature of existing checklist support systems

We conclude the inefficiencies as “too static” in two respects.

1. Checklist support systems are static in terms of the process. Here, the process refers to the collection of checklist-related activities and the order between them. Currently, checklist support systems have a poor division of responsibilities of the people involved. Thus, users may forget or feel unclear about their roles and fail to perform checks at the appropriate time.

2. Checklist support systems are static in terms of the context. Here, the context refers to the data related to the condition of the patient (e.g., diagnosis, co-morbidities, laboratory tests, prescription, demographic information, etc.). In current checklist support systems, items in the checklist are the same for every patient, regardless whether they may need significantly different concerns. Thus, checklists are excessively long. Based on the aforementioned studies on alert fatigue, it is clear that a lack of prioritization support can drain users’ motivation.

As a result, solving the “static” problems is the key to developing a well-accepted checklist support system. Comparing with current checklist support systems, we characterized such a system as a dynamic checklist support system, which should be process-oriented and context-aware.

III. Tracebook: A Dynamic Checklist Prototype

In this section, we describe a dynamic checklist support system, Tracebook, which supports all the checklist functionalities stated in the above section and provides dynamic features. An additional benefit of Tracebook is that it can make clinical processes more transparent, traceable and accountable by tracking and showing the responsible persons of already completed activities.

A. Features

Characterized as process-oriented, Tracebook has the following features.

Tracebook supports both predefined processes and ad-hoc processes. Both predefined processes like the process in SURPASS and ad-hoc processes like the process in the Surgical Crisis Checklist are supported.

Tracebook supports complex flows. Complex flows which have events, conditional branches, gateways, timers, etc. can be executed in Tracebook.

Tracebook gives an overview of clinical processes. Tracebook logs each user’s choices for a patient as well as the context when they made the decision.

Tracebook is event driven. Event driven means that certain activities can be triggered by clinical events automatically.

Tracebook can assign checklists to both a role and a specific user automatically. Tracebook can distribute tasks to
a certain group of user by their role or a particular user based on rules.

Characterized as context-aware, Tracebook has the following features.

**Context-based task filtering and assignment.** Tasks for a certain scenario can be filtered and assigned to the user according to patient data.

**Context-based data and information supplementary.** For each task, based on particular rules, patient data or additional information related to the task can be acquired by users at the time they need it.

**Triggering other tasks automatically.** With the context and/or decisions users make, Tracebook can check whether preconditions of a task are met or successor actions should be started.

**Note and remark on each item.** Tracebook allows a user to make a note and/or a remark on each item.

### B. Walk-through of an OR-ICU checklist

We use an OR-ICU checklist, which is adopted from SURPASS and contains ad-hoc checkable items like Crisis Checklist, and an imaginary patient, John Doe, to walk through the features of Tracebook.

The OR-ICU checklist has three stages, pre-operative check, time-out in OR and admission to the ICU. When the operation is planned, the pre-operation checklist should be checked by a surgeon, a nurse and an anesthesiologist respectively in their offices. When the patient is sent to the OR, an anesthesiologist, a surgeon and a nurse work together to perform a verbal confirmation. Further, when the patient is admitted to the ICU, an ICU nurse and a doctor need to check the admission to the ICU checklist. The whole process is illustrated in Fig. 1. Besides the scheduled checks, ad-hoc checklists for unscheduled events are also provided.

![Fig. 1: OR-ICU checklist process.](image)

When the operation is planned, the EMR system sends an event to Tracebook. The workflow engine inside of Tracebook catches this event and starts a new OR-ICU checklist process. Activities in the process are assigned to the predefined roles by the workflow engine. The users can see the checklist immediately (e.g., emails are sent to participants’ mailboxes). Alternatively, they can login to Tracebook later to find a list of checklists that they can check.

When a user chooses a checklist, he or she will confirm the listed items and mark them. Instead of giving a long list that contains irrelevant items for the specific patient, in Tracebook, contextual clinical rules are used to filter existing items based on specific patient data in the EMR. Only patient-specific checks are displayed. Here for example (in Fig. 2, red font), two clinical rules “IF Warfarin Prescribed=true AND INR≥4.0 THEN return ‘Patient INR too high (INR value) noticed’” and “IF Renal Insufficiency=true THEN return ‘Renal insufficiency noticed’” are predefined in Tracebook. The rule engine in Tracebook acquires patient data from the EMR and executes these rules. As a result, two more check items, “Patient INR too high (4.1) noticed” and “Renal insufficiency noticed” are provided by Tracebook. Another added value to the checklist provided by Tracebook is the quick access to patient data and multimedia resources. As illustrated in Fig. 2, when the user clicks on the underscored text, Tracebook will display an information pop-up to show the most recent data or lead to additional resources like clinical guidelines. A note or remark can be attached to each check. After checking and remarking, the checklist is submitted.

![Fig. 2: Example of a context-aware checklist.](image)

When the operation is started, the medical team is assembled together in the OR to discuss the patient situation. The time-out checklist is assigned to the users who did the check in the previous stage by the workflow engine. After the operation, John is sent to the ICU. There, an intensivist and a nurse need to know his conditions, including the operation details. In Tracebook, there is a comprehensive view in Business Process Modeling and Notation (BPMN) format customized with participants’ photos (in Fig. 2). Tracebook queries from the workflow engine that which tasks will be performed, which tasks should be performed at this moment and which tasks have been performed. Tasks that should be performed at this moment are highlighted with a red rectangular. Tasks that have been performed are marked with the performers’ photos. Clicking on each photo, the current user can see what the others have checked and remarked. Those patient data and additional resources are kept together with the history checklist.
In this way, warnings and important remarks which are easily ignored in the EMR and the other information systems are kept together. Successor users can know from that why previous users made a particular decision. Another important feature is that the responsible person’s contact information is placed together with their checks. When users have any doubt, they can directly call the one who made the check.

Unfortunately, John is infected with vancomycin-resistant enterococci (VRE). This is an unexpected situation that can not be planned. There is an “Ad-hoc checklist” page in Tracebook. Users can see the completed ad-hoc checklists which have been performed in previous stages for this patient. By typing keywords, users can find the VRE checklist very easily and perform their own checks.

This prototype makes the whole process transparent and traceable. This guarantees a better team communication and promotes team building. Also, for each specific role involved in the process, they have a well defined, highly patient-specific checklist to confirm at the right time.

IV. RELATED WORK

SURPASS Digital, as we described in Section II, made impressive progress in usability and integration with the EMR [8]. However, there are several major differences between SURPASS Digital and Tracebook. Firstly, SURPASS Digital allows users to pick up a checklist in a list while Tracebook can send a checklist to a specific group of users at the right time. Secondly, in SURPASS Digital, items in each checklist are unchangeable in the runtime while Tracebook provides personalized items for each patient. Finally, SURPASS Digital provides a sequential view of the completed tasks while Tracebook has a comprehensive and straightforward view in BPMN format.

In 2012, Avrunin et al. [10] proposed the smart checklists for human-intensive medical systems. Their system can guide process actors in identifying and responding to exceptional or hectic circumstances, help with process deviations and help assure deadlines are met. Checkable items can change dynamically taking into account the history of the process execution, summaries of past execution, and projections of possible future execution of the current process. The smart checklist system focuses on process execution and changes the checklist items dynamically, based on the process context. However, Tracebook dynamically changes the items by the context regarding to patient state. Additionally, Tracebook stresses the accountability of clinical process.

V. DISCUSSION

The aim of Tracebook is to improve the adoption of checklists. However, usability is an important factor that influences the adoption of Tracebook. This problem can be settled by a series of usability studies. Currently, we are in the process of setting up a first usability study.

Another important aspect of clinical information systems is security. Tracebook has not stress too much on it yet. However, we argue that Tracebook is only used in the hospital intranet. The service itself is not on a public Internet accessible port and we argue that Tracebook is only used in the hospital intranet. However, Tracebook has not stress too much on it yet. However, we argue that Tracebook is only used in the hospital intranet. The service itself is not on a public Internet accessible port and would benefit from the existing intranet security mechanisms (firewall, etc).

VI. CONCLUSIONS

In this paper, we discuss the advantages and limitations of current checklist support systems and present an innovative dynamic checklist support system: Tracebook. Although checklists show their unique power in promoting patient safety by providing users a clear view of critical tasks and helping with the multidisciplinary communication, they are not implemented successfully due to their static supporting systems. We propose a dynamic checklist support system which is process-oriented and context-aware to improve this. General features of such a system are discussed. And a prototype of an OR-ICU checklist has demonstrated the feasibility of our approach.

Future work will focus on the deployment, implementation and evaluation of this system. We are planning to implement it in a controlled OR-ICU setting and measure its impact on the checklist acceptance.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Brain Bridge Project sponsored by Phillips Research.

REFERENCES


