

# Smart Checklists to Improve Healthcare Outcomes

Stefan C. Christov  
Quinnipiac University  
Hamden, CT 06518, USA  
schristov@quinnipiac.edu

George S. Avrunin  
University of Massachusetts  
Amherst, MA 01003, USA  
avrunin@cs.umass.edu

Heather M. Conboy  
University of Massachusetts  
Amherst, MA 01003, USA  
hconboy@cs.umass.edu

Lori A. Clarke  
University of Massachusetts  
Amherst, MA 01003, USA  
clarke@cs.umass.edu

Nancy Famigletti  
University of Massachusetts  
Amherst, MA 01003, USA  
nfam@cs.umass.edu

Leon J. Osterweil  
University of Massachusetts  
Amherst, MA 01003, USA  
ljo@cs.umass.edu

## ABSTRACT

This paper presents an approach for automatically generating Smart Checklists—context-dependent, dynamically updated views of on-going medical processes based on current activities and previously validated process models of best practices. This approach addresses not only nominal scenarios but includes guidance when exceptional situations arise. The framework for creating these checklists is described, along with an example and discussion of issues.

## CCS Concepts

•Applied computing → Health care information systems;

## Keywords

Process guidance, checklist, healthcare

## 1. INTRODUCTION

There has been considerable evidence that the use of checklists in healthcare, as well as many other domains, reduces errors. Although several studies over the years have documented the prevalence of errors in healthcare processes, there is considerable resistance to using checklists, in large part because they are usually static, lack context, and only focus on normative situations (e.g., [9]). To address these shortcomings, we are investigating an approach for generating dynamic, context-sensitive checklists that contain detailed information about the recommended ways to perform a medical process for a wide range of scenarios, including scenarios when exceptional situations arise.

The proposed approach is based upon detailed and carefully validated process models. In earlier work, we demonstrated how these models can be analyzed using model checking and fault tree analysis to find defects and vulnerabilities in processes [3]. Applying process modeling and analysis to

complex medical processes improved process understanding and, in one study, led to a substantial reduction in the number of chemotherapy errors that reached the patient [10].

In the work described here, we take the next step, and use these validated process models to drive the generation of context-sensitive checklists during the actual performance, or execution, of a process. Ideally, these *Smart Checklists* are dynamically updated by interpreting information obtained from the executing process in terms of a validated process model. To date, we have developed technology to generate Smart Checklists that takes advantage of the hierarchical representation and abstraction in the process model to visualize process execution information and that uses process execution events to update the generated Smart Checklists during a process execution. Based on feedback from medical professionals and human factors experts, we have designed an initial Smart Checklist user interface (hereafter, for brevity we use “Smart Checklist” to refer to the Smart Checklist user interfaces generated by the framework). In the future, we plan to evaluate the proposed approach in human simulation studies to determine how it impacts the safety and efficiency of some healthcare processes.

The next section presents a description of the framework for generation of Smart Checklists and the technology it is built upon. Section 3 provides an example Smart Checklist for a part of a blood transfusion process. Section 4 discusses some of the more interesting issues, and Section 5 summarizes the current status and directions for future work.

## 2. APPROACH

Figure 1 illustrates at a high-level the framework for generating Smart Checklists [4]. A *Smart Checklist Generator* uses as input a *process model* and the events captured by a *Process Execution Monitor* to generate and continually update Smart Checklists that guide human process performers during the execution of a process.

The *process model* represents the recommended ways to perform the process based on best-practice, guidelines, local policies, etc. To be a useful basis for guidance of medical professionals, this model needs to capture the complexities of the actual medical process. Thus, in addition to capturing the process activities and their orderings, the model also captures information about the human and automated process performers responsible for completing these activities and the artifacts used and produced throughout the process. The process model also contains information about the ex-

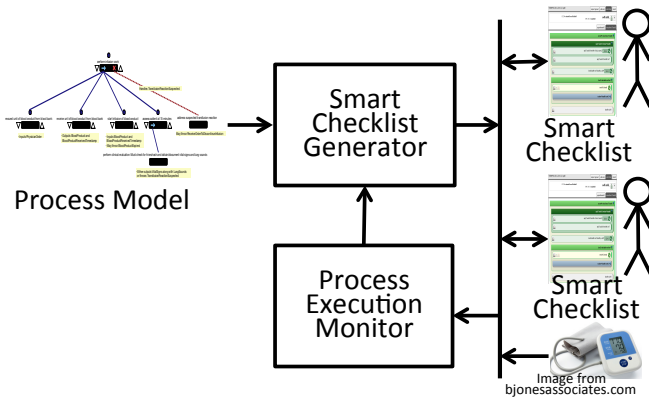


Figure 1: Smart Checklist Framework.

ceptional situations that might arise and the recommended responses to these exceptional situations. Providing support to human process performers during exceptional situations is particularly important since errors are especially likely to occur during such situations [11].

In the current prototype of the Smart Checklist Framework, process models are specified in the Little-JIL process modeling language [5], because Little-JIL’s rich semantics are suitable for representing many of the complexities of medical processes. Furthermore, Little-JIL has formal semantics allowing Little-JIL process models to be automatically analyzed and executed.

The *Process Execution Monitor* is responsible for capturing events of interest, such as the completion of a process activity or the recognition of an exceptional situation, and for relaying these events to the Smart Checklist Generator. Events of interest could be generated by medical devices (represented by the device for measuring blood pressure in Figure 1) or by manual input by human process performers via a Smart Checklist.

The *Smart Checklist Generator* uses the execution events captured by the Process Execution Monitor to determine the execution state of the process based on the process model and provides a visualization of different aspects of that execution state via the Smart Checklists. This visualization includes past activities that have already been performed, current activities that are being performed, as well as potential next activities based on the recommended process captured in the model. Activities can be shown along with associated information, such as artifacts needed, process performers in charge of completing the activities, and exceptional situations that could arise or have already arisen.

The *Smart Checklists* produced by the Smart Checklist Generator are based on a set of predefined activity visualization templates that are parameterized by various kinds of process information, such as activity name, activity execution status, artifacts, and exceptional situations. These activity visualization templates are instantiated as a process is being executed and the collection of these instantiated templates forms a Smart Checklist.

### 3. EXAMPLE

We have implemented a prototype of the Smart Checklist Framework shown in Figure 1 and applied that proto-

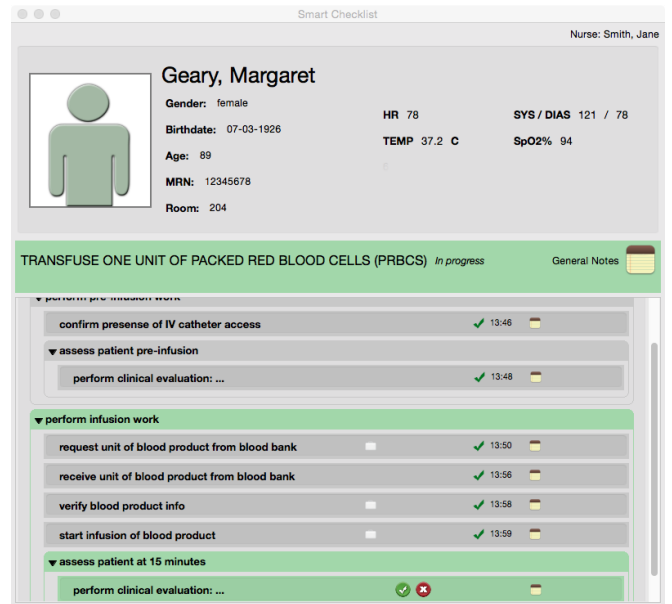


Figure 2: Smart Checklist guiding a nurse through the transfusion of one unit of packed red blood cells.

type to several case studies in the healthcare domain: blood transfusion [6], cardiac surgery [7], and infusion therapy [1]. This prototype automatically generates Smart Checklists that guide medical professionals as they perform a medical process. Figure 2 shows the generated Smart Checklist that guides a nurse named Jane Smith through the process of transfusing one unit of packed red blood cells (PRBCs) for a patient named Margaret Geary. This Smart Checklist identifies the medical professional (in the top right corner), includes some patient-specific information (shown in the top half of the figure), and provides the context-sensitive process information (shown in the lower half of the figure).

The patient-specific information shown in this example includes the patient identifiers such as the patient’s photo, full name, date of birth, and medical record number (shown on the left) as well as some of the patient’s physiological data (shown on the right). These values are provided by integration with the Open Integrated Clinical Environment [2]. The current set of information included and its layout was customized for a nurse performing this process. Since different types of information and layouts may be appropriate for other process performers or for other processes, the Smart Checklist Generator supports flexible reconfiguration of the visual representation.

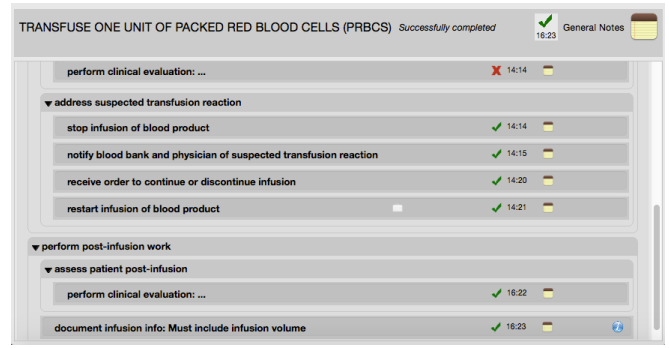
As part of the blood transfusion process, the nurse has already verified the patient’s identify and has confirmed that the patient signed a consent form before performing the following portion of the process. The context-sensitive process information at this stage includes the process header (the green rectangle with “TRANSFUSE ONE UNIT ...”) and the dynamic listing of the process activities (shown in the remainder of the lower half). The process header shows the process name (the capitalized text), the process execution status (“In progress”), and the general notes button (denoted by a yellow and black icon at the far right) that the nurse would click to document general information about the process execution. The dynamic listing of the process activities

shows that the past activities (denoted by the gray backgrounds) are *review patient history* (scrolled up and thus not visible in Figure 2) and *perform pre-infusion work* and that the current activity (denoted by the green background) is *perform infusion work* along with its subactivities. The nurse has successfully completed some of these subactivities (*request unit of blood product from blood bank*, *receive unit of blood product from blood bank*, *verify blood product info*, and *start infusion of blood product*) and is currently engaged in the activity *perform clinical evaluation: Must check for hives/rash and obtain/document vital signs and lung sounds* (the ellipses in the corresponding line in the figure are there to save space). This clinical evaluation involves the nurse assessing the patient’s condition to detect a potential transfusion reaction. If such a reaction is not suspected, then the nurse would click on the button with the checkmark to indicate that the activity was successfully completed. Otherwise, the nurse would click on the button with the “X” to indicate that the activity was not successfully completed because problems arose. The nurse could also use the notes button (to the right of the button with the “X”) to document this decision.

Once the nurse has indicated whether the activity is successfully completed or not, the Smart Checklist Generator dynamically updates the visualization of that activity and adds visualizations of the next activities that need to be performed. The background color of the activity just performed will change to gray, the execution status of that activity would change to “successfully completed” (indicated by a green checkmark) or “not successfully completed” (indicated by a red “X”), and a timestamp indicating when the activity was performed would appear. The next activities will depend on the current activity’s execution status and the process model.

If the nurse is not concerned about a transfusion reaction and clicks the checkmark button, the Smart Checklist will update to show that the next activity is *perform post-infusion work*. On the other hand, if the nurse clicks on the “X”, the checklist will bring up a dialog box in which the nurse indicates that a transfusion reaction is suspected. Then this Checklist would be updated to show the appropriate activities for responding to this exceptional situation. The hospital policy, as reflected in the process model, calls for the nurse to stop the infusion and notify the blood bank and physician. The physician then determines whether to continue the infusion and enters an appropriate order.

Figure 3 shows the activities portion of the final Smart Checklist, showing that the nurse stopped the infusion and notified the physician and blood bank. After the physician entered an order to continue the infusion, the nurse restarted the infusion and, after the infusion completed, performed the post-infusion work. In this figure, all of the remaining activities were successfully completed, thus they have all turned gray. Additionally, the process header has also turned gray and its execution status has changed from “In progress” to “Successfully completed.” The Smart Checklist Generator allows the process performer to customize what process information is displayed (e.g., any intermediate activities such as *perform post-infusion work* can be shown or hidden) and how process information is displayed (e.g., the background colors for the past/current activities can be selected). Additionally, the process model can be annotated with other context-sensitive information such as natural lan-



**Figure 3: Final Smart Checklist after the nurse has successfully completed the transfusion**

guage descriptions for some activities. If any activity is annotated with such a description (e.g., activity *document infusion info: Must include infusion volume*), then that activity would display an info button (indicated by a blue icon with a white “i”). The process performer could use the info button to bring up the activity’s description.

#### 4. DISCUSSION

Determining exactly which information about the patient and the activities being performed should be shown and what visualizations should be used to effectively convey this information to process performers depends on multiple factors, such as characteristics of individual process performers (e.g., role and level of expertise) and the current process execution state (e.g., if exceptional situations arise, more detailed instructions might be provided using color schemes indicating a potential problem). The sample Smart Checklists shown in section 3 illustrate the capabilities of the Smart Checklist Framework, but decisions about what information to show and how it should be shown on Smart Checklists requires further investigation, involving use of human factors approaches, user studies to empirically evaluate alternative visualizations, and consideration of visual metaphors with which process performers in a given domain are already comfortable. The Smart Checklist Framework presented in this paper is designed to support customization of the generated Smart Checklists, and as more is learned about what information is useful for which users, the Smart Checklists can be easily adjusted.

Since the process state visualization (i.e., the generated Smart Checklists is based on an underlying process model, the choices made in this process model affect the kinds of process execution state information that can be shown. For instance, the hierarchical decomposition of the process affects what is displayed. If this hierarchical decomposition is not consistent with the mental model of process performers, the resulting Smart Checklists might be difficult for them to understand. We have started to develop guidelines for creating process models to help address this issue. Example guidelines would be to recommend that activity names use agreed upon terminology and that process activities are grouped together into a higher-level activity if that grouping corresponds to how process performers view these activities.

Monitoring an executing process and recording events of interest is an essential component of the Smart Checklist Framework. This monitoring could be challenging, but it

should be facilitated by the increasing use of electronic health records (EHRs) and medical devices with digital inputs. For some complex procedures, medical scribes are used to help with the creation of documentation [8] and thus could capture process activities as they are being performed. In some cases, voice recognition systems can be used to capture voice commands by medical professionals.

We performed a preliminary evaluation of the current implementation of the Smart Checklist Framework and, in particular of the generated Smart Checklists, by presenting them to a focus group of medical professionals. The Smart Checklists were well-received, with medical professionals recognizing the potential of the approach to reduce errors, improve process documentation, and serve as a training aid.

The medical professionals confirmed that it would be useful to show past process activities in addition to the activities that need to be currently performed. They thought that being able to access the process execution history would be particularly helpful during hand-offs. Some of the medical professionals suggested that the Smart Checklist Framework should support different modes, depending on the level of expertise of the process performers (e.g., less detail for expert than for novice process performers). The medical professionals pointed out, however, that a detailed list of possible problems that might arise during the performance of an activity would be useful even for expert process performers.

The current Smart Checklists are from the perspective of a single process performer. The medical professionals confirmed our expectations that in some situations it would be useful to access the tasks of other performers as well to get a better sense of the overall execution state of the process and also suggested that it might be useful if the Smart Checklist Framework facilitated communication between different process performers working together as a team.

The medical professionals found the activity hierarchy shown on the Smart Checklists useful as a way to visualize the structure of the process. Some of them mentioned that this hierarchy would be particularly useful when browsing through the process execution history as this history could be potentially long, but the hierarchy may not be necessary for the current activities that are in progress, as usually there will be a small number of such activities.

## 5. CONCLUSION AND FUTURE WORK

The preliminary evaluation of the Smart Checklist Framework has been promising. Medical professionals and human factors experts who have reviewed it have recognized the potential of that framework to reduce errors, improve process documentation, and serve as a training aid. They have also suggested areas for improvement in terms of choosing what information to show process performers and how to visualize this information. In particular, it would be useful to be able to customize the generated Smart Checklists based on the role and the amount of experience of process performers, to support both individual and team-based Smart Checklists, and to support team communication. The issue of capturing process execution events needs to be further investigated by potentially integrating the Smart Checklist Framework with medical devices and EHRs. Finally, the Smart Checklist approach needs to be thoroughly evaluated, initially in simulated but eventually in real clinical settings.

Providing information about the execution state of a sys-

tem (e.g., dashboards in cars or in pilot cockpits) is a widely used approach to assist humans who operate within complex systems. The Smart Checklist Framework aims to support such an approach for complex processes. We believe that recent advances in process modeling and analysis as well as the increasing use of technology in healthcare create an environment for implementing such an approach, which could in turn lead to improving healthcare outcomes.

## 6. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Awards IIS-1239334 and CMMI-1234070. The authors gratefully acknowledge the contributions of Elizabeth Henneman, Jenna Marquard, and Julian Goldman.

## 7. REFERENCES

- [1] MD PnP (Medical Device Plug-n-Play Program), Clinical scenario 1 for patient controlled analgesia. [www.mdnp.org/MD\\_PnP\\_Program\\_\\_Clinical\\_S.html](http://www.mdnp.org/MD_PnP_Program__Clinical_S.html).
- [2] MD PnP Program, Open Integrated Clinical Environment (Open ICE). [www.openice.info](http://www.openice.info).
- [3] G. S. Avrunin, L. A. Clarke, L. J. Osterweil, et al. Experience modeling and analyzing medical processes: UMass/Baystate medical safety project overview. In *1st ACM Int. Health Informatics Symposium*, pages 316–325, 2010.
- [4] G. S. Avrunin, L. A. Clarke, L. J. Osterweil, et al. Smart checklists for human-intensive medical systems. In *Workshop on Open Resilient human-aware Cyberphysical Systems*, pages 1–6, June 2012.
- [5] A. G. Cass, B. S. Lerner, S. M. Sutton, et al. Little-JIL/Juliette: A process definition language and interpreter. In *22nd Int. Conf. on Software Engineering*, pages 754–757, 2000. ACM.
- [6] S. C. Christov, G. S. Avrunin, L. A. Clarke, et al. A benchmark for evaluating software engineering techniques for improving medical processes. In *2010 ICSE Workshop on Software Engineering in Health Care*, pages 50–56, 2010. ACM.
- [7] H. M. Conboy, J. K. Maron, S. C. Christov, et al. Process modeling of aortic cannulation in cardiac surgery: Toward a smart checklist to mitigate the risk of stroke. *Fifth Workshop on Modeling and Monitoring of Computer Assisted Interventions*, 2014.
- [8] K. Hafner. A busy doctor’s right hand, ever ready to type. *The New York Times*, January 12 2014.
- [9] L. A. Hassell, A. V. Parwani, L. Weiss, et al. Challenges and opportunities in the adoption of College of American Pathologists checklists in electronic format: perspectives and experience of reporting pathology protocols project (RPP2) participant laboratories. *Archives of Pathology and Laboratory Medicine*, 134(8):1152–1159, 2010.
- [10] W. Mertens, S. Christov, G. S. Avrunin, et al. Using process elicitation and validation to understand and improve chemotherapy ordering and delivery. *The Joint Commission J. on Quality and Patient Safety*, 38(11):497–505, 2012.
- [11] S. J. Spear and M. Schmidhofer. Ambiguity and workarounds as contributors to medical error. *Annals of Internal Medicine*, 142(8):627–630, 2005.