Using Temporal Logic to Verify Blood Supply Chain Safety

Noha Hazzazi, Bo Yu, Duminda Wijesekera, and Paulo Costa
{nhazzazi|byu3|dwijesek|pcosta}@gmu.edu.
CARE, Volgenau School of Engineering, George Mason University, Fairfax, VA, 22030

Abstract—The need for transfusion blood increases each year. The Food and Drug Administration (FDA), the American Association of Blood Banks (AABB) and standardization bodies that hold jurisdiction in other countries continuously update blood transfusion safety mandates. Verifying blood bank processes for safety takes labor and time. We automate this verification process by modeling the workflows of the blood processing supply chain, extracting FDA and AABB requirements as Temporal Logic formulas and verifying that the workflows comply with the mandates. We also show how this process can seamlessly integrate into an Electronic Medical Record System.

Keywords: Blood bank, Verification, Temporal Logic

1. Introduction

Blood transfusion is a usual procedure for patients suffering major surgeries, injuries or illnesses such as hemolytic anemia. Each year, almost five million Americans need a blood transfusion [1]. Transfusions are administered in 137 countries serving a population of 3.1 billion worldwide [2]. The healthcare industry faces increasing challenges for ensuring safe, quality blood transfusions, including:

- Risk of transmission of infection through unsafe blood or blood products
- Technical and clerical errors in the processing and testing of blood
- Errors in the administration of blood or blood products.

The World Health Organization (WHO), in its Strategy for Blood Safety and Availability for Improving Patient Health and Saving Lives, says that cumulatively 58% of transfusion fatalities are caused by transfusion errors and related issues [3]. Transfusions are the tip of a long, well-regulated supply chain that consists of collecting blood from donors; testing and decomposing blood into components such as red blood cells, platelets etc.; preserving them in regulated conditions (e.g. adding preservatives and refrigeration); and conducting the transfusion based on physicians’ orders. The complexity of this chain results in many potential causes of unsafe blood transfusions.

In past decades, many researchers and organizations transferred efforts and resources from creating standards to adopting new IT technologies that would improve blood transfusion safety. For example, the WHO as well as the AABB among others defined blood safety standards [2]. In the US, blood banks must follow FDA blood safety standards [4]. To make the transfusion chain safer, many health organizations adopted blood bank systems or other computerized systems, such as the TANGO Automated Blood Bank System, SIBAS [5]. Adverse effects of transfusions are mostly examined and appropriate corrective actions are taken routinely.

In most countries, blood product safety relies on manual documentation that are not linked together, such as paper-based or computerized forms scattered throughout the blood supply chain. Blood bank systems have moved from manual documentation to complex systems with improved functionality, such as inventory control and policy management [6], [7]. However, current systems assist in collecting the necessary documentation in electronic formats but do not provide electronically verifiable safety assurance in each step and for the supply chain as a whole as specified by each locality’s safety standards. We are unaware of any system that uses safety-verified workflows in administering blood products. Providing a method to do so is the objective of this paper.

Observed errors committed by blood bank staff, especially by limited staff during night shifts who nonetheless handle normal workload [8], compels electronically enforceable blood safety verification. We propose to introduce a safety verification method that can be built into existing Electronic Medical Record and blood bank systems using established engineering practices. Our approach includes formally modeling the supply chain of blood as a workflow, specifying the safety regulations in a logical format, and verifying that the appropriate components of the entire supply chain satisfy all applicable safety regulations. This paper shows an early prototype of our proposal.

This paper is divided as follows. Section 2 describes our process to verify the safety of the supply chain of blood, which starts with modeling the supply chain as a workflow system, which states are formed by the input/output of participants - or processes working on behalf of them - resulting into succeeding states (Figure 2). Then, in Section 3 FDA and AABB safety regulations are translated to Temporal Logic statements that have to hold in particular states of the blood processing supply chain. We use a model checker to verify these statements and whether the states on the supply chain satisfy the appropriate Temporal Logic Formulas. Section 4 shows how this entire process is transparent to an EMR system using an open source tool. Finally, Section 5 describes related work and Section 6
presents our conclusions.

2. The Blood Safety Workflow

Figure 1 shows our waterfall model of collecting information, remodeling, validating, verifying, and implementing the blood safety workflow-enforced blood bank system.

This section covers our model of the high level view of the FDA regulations and AABB standard. We interviewed blood bank professionals and created the complete workflow shown in Figure 2. The diagram shows all composite processes in filled color, with their associated decomposition contouring the same color. The main processes are (1) Registration (2) Physical Exam (3) Collection (draw blood) (4) Post Donation, and (5) Transfusion. All processes, from beginning of donation until transfusion, abide to the safety requirements set forth by the FDA and AABB. To the best of our knowledge, there is no published blood bank workflow that has been verified against the FDA and AABB requirements.

The first main process is Registration (four dark cyan color boxes at the top left of Figure 2), which captures donor demographics prior to Physical Exam. Its specification requires the donor to be identified, registered, and provided with educational material, all stemming from a safety measure mandate of tracking all processes from registration until patient follow-ups [11]. For instance, the detailed procedure in the "Identify & Verify Donor" step (second box from top) involves identifying the donor by matching its ID photo with the person, and verifying its expiration date and validity.

The second step is the Physical Exam (Aqua-colored boxes in Figure 2), which determines the donor’s suitability for donating blood. This composite process depends on the type of donation; as there are separate regulatory requirements for whole blood or apheresis donation. To illustrate the process we will focus on high level whole blood suitability.

Whole blood donors are required to check their hemoglobin level, blood pressure, temperature, arm diseases, and donation interval. These measures are specified by the FDA to ensure donor and blood safety. For instance, in CFR 640.3 the FDA specifies (1) normal temperature and (2) systolic and diastolic blood pressures are within normal limits. However, no quantifiable definition of normal is provided. For this reason, many blood banks apply other standards (e.g. AABB) in support to the FDA’s specification. The AABB, for example, defines thresholds of $\leq 37.5^\circ C$ for temperature, $\leq 180mmHg$ for Systolic pressure, and $\leq 100$ mm Hg [9] for Diastolic pressure. Regarding the other requirements for whole blood, FDA specifies that I donor must (a) have its hemoglobin $\geq 12.5$ grams, (b) be free from skin diseases or arm scars, and (c) from diseases transmissible by blood.

In our model, we defined the properties of the physical exam/suitability process based on these requirements.

Once eligibility is established, the donor is ready for the Collection (Cadet Blue-colored boxes in Figure 2). Collections also has specific requirements for regular donations (whole blood) and apheresis, which covers aspects such as unit identification, donor arm preparation, temperature control. Whole blood and apheresis require each unit to be linked to a donor via his/her donor ID. In CFR 640.4, FDA states that the skin of the donor at the site of phlebotomy shall be prepared thoroughly and carefully by a method that gives maximum assurance of a sterile container of blood and a proper donor arm preparation to prevent blood contamination. Temperature control after the blood unit is collected is another requirement, with the FDA defining that collected blood must be placed in storage at a temperature between 1 and 6°C immediately after whole blood collections. The FDA CFRs 640.4, 640.14, 640.22, 640.32 state information about whole blood collection but are not clear on either apheresis collection or on the volume of blood collected.

For this reason, we resorted to the AABB specification for the volume of blood required on whole blood and apheresis collections (5.6.4 and 5.5.3.5, respectively, in [12]).

After collection, the next step is post donation consisting of blood processing and donor monitoring (Fuchsia-colored boxes in Figure 2). As stated by blood bank professionals, the donor is monitored to ensure he/she did not develop any adverse reactions to ensure his/her safety. It is mandated to collect extra samples of the donated blood unit at the time of collection for testing (modeled in Figure 2). CFR 640.5 states that all laboratory tests shall be made on a specimen of blood taken from the donor at collection time.

CFR 640.5 also specifies the main tests required to process the blood unit, as well as its acceptable results. This includes tests such as serological test for syphilis, blood grouping and Rh tests. Our model in Figure 2 computes the FDA mandated requirement for the method used for blood grouping. This is stated in CFR640.5 part (b) as at least two blood group tests shall be made and the unit shall not be issued until grouping tests by different methods or with different lots of antisera are in agreement. Regarding Rh grouping, most blood banks currently carry out both the blood grouping and Rh in parallel, which is also captured in our model.

Further tests must be done for sterility, blood unit inspection, and test for communicable disease based on CFR 640.5 (Sections d, e, f). Blood used for transfusion should not be tested for sterility, but we did not model this aspect since our focus this time was on usable blood. All donated blood shall be visually inspected and the FDA states that it
should be inspected at storage and prior to using. The list of communicable diseases required to check is covered in CFR 610.40 and included in our model.

Once blood is collected, tested and stored, the next step is Transfusion (Midnight Blue-colored boxes in Figure 2), in which an authorized health professional requests blood transfusion, transfuses the compatible blood and monitors the patient. The AABB have specified the details required to request for blood transfusion (cf. parts 5.11.1.1 and 5.28.2 [12]) and for checking blood group ABO/Rh (cf. parts 5.12 and 5.12.1 [12]) and for checking blood group ABO/Rh (cf. parts 5.12 and 5.12.1 [12], which are both captured as processes 610.40 and included in our model.

3. Temporal Logic Translation

In this section, we show how we translate FDA and AABB mandated safety requirements. The verification process consists of defining the predicates, states, and properties in order to specify and verify safety requirements. We start this step by translating the FDA and AABB safety requirements into Linear temporal Logic (LTL) statements.

3.1 Syntax

In order to specify LTL syntax, let $VAR = \{ \vec{x}_i ; i \geq 0 \}$ be a set of variables, $CONST = \{ \vec{c}_i ; i \geq 0 \}$ be a set of constants and $\Phi = \{ p_i : i \geq 1 \}$ be a set of atomic predicate symbols.

We say that $p_i(\vec{x}_i), p_i(\vec{x}_i) \land p_k(\vec{x}_k), p_i(\vec{x}_i) \lor p_k(\vec{x}_k), \neg p_i(\vec{x}_i), p_i(\vec{x}_i) \land p_k(\vec{x}_k), \forall \vec{x}_i p_i(\vec{x}_i), \exists \vec{x}_i p_i(\vec{x}_i), p_i(\vec{x}_i) \rightarrow p_k(\vec{x}_k)$ and $p_i(\vec{x}_i), \diamond p_i(\vec{x}_i), X p_i(\vec{x}_i)$ (sometimes this next-time operator $X$ is written as $\Box p_i(\vec{x}_i)$) are predicates. Following standard convention, a fully instantiated predicate is one in which all variables are replaced by constants where we write $p_i(\vec{c}_i)\vec{x}_i$ to indicate that the variables $\vec{x}_i$ in $p_i(\vec{x}_i)$ have been replaced with constants $\vec{c}_i$.

3.2 Semantics

We now summarize the commonly used semantics of temporal logic. Let $S = \{ s_i : i \geq 0 \}$ be a collection of states (sometimes referred to as worlds) and an accessibility relation among states as $R \subseteq S \times S$. We assume that there is a mapping (referred to as an assignment of the fully instantiated instances of the predicate symbols), say $Inst = \{ \text{inst}_k \geq 0 \}$ with the mapping $AtMap = M$.
Inst \mapsto \mathcal{P}(S)$. Then we define the satisfaction relations for the predicates in the states as follows:

- $s_i = \text{inst}_k$ if $s_i \in \text{AtMap}(\text{inst}_k)$.
- $s_i = \text{inst}_k \land \text{inst}_j$ if $s_i = \text{inst}_k$ and $s_j = \text{inst}_j$.
- $s_i = \neg \text{inst}_k$ if $s_i \not\in \text{AtMap}(\text{inst}_k)$.
- $s_i = \text{inst}_k \lor \text{inst}_j$ if $s_i = \text{inst}_k$ or $s_j = \text{inst}_j$.
- $s_i = \forall x \text{pk}$ if $s_i = \text{inst}_k$ for every instance $\text{inst}_k$ of $\text{pk}$ and the only free variable of $\text{pk}$ is $x$.
- $s_i = \exists x \text{pk}$ if $s_i = \text{inst}_k$ for some instance $\text{inst}_k$ of $\text{pk}$, and the only free variable of $\text{pk}$ is $x$.
- $s_i = \bigcirc \text{inst}_k$ if $s'_i = \text{inst}_k$ for some $s'_i \in R^*(s_i)$, where $R^*$ is the reflexive transitive closure of $R$.
- $s_i = \blacksquare \text{inst}_k$ if $s'_i = \text{inst}_k$ for every $s'_i \in R^*(s_i)$, where $R^*$ is as stated above.
- $s_i = \Diamond \text{inst}_k$ if $s'_i = \text{inst}_k$ for some $s'_i \in R^*(s_i)$.

The sample version described in the paper uses only seven states $S = \{s_1, s_2, \ldots, s_7\}$. We also use 66 predicates, i.e. $P = \{t_1, t_2, \ldots, t_{66}\}$, seven constants. Our model checker uses $X$ for $\bigcirc$. Sample safety requirements shown in this paper currently do not use the connective $\bigcirc$. We show verification related to the registration, donor suitability and the main workflow.

3.3 Mapping Safety Requirements as Assertions in States and State Transitions

We describe the state transitions and safety properties that must be satisfied by state transitions as per FDA and AABB specifications. Given the complexity of the process, we modeled the workflow as consisting of sub-workflows and sub-sub-workflows. These result in having hierarchical states and safety assertions associated with transitions between them, which we write as temporal logic formulas and verify using a model checker. For the purpose of this paper, we described sample state transitions, how we decomposed FDA and AABB requirements into LTL assertions about hierarchical states, and how we verified them. Table 1 provides a summary of our assertions.

As shown in Figure 3, state begin at $S_1$ and safety properties are shown in Table 1.

- $S_1$: The user starts the workflow by entering start into the system. On trigger $t_1$ (the user entering start) a transition $n_1$ takes the system from $S_1$ to $S_2$ as shown in Figure 3.
- $S_2$: The user gets donor’s demographics [DonorName, DonorAge, NationalSA, GovID, Address, Occupation, Homephone, Mobile, Email, MartialStatus, Gender, Volunteer Autologous, FirstTimeDonation, DonationTypeRegular]. Based on the entered demographics the system routes the user between USregistration or SRegistration. (In this paper we will cover only SRegistration and $S_{\text{Asuitability}}$ that corresponds to the registration process in a Saudi hospital, modeled by states $S_3$ and $S_4$). On trigger of $t_2$ (it passes the nationality such as Canada, US, New Zealand, Saudi Arabia, China), the state in transition $n_2$ from $S_2$ to $S_3$. Conversely, if other nationality state transitions from $S_2$ to $S_8$ (shown in transition $n_{30}$).
- $S_3$: In this composite state, the system checks the donor demographic information. It decomposes the sub-workflow (into $S_{3.1}$-$S_{3.4}$) and outputs a Boolean flag $\text{VerifiedID}$. It is modeled as transition $n_{32}$ that on trigger $t_2$ (user identify and verify the donor ID) transition from $S_3$ to $S_4$. Conversely, if the donor ID is not verified, transition $n_{32}$ takes $S_3$ to $S_6$, as shown in Figure 3.
- $S_4$: In this composite state, the system checks if the donor is suitable for the donation and output a Boolean flag $\text{DonorSuitable}$. This state decomposes the tasks into sub-workflows (going from states $S_{4.1}$ to $S_{4.8}$) and outputs a Boolean flag $\text{DonorSuitable}$ indicating the donor passing the physical exam. It is modeled as transition $n_{33}$ that on trigger $t_3$ (user identify the donor suitability) transition from state $S_4$ to $S_5$.

Now we briefly describe the decomposition of a few complex states. Compound state $State$ 3 in Figure 3 is decomposed into eight sub-states ($S_{3.1}$ to $S_{3.8}$) but we describe only three states in this paper. The second row in Table 1 shows the safety property of validating the ID of a new donor should hold in state $S_{3.1}$. In the beginning state of sub-workflow the system imports the donor demographics [GovID] from the main workflow. When the donor is successfully verified, transition $n_4$ takes the system from $S_{3.1}$ to $S_{3.2}$, as shown in Figure 3.

We show the decomposition of compound state $State$ 4 in Figure 4 where the safety property associated with the third row of Table 1 holds in state $S_{4.1}$, Because $S_{4.1}$ is the beginning state of sub-workflow where the system imports the donor demographics [GovID, DonorAge, Autologous and FirstTimeDonation] and starts as state $S_4$. The decomposition takes the system through 14 states, numbered $S_{4.4.1}$ through $S_{4.4.14}$ of which we describe 5. It is modeled as
transition $n_{10}$ that is triggered by action start and goes from state $S_{4.1}$ to $S_{4.2}$ in Figure 4. We only describe two states of this sub-sub-workflow and associated safety properties for brevity.

We show the decomposition of compound state state 4.4 in Figure 5 with the fourth property holding at $S_{4.4.4}$ explained in Table 1:

- $S_{4.4.1}$: The start state of the sub-workflow, where the user clicks the start button to begin the sub-workflow. This transition $n_{17}$ takes the system from state to $S_{4.4.1}$ to state $S_{4.4.2}$ triggered by the action start.
- $S_{4.4.2}$: In this state, the system imports [FirstTimeDonation] and checks if the donor had previous donations, modeled as the transition $n_{18}$ that takes the system from state $S_{4.4.3}$ to state $S_{4.4.4}$ triggered by the action $t_{57}$.
- $S_{4.4.3}$: The user enters [DonationIntervalDays and DonorSuitable] to route the donor. It is modeled as transition $n_{19}$ that takes the system from state $S_{4.4.3}$ to state $S_{4.4.4}$ triggered by actions $t_{63}$.
- $S_{4.4.4}$: The user enters the donor [Temp, Syspressure, Diapressure, Armclear, Hemoglobin], as the blood bank technician performs the physical exam. This is modeled as transition $n_{20}$ that takes the system from state $S_{4.4.4}$ to state $S_{4.4.5}$ automatically.
- $S_{4.4.5}$: The system checks if the entered temperature is in the normal range. In order to do so, the system imports [Temp] into the process and ask the user to enter [Donorsuitable]. This is modeled by the transition $n_{21}$ that takes the system from state $S_{4.4.4}$ to state $S_{4.4.5}$ triggered by action $t_{66}$.

4. Implementation

This section describes our implementation as shown in Figure 6, which extends previous work [13], [14] by adding a workflow specification verification component. The extended system includes three components. (1) Electronic Medical System, (2) Workflow Management System (WFMS), and (3) Workflow Specification Verification System. For (1) we adopt an open-source EMR system, OpenMRS, as our EMR System component. We create the blood bank user interfaces to be used by blood banks. All patient data is stored in OpenMRS databases. For (2) we have Workflow Editor and Workflow Engine. Workflow editor models the comprehensive, completed blood bank workflow that we created for collecting and administering blood products. Workflow Engine enforces such workflow, which takes blood bank professionals through safety verified processes when collecting and administering blood products. We also use an open-source WFMS—namely, YAWL—as our WFMS, which is implemented as a loadable module in OpenMRS[15]. For (3), which is highlighted in yellow in Figure 6, the two main components are Workflow Specification Translator and a Model Checker. The Workflow Specification Translator creates Divine (DVE) syntax of the workflow specifications. The Model Checker verifies the model for safety requirements specified as LTL properties. As stated previously, we encode FDA and AABB blood safety requirements to ensure that our workflow model is compliant. Finally, WFMS databases store the verified blood bank workflow specification for collecting and administering blood products.
We describe the verification of our model against safe blood requirements, highlighted in yellow in Figure 6.

5. Related Work

Systems such as SCC Soft Computer, MAK-system and many others [16], [17] do not formally model and verify blood safety requirements. Instead, they validate safety by using FDA specified validation guidelines as shown in the FDA 510(k) Blood Establishment Computer Software [18].

Ruan et al. [19], [20] specify an agent-based alarm system as properties in logic, verified against formally modeled palliative care therapeutics. They created their agent-based alarms using a subset of detailed palliative workflows, checked against norms set by palliative care providers using first order LTL based modeled checker [19]. Our work captures much more than a subset of blood workflow; it details the vein-to-vein processes. Also, we check the properties by extracting and translating from governmental regulations from the FDA and others such as AABB.

Kristensen et al. [21] utilize a workflow management tool and LTL to verify some specified properties using so-called sweep-lines in model driven architectural design. They focused more toward the state space explosion, but also discussed the use of checking if all properties hold in specific states. Our work differs as we focus toward ensuring blood bank regulations are checked in specific states to ensure all properties hold to meet the safety requirements utilizing the same model. Also, enforcing the same model verified into an EMR system. Devine et al. set forth approaches to verify clinical guideline properties that are linked to a model checker. Divine[22]. Their work focuses on checking the model and properties for consistency by using LTL and Promela, a model checker for clinical guidelines in Ischemic stroke prevention and management. Our work, by contrast, focuses on blood bank safety by enforcing compliance with governmental regulations and standards throughout the whole blood bank workflow.

Our work also differs from what was listed in related work by adding automated verification to ensure that practiced blood supply chains satisfy regulatory mandates. In addition, we show how EMR users can use these verified, safe workflows to provide seamless blood-related services transparent to the caregiver.

<table>
<thead>
<tr>
<th>States, Property and Predicates Summary.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State</strong></td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S3.1</td>
</tr>
<tr>
<td>S4.1</td>
</tr>
<tr>
<td>S4.4.1</td>
</tr>
</tbody>
</table>
6. Conclusions

We have proposed and partly built what we believe to be the first system that uses generic workflow management to drive the blood collection and administration system. We have also used blood bank workflows to create a method to verify that the workflow steps satisfy the safety requirements mandated by governing authorities such as the FDA and the AABB. Our prototype shows that our methodology is sufficiently generic to model FDA mandates and AABB recommendations.

Blood safety is dynamic: Continual changes in mandates require updating the FDA regulations and AABB standards. Our system accepts these changes because the methodology is generic and, thus, can be used to specify changing safety standards and newer workflows. For example, hemovigilance started in 1994 as a means to further increase blood safety [23]. Hemovigilance attempts to track donation and transfusion processes to decrease the number of unwanted occurrences or events. We are in the process of building verification systems for Hemovigilance.

7. Acknowledgment

This work is sponsored in part by the Saudi Arabian Cultural Mission (SACM) and King Abdulaziz University (KAU). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, by the Government of the Kingdom of Saudi Arabia.

References