Process Programming to Support Medical Safety: A Case Study on Blood Transfusion

Lori A. Clarke[1], Yao Chen[1], George S. Avrunin[1], Bin Chen[1], Rachel Cobleigh[1],
Kim Frederick[1], Elizabeth A. Henneman[2], and Leon J. Osterweil[1]
[1]Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
{clarke, yaoc, avrunin, chenbin, rcobleig, kfrederi, ljo}@cs.umass.edu
[2]School of Nursing
University of Massachusetts
Amherst, MA 01003 USA
henneman@nursing.umass.edu

SPW2005 Beijing, China
May 25 2005

Introduction

- Medical errors
  - Result in approximately 98,000 deaths per year in the United States
  - Caused by faulty processes and conditions, (Institute of Medicine)
- IOM advocates using more information technology to help improve medical care.

Medical Safety Project

- Medical Safety Project at UMASS-Amherst
  - Researchers in Dept. of Computer Science have been working with researchers and medical practitioners from UMASS School of Nursing & from Baystate Medical Center
- Investigating applying software engineering technologies and evaluating effectiveness

Our Approach

- Process programming to model medical processes
  - Little-JIL process programming language
- Requirements engineering to capture medical safety properties as formal statements
  - Propel property elucidation system
- Finite-state verification to detect errors
  - LTSA (Labelled Transition System Analyser)
  - SPIN (Simple PROMELA Interpreter)
  - FLAVERS (Flow Analysis for Verification Systems)
- Case Study – In-Patient Blood Transfusion Process

Outline

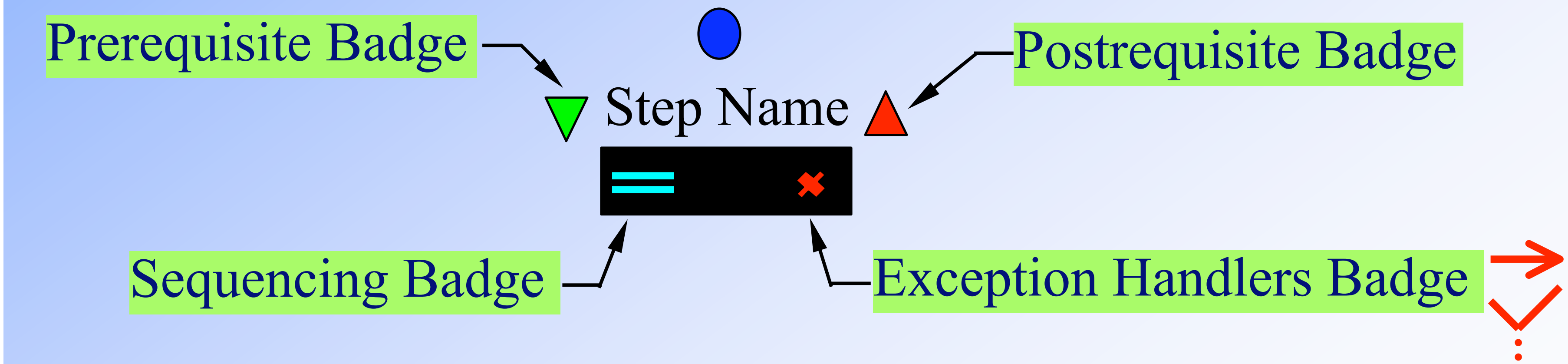
- Defining Processes
- Representing Properties
- Analyzing Processes
- Observations
- Conclusions and Future Work

Defining Processes

- Medical processes
  - Complex, concurrent, and exception-rich
- Process language requirement
  - Capture complexity in medical processes
  - Precise enough to support static analysis and to eventually drive simulations and executions
  - Understandable to a medical professional

Little-JIL Overview

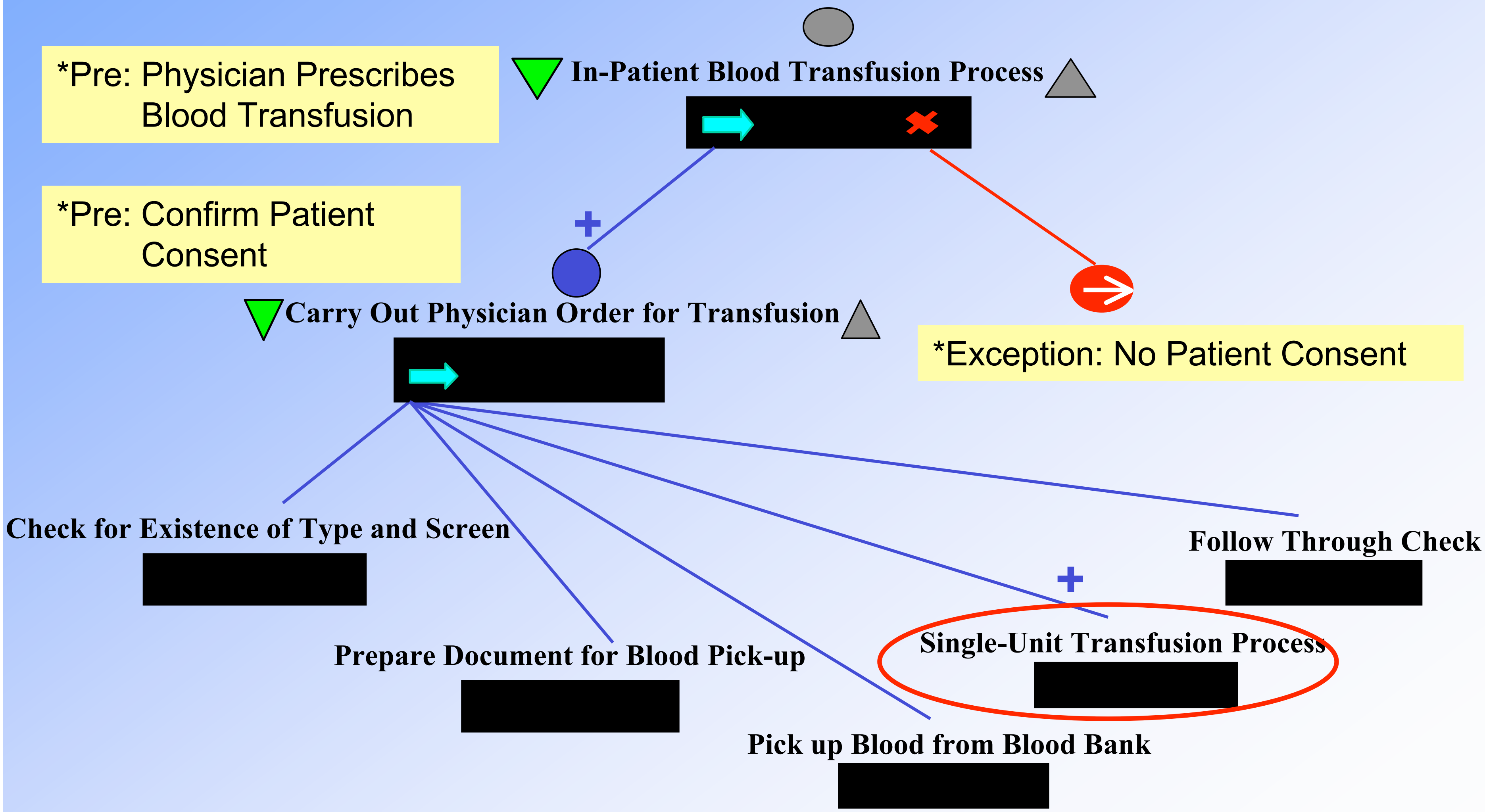
- Visual language for coordinating tasks
- Uses hierarchically decomposed steps
- Step icon



In-Patient Blood Transfusion Process Example

- Consists of 23 Little-JIL Diagrams
  - Decompose in-patient blood transfusion process into conceptually meaningful subprocesses
- Present a few of the Little-JIL in-patient blood transfusion process diagrams to give an indication of what the model looks like

In-Patient Blood Transfusion



*Pre: Physician Prescribes Blood Transfusion

*Pre: Confirm Patient Consent

In-Patient Blood Transfusion Process

Carry Out Physician Order for Transfusion

*Exception: No Patient Consent

Check for Existence of Type and Screen

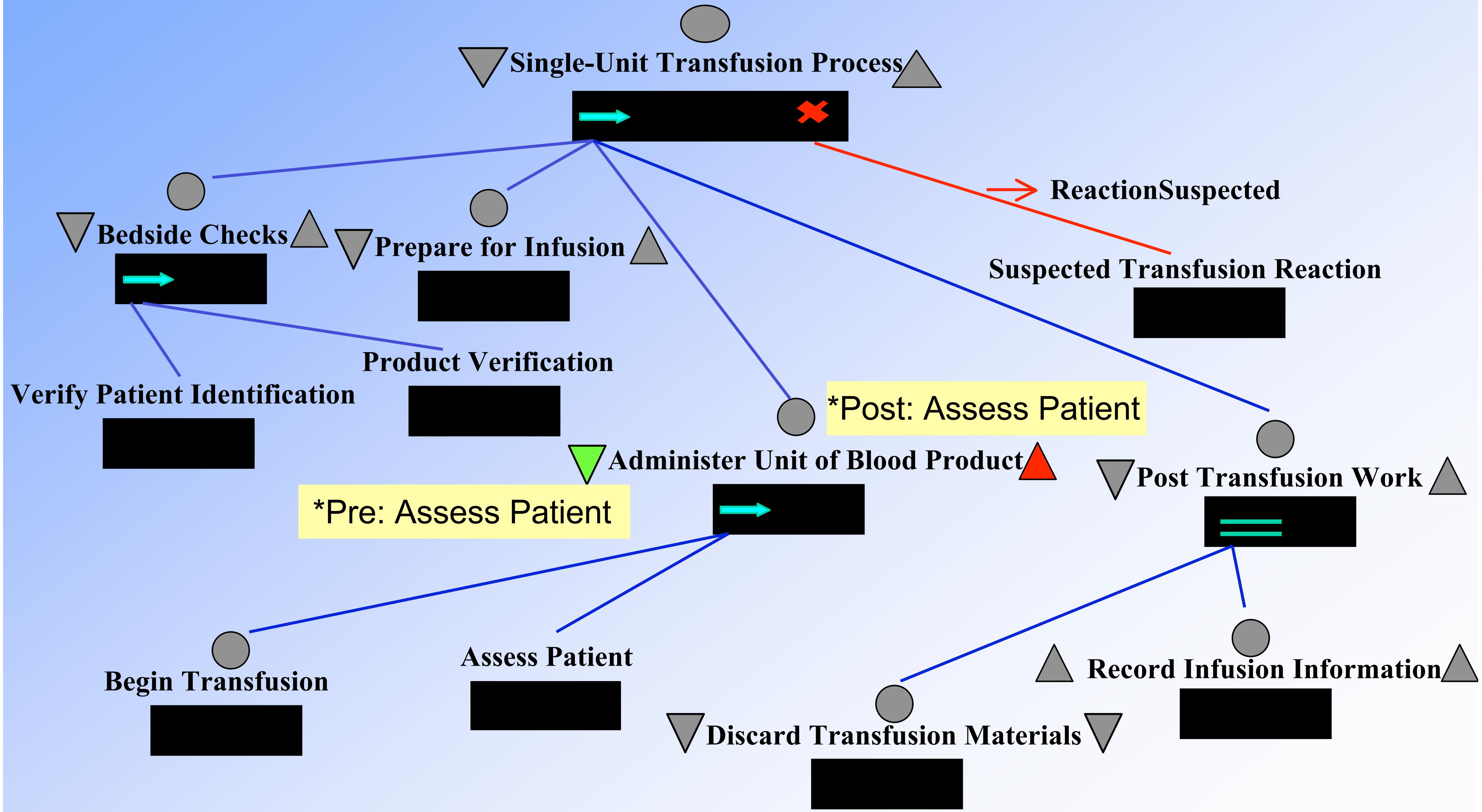
Prepare Document for Blood Pick-up

Pick up Blood from Blood Bank

Single-Unit Transfusion Process

Follow Through Check

Single-Unit Transfusion Process


Single-Unit Transfusion Process
ReactionSuspected
Bedside Checks
Prepare for Infusion
Suspected Transfusion Reaction
Product Verification
Verify Patient Identification
*Post: Assess Patient
Administer Unit of Blood Product
Post Transfusion Work
*Pre: Assess Patient
Assess Patient
Begin Transfusion
Record Infusion Information
Discard Transfusion Materials

In-Patient Blood Transfusion properties

- Policies often exist that are a starting point for these properties, e.g.,
  - The patient’s informed consent must be confirmed prior to carrying out a physician’s order for a blood transfusion.
  - The patient’s identification must be verified immediately before obtaining each blood specimen.
  - The patient’s identification must be verified prior to administering each unit of blood product.
  - ……

Representing Properties

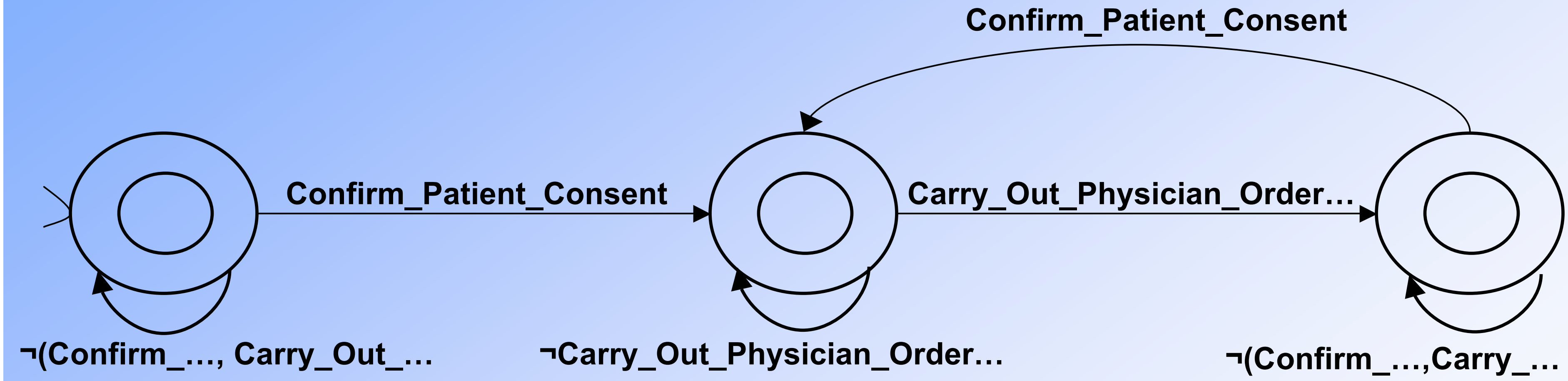
- Propel System
  - Aims to make the job of writing and understanding properties easier
  - Provides three alternative formats
    - Interactive question tree
    - Disciplined natural language
    - Graphical finite-state automata
  - e.g., Patient informed consent must be confirmed prior to each blood transfusion process being initiated.

Propel Question Tree

- How many events of primary interest are there?
  - One: event A
  - Two: events Confirm_Patient_Consent and Carry_Out_Physician_Order_for_Transfusion
    - How do Confirm_Patient_Consent and Carry_Out_Physician_Order_for_Transfusion interact?
      - Confirm_Patient_Consent causes Carry_Out_Physician_Order_for_Transfusion to occur
      - Carry_Out_Physician_Order_for_Transfusion cannot occur until after Confirm_Patient_Consent has occurred
        - Is Confirm_Patient_Consent required to occur at least once?
          - Yes, Confirm_Patient_Consent is required to occur at least once
          - No, Confirm_Patient_Consent is not required to occur at least once
            - After Confirm_Patient_Consent occurs, can …..?

              ……

Finite-state Automaton



Disciplined Natural Language

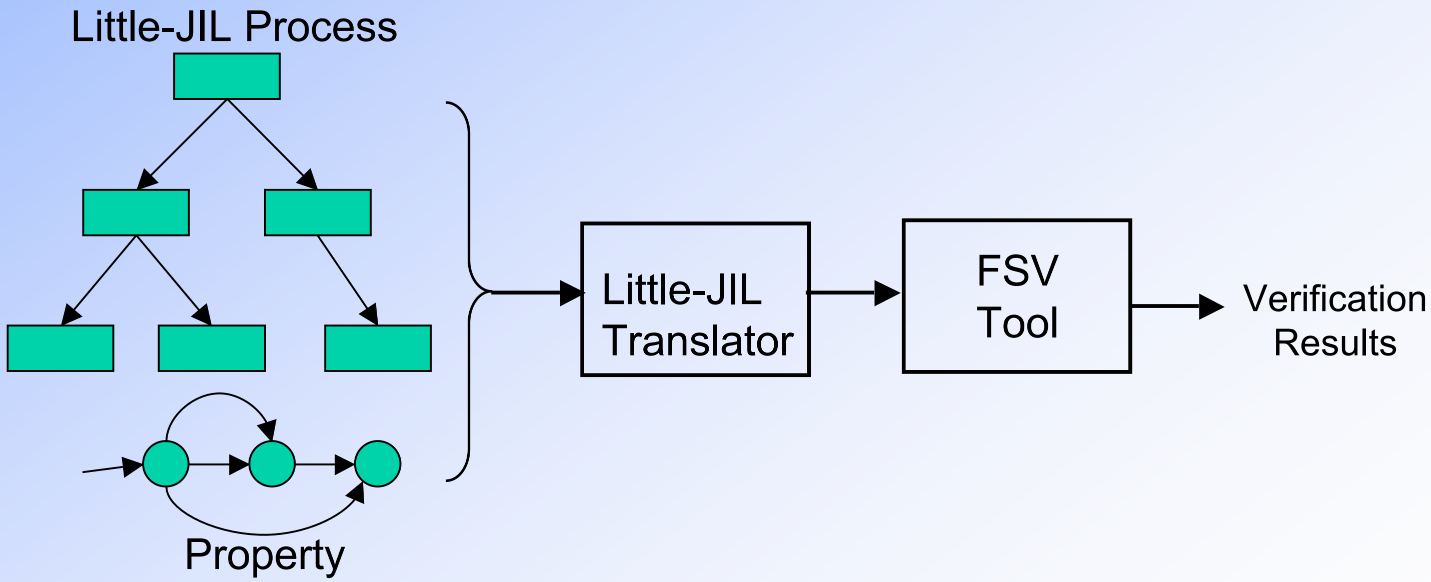
- *Carry_Out_Physician_Order_for_Transfusion cannot occur unless Confirm_Patient_Consent has already occurred.*
- *Confirm_Patient_Consent is not required to occur, however, and if it does not occur, Carry_Out_Physician_Order_for_Transfusion can never occur. Even if Confirm_Patient_Consent does occur, Carry_Out_Physician_Order_for_Transfusion is not required to occur.*
- *Before the first Confirm_Patient_Consent occurs (or the scope interval ends), the events in the alphabet of this property, other than Carry_Out_Physician_Order_for_Transfusion, can occur any number of times.*
- *After Confirm_Patient_Consent occurs and before the first subsequent Carry_Out_Physician_Order_for_Transfusion occurs (or the scope interval ends), the events in the alphabet of this property, including Confirm_Patient_Consent but not Carry_Out_Physician_Order_for_Transfusion, can occur any number of times.*
- *After the first subsequent Carry_Out_Physician_Order_for_Transfusion occurs:*
  - *the events in the alphabet of this property, other than Confirm_Patient_Consent or Carry_Out_Physician_Order_for_Transfusion, could occur any number of times;*
  - *Carry_Out_Physician_Order_for_Transfusion cannot occur again until after another Confirm_Patient_Consent occurs;*
  - *Confirm_Patient_Consent can occur and if it does, then the situation should be regarded as exactly the same as when the first Confirm_Patient_Consent occurred, meaning that all restrictions described on the events would again apply.*

Analyzing Processes

- Apply finite-state verification techniques to determine if the process definition is consistent with each property
  - Considers every trace through the process
- If a property does not hold, the verification tool will provide a counterexample trace
- Process improvement: change process, property, or both
  - Reverify until satisfied

Process Verification

- Little-JIL process translated to intermediate representation
- Intermediate representation translated to the expected input for the selected Finite-State Verification (FSV) tool



Observations

- Took several iterations to represent the in-patient blood transfusion process
  - Difficult to find right level of granularity
  - Formulating properties helped improve the process definition
- There is a tension between expressiveness and analyzability
  - Some of the more expressive constructs in Little-JIL are difficult to model. e.g., choice step
- Medical professionals could understand the process definition

Observations(Cont’d)

- The verifiers revealed errors in the process
  - All verifiers used found the same errors in the process
- All verifiers have some limitations
  - Need optimization and abstraction to reduce the size of the model generated
  - FLAVERS is currently best able to handle the larger problems, but requires more insight to tune the model

Conclusions and Future Work

- Appears to be a promising approach but more work is needed
  - Support for timing in the process language, property specifications, and analysis tools
  - Support for simulation and simulation-based analysis
    - Reduce patient waiting time
    - Optimize use of resources, e.g., number of beds
  - Support for execution in the clinical setting
    - Process-guided automation
    - “Hands free” interface
  - More evaluation

Thanks