Smart Checklists for Human-Intensive Medical Systems

George S. Avrunin[1] Lori A. Clarke[1] Leon J. Osterweil[1]
Julian M. Goldman[2] Tracy Rausch[3]

[1]University of Massachusetts Amherst

[2]Massachusetts General Hospital

[3]DocBox, Inc.

WORCS 2012

Human-Intensive Systems

Human-intensive systems (HISs): Systems involving people, devices, and software applications in which the participation and expertise of the humans play a central role in achieving success

- E.g., medical care, air traffic control, nuclear plant management

Human-Intensive Systems

Human-intensive systems (HISs): Systems involving people, devices, and software applications in which the participation and expertise of the humans play a central role in achieving success

- E.g., medical care, air traffic control, nuclear plant management

HISs are:

- Typically concurrent and exception-rich; correct actions of participants are heavily dependent on context (current state and history)
- Hard to understand, develop, and maintain since they add the complexity and variability of human participation to complex cyber-physical systems

Human-Intensive Systems

Human-intensive systems (HISs): Systems involving people, devices, and software applications in which the participation and expertise of the humans play a central role in achieving success

- E.g., medical care, air traffic control, nuclear plant management

HISs are:

- Typically concurrent and exception-rich; correct actions of participants are heavily dependent on context (current state and history)
- Hard to understand, develop, and maintain since they add the complexity and variability of human participation to complex cyber-physical systems
- *Error-prone*—100,000 avoidable deaths per year in US hospitals from medical errors

Checklists

- Use of checklists to support human participants is well-established in domains like aviation
- More recently introduced in medicine with some positive results, but also notable shortcomings:
  - typically simple, largely sequential, static lists
  - don't handle exceptions or reflect complex dynamic context
  - seen as adding to workload

We are exploring the use of *smart, context-aware, dynamic* checklists to assist human participants in medical processes

- Building on our previous work on formalizing and analyzing medical processes

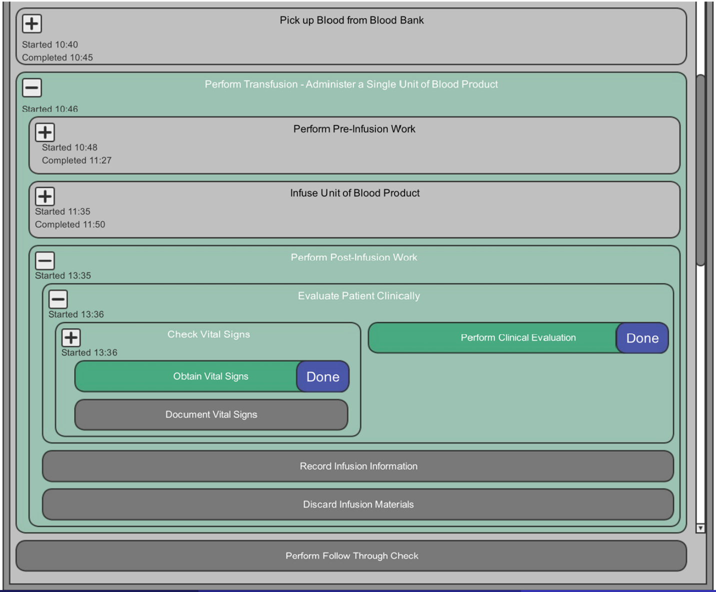
Examples

- OR-ICU handoff of patient undergoing coronary artery bypass graft surgery
  - When surgery is completed, patient moved to ICU
  - ICU personnel must prepare appropriate equipment (infusion pumps, blood pressure monitors, lung ventilator, etc.), supplies, and medication
  - During surgery, information about medications, ventilator settings, any atypical devices/therapies transmitted to Smart Checklists for ICU personnel; devices in ICU autoconfigured (with clinician confirmation using Smart Checklists)

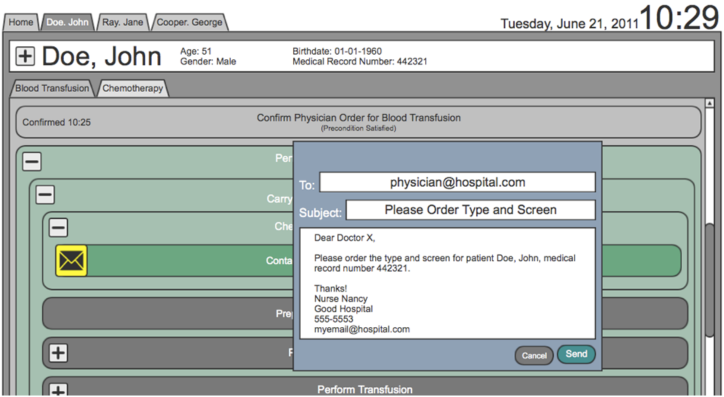
Examples

- OR-ICU handoff of patient undergoing coronary artery bypass graft surgery
  - When surgery is completed, patient moved to ICU
  - ICU personnel must prepare appropriate equipment (infusion pumps, blood pressure monitors, lung ventilator, etc.), supplies, and medication
  - During surgery, information about medications, ventilator settings, any atypical devices/therapies transmitted to Smart Checklists for ICU personnel; devices in ICU autoconfigured (with clinician confirmation using Smart Checklists)
- Interruption of ventilator for x-ray
  - When ventilator turned off for x-ray, Smart Checklists remind appropriate personnel to turn it back on (with increasing urgency!)

What a Checklist Might Look Like



Another Example



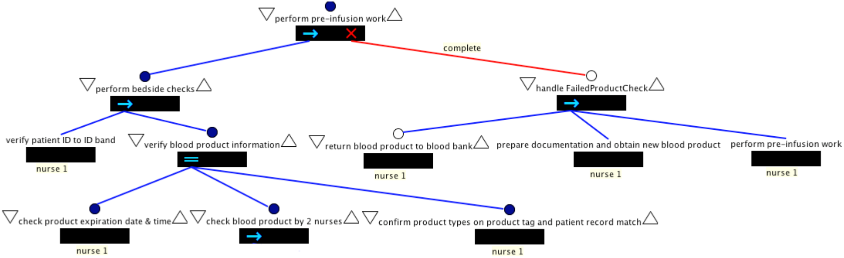
Our Previous Work Focused On

- Modeling Processes
  - Little-JIL process language
    - rich language with well-defined semantics; includes concurrency, exception-handling, etc.
    - describes agents, resources, artifacts

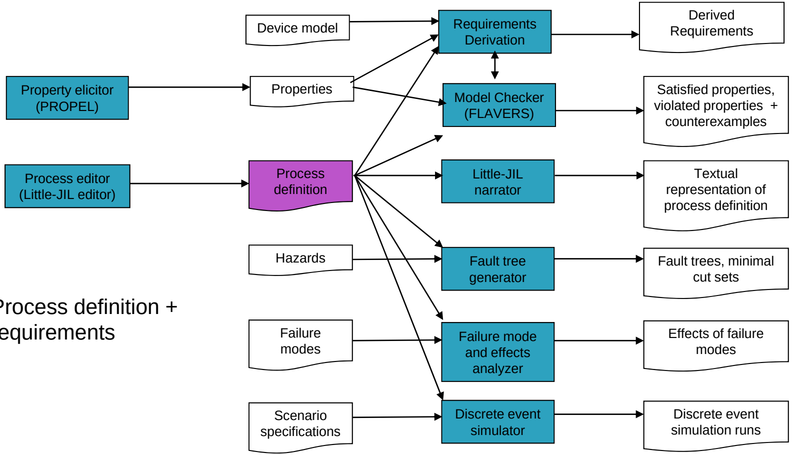
Our Previous Work Focused On

- Modeling Processes
  - Little-JIL process language
    - rich language with well-defined semantics; includes concurrency, exception-handling, etc.
    - describes agents, resources, artifacts
- Analyzing process models
  - Error detection
    - Model checking
  - Vulnerability analysis
    - Fault-tree Analysis and Failure Modes and Effects Analysis
  - Evaluation of efficiency
    - Discrete event simulation

A Little-JIL Fragment



Process Improvement Environment



Process definition + requirements

Static Analysis

Process Improvement Feedback

Case Studies

- Blood transfusion
- Chemotherapy administration
- Patient flow in emergency department

Case Studies

- Blood transfusion
- Chemotherapy administration
- Patient flow in emergency department

Medical professionals say we changed the way they describe and teach their processes

Case Studies

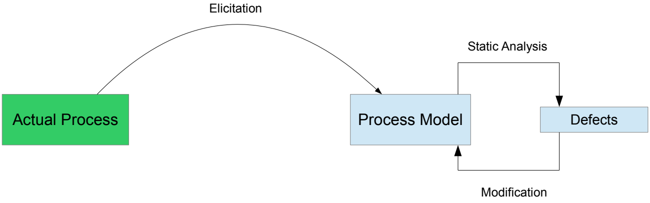
- Blood transfusion
- Chemotherapy administration
- Patient flow in emergency department

Medical professionals say we changed the way they describe and teach their processes

Chemotherapy process saw 70% reduction in errors that reach the patient. [Mertens 2012]

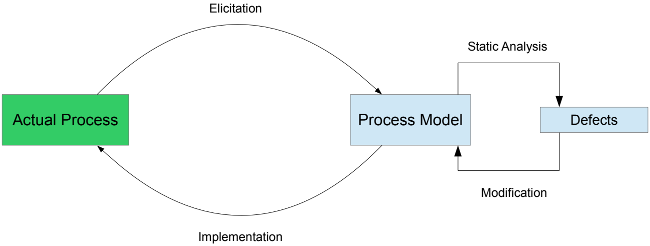
Process Improvement

Case studies used a static process improvement cycle:



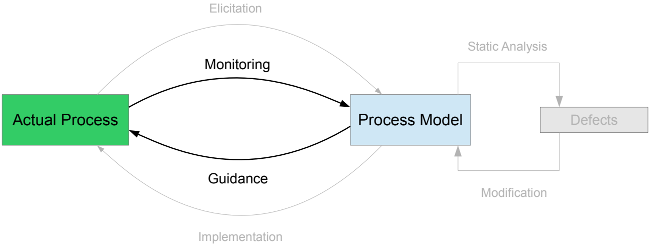
Process Improvement

Case studies used a static process improvement cycle:



Process Improvement

Case studies used a static process improvement cycle:



Now we want to use the model to monitor and guide *ongoing* process execution

Research to Achieve This Vision

We have recently begun to develop a prototype system, *SmartCheck*, to explore these ideas in the health care domain.

Research directions we are pursuing now:

- Architecture to support communication between “real world” and “model world” and among checklist components
- Monitoring
- Retrospection and prospection of process state
- Deviation detection and explanation
- Real-time and profile-based analyses

Architecture

Building on

- DocBox technology
  - Developed with Medical Device Plug-and-Play Interoperability Program
  - Creates links between human performers, devices, and hospital network
- Janus message passing system (UMass)
  - Translates between agent activities and Little-JIL process events
  - Has been used primarily with human agents, but will communicate with devices and hospital software applications through DocBox platform

Most analysis components are both producers and consumers of information; have to manage their communication.

Monitoring

- Requires event recording mechanism
  - For now, we assume hospital electronic medical record system or some other system records events we need
- Challenges
  - Events may be dropped, misinterpreted, recorded out of order
  - Repetition may be harmless (check ID, record temperature, check ID) or harmful (administer medication, check ID, administer medication)

Process Context

Human process performers need information about history of process execution, artifacts, etc.

- Especially important in circumstances, like OR-ICU handoff, where some performers enter process with little knowledge of history of particular process execution
- Little-JIL maintains some information but will need new technologies to gather, store, and summarize such information as what steps were performed by which entities, using which inputs
- Initially using Data Derivation Graphs [Osterweil 2008, 2010; Lerner 2011] to manage the information; controlling size will be important

Process Context

Human process performers need information about history of process execution, artifacts, etc.

- Especially important in circumstances, like OR-ICU handoff, where some performers enter process with little knowledge of history of particular process execution
- Little-JIL maintains some information but will need new technologies to gather, store, and summarize such information as what steps were performed by which entities, using which inputs
- Initially using Data Derivation Graphs [Osterweil 2008, 2010; Lerner 2011] to manage the information; controlling size will be important

Also want to provide prospective information about future execution

- What steps are coming up? What are likely resource utilization consequences of alternative?
- Need estimates based on analyses of historical profiles, simulations, etc.

Detect Deviations from Process

Notify participants when process execution deviates from process model

- Map recorded events to process steps (not necessarily 1-1 correspondence)
- Determine whether recorded sequence of events corresponds to a (prefix of a) trace in process model

Detect Deviations from Process

Notify participants when process execution deviates from process model

- Map recorded events to process steps (not necessarily 1-1 correspondence)
- Determine whether recorded sequence of events corresponds to a (prefix of a) trace in process model

May not be able to do this immediately . . .

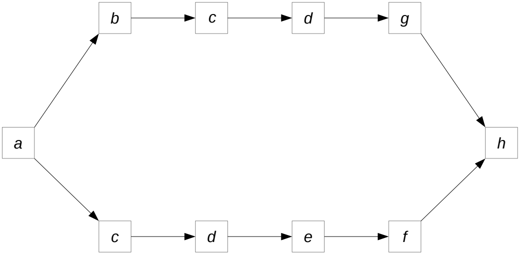
Detect Deviations from Process

Notify participants when process execution deviates from process model

- Map recorded events to process steps (not necessarily 1-1 correspondence)
- Determine whether recorded sequence of events corresponds to a (prefix of a) trace in process model

May not be able to do this immediately . . .

recorded sequence:
acdg



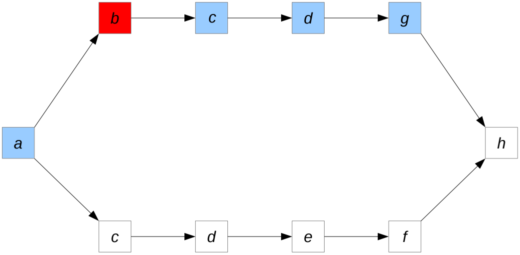
Detect Deviations from Process

Notify participants when process execution deviates from process model

- Map recorded events to process steps (not necessarily 1-1 correspondence)
- Determine whether recorded sequence of events corresponds to a (prefix of a) trace in process model

May not be able to do this immediately . . .

recorded sequence:
acdg



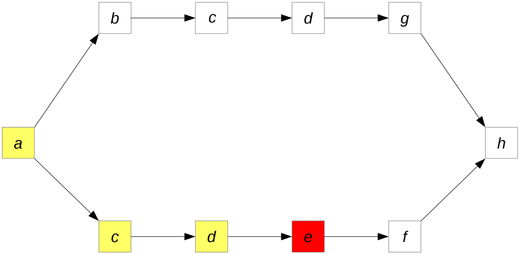
Detect Deviations from Process

Notify participants when process execution deviates from process model

- Map recorded events to process steps (not necessarily 1-1 correspondence)
- Determine whether recorded sequence of events corresponds to a (prefix of a) trace in process model

May not be able to do this immediately . . .

recorded sequence:
acdg



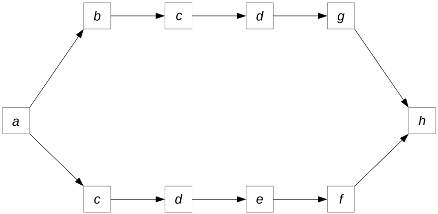
"Explain" Deviations from Process

When deviation is detected, want to suggest likely sources of problem to human participants

"Explain" Deviations from Process

When deviation is detected, want to suggest likely sources of problem to human participants

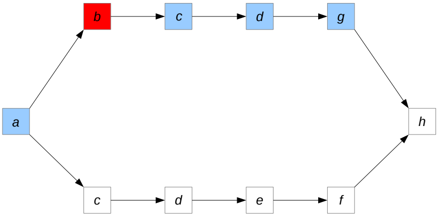
recorded sequence:
acdg



"Explain" Deviations from Process

When deviation is detected, want to suggest likely sources of problem to human participants

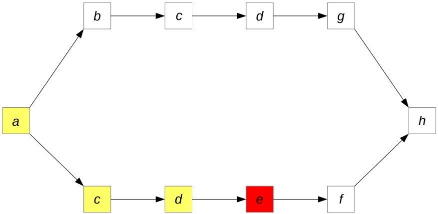
recorded sequence:
acdg



"Explain" Deviations from Process

When deviation is detected, want to suggest likely sources of problem to human participants

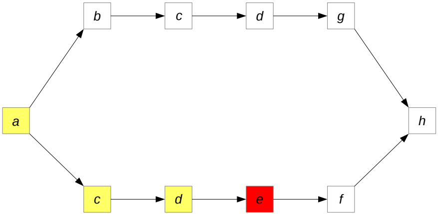
recorded sequence:
acdg



"Explain" Deviations from Process

When deviation is detected, want to suggest likely sources of problem to human participants

recorded sequence:
acdg



Problem is either a dropped *b* or doing *g* instead of *e*. But correct recovery actions are likely to be different.

Experimenting with use of string matching techniques to measure “edit distance” between recorded sequence and traces and provide participants with a useful ranking of likely explanations.

Real-Time and Profile-Based Analyses

- Real-time
  - Static analysis to identify timing vulnerabilities and guide decisions about meeting deadlines
  - Dynamic analysis to identify upcoming deadlines, issue (appropriately intrusive) warnings
  - Little-JIL currently provides a primitive timer construct, but not powerful enough to express kinds of hard and soft real-time constraints arising in medical care, so research on specification will be needed

Real-Time and Profile-Based Analyses

- Real-time
  - Static analysis to identify timing vulnerabilities and guide decisions about meeting deadlines
  - Dynamic analysis to identify upcoming deadlines, issue (appropriately intrusive) warnings
  - Little-JIL currently provides a primitive timer construct, but not powerful enough to express kinds of hard and soft real-time constraints arising in medical care, so research on specification will be needed
- Profile-based analysis
  - Accumulate historical information summarizing multiple process executions
    - Use this to, e.g., determine whether a particular check is wasteful or suggest inserting an additional check to catch a large number of errors
    - Incorporate good estimates of probabilities of events/transitions for probabilistic model checking, FTA, FMEA, and string matching for explaining deviations

Evaluation

- Does architecture of our prototype adequately support communication and interaction among components and process agents?
- How well does our system represent past, present, and future context information? How well does it respond to queries about context from process performers?
- What kind of detail/fidelity in the event stream is required?
- How well does our system detect deviations and identify likely causes?
- How can profiles of past executions be used to improve process models, monitoring, and guidance?
- How well can we specify and supply real-time information?

Will explore these using simulated event streams from a variety of sources and panels of experts; eventually move to simulated clinical settings

Thanks