USING CONSTRAINT LOGIC PROGRAMMING TO SUPPORT THE CONCURRENT APPLICATION OF MULTIPLE PRACTICE GUIDELINES: ADJUSTING MEDICATION DOSAGES AND RESOLVING REPEATED ACTIONS RELATED TO TREATMENT

Martin Michalowski², Szymon Wilk¹,³, Wojtek Michalowski¹, Di Lin⁴, Ken Farion⁵, Subhra Mohapatra

MET Research Group, University of Ottawa
in collaboration with
1. Telfer School of Management
2. Adventium Labs
3. Poznan University of Technology
4. McGill University
5. Children’s Hospital of Eastern Ontario
Outline

• Clinical practice guidelines (CPGs)
• Issues and a clinical case study
• Overview of mitigation approach
• Extensions to support complex relationships and repeated actions
• Discussion and future work
Clinical Practice Guideline (CPG)

- Clinical algorithm
  - Evidence-based best practice in healthcare
  - Guides decisions and criteria regarding diagnosis, management, and treatment of a single condition
  - Used to treat patients
  - Might be represented in multiple different (and not always compatible) formats,
    - Forms a narrative, in a flowchart or decision table
- A valid therapy always exists for a single CPG
Motivation

- 50% of people 65 years old or older have a comorbid condition [Institute of Medicine, 2001]

However...

- Most attention has been paid to an individual CPG instead of adapting the guidelines to manage comorbid condition
  - Creating formal/executable CPG representations
  - Translating unstructured text into formal CPG models
  - Verifying CPG models
  - Using CPG models to assist MD take actions

- Research on combining CPGs is still in its infancy and development of combined CPG is mostly expert-based
## Research Questions

1. How to represent multiple CPGs associated with comorbid conditions as a single executable model?  
2. How to process this model (verify, revise) to ensure a therapy for comorbid conditions exists?  
3. How to support complex interactions in these models?  
   - Dosing of medications  
   - Repeated Actions  

*Foundations*  
*New Extensions*
Clinical Case Study

- Concurrent application of CPGs for a patient who is being treated for Wolff Parkinsons White Syndrome (WPW) and suffers an Atrial Fibrillation (AF)
  - Common comorbid condition managed in the ED
  - Overlapping and possibly contradicting treatments
  - Dosages of medication need to be adjusted depending on other measurements
  - Repeated actions that manifest themselves as loops in the CPG
    - Number of iterations is not explicit
Mitigation Approach Overview

- Two CPGs applied to patient with comorbid conditions to obtain combined *therapy*
- Combined therapy does not exist in case of adverse interactions between individual therapies
  - Direct adverse interactions caused by contradictory actions (e.g., to give medication A, not to give medication A)
  - Indirect adverse interactions caused by drug-drug or drug-disease interactions (e.g., giving medication A is forbidden when some disease is present)
- Mitigation (identification and addressing) of adverse interactions requires clinical acumen (experts, textbooks, clinical evidence)
- Clinical acumen encoded in form of operators
  - *Interaction operators* to model adverse interactions
  - *Revision operators* to model revisions
Mitigation Approach Overview (cont.)

Phase 0: pre-processing

- Transform CPGs into actionable graphs
  - Actionable graphs
  - Create logical models from actionable graphs
    - Logical models of CPG₁ and CPG₂

Phase 1: mitigating direct adverse interactions

- Check for direct adverse interactions
  - Adverse interactions exist?
    - Yes
      - Address direct adverse interactions
        - Adverse interactions addressed?
          - Yes
          - No
      - No
- Check for indirect adverse interactions
  - Adverse interactions exist?
    - Yes
      - Address indirect adverse interactions
        - Adverse interactions addressed?
          - Yes
          - No
    - No

Success

Failure
Key Concepts: Actionable Graphs and Paths

- Actionable graph \((AG_i)\) for \(CPG_i\) is defined as a directed graph
  \[ AG_i = <N_i, A_i> \]

- \(N_i\) = set of context, action and decision nodes
  - Context node provides clinical context, \(AG_i\) has a context node as its root, indicating the disease handled by \(CPG_i\).
  - Action node corresponds to an action step from \(CPG_i\)
  - Decision node corresponds to a decision step from \(CPG_i\)

- \(A_i\) = set of arcs representing transitions between nodes
  - Inspired by SDA* \((State-Decision-Action)\) formalism for health [Isern et al., 2009]
Key Concepts: Actionable Graphs and Paths

Actionable graph for WPW

1. Patient diagnosed with WPW
   - Dosage of F (DF0=50mg/day)
   - WPW stable (WSk)
     - DF=DFk
     - Patient release (PR)
   - DFk<DFmax
     - Adjust dosage of F (DFk=DF0+kΔDF)
     - No
       - WPW stable (WSk)
         - DF=DFk
         - Patient release (PR)
     - Yes
       - Another treatment (AT)

Actionable graph for AF

1. Patient diagnosed with AF
   - Hemodynamic instability (HI)
     - Yes
       - Structured heart disease (HD)
         - No
           - Electrical cardioversion (EC)
             - Yes
               - Amiodarone IV (AIV)
                 - No
                   - Flecaïnide IV (FIV)
                   - Recurring AF episode (RAE)
                     - Yes
                       - Oral amiodarone (A)
                         - No
                           - Patient release (PR)
                             - No
2. No
   - Yes
     - Another treatment (AT)
Key Concepts: Logical Models

- A *logical model* ($LM_i$) provides a logical representation of an $AG_i$
  
  $$LM_i = <d_i, V_i, PLE_i>$$

- $d_i$ = label of disease associated with $AG_i$
- $V_i$ = set of action and decision variables associated with actions and decision nodes in $AG_i$
- $PLE_i$ is a set of logical expressions representing paths in $AG_i$
Key Concepts: Logical Models

Logical Model created from \( AG_{AF} \)

\[
d_{AF} = AF \\
V_{AF} = \{HI, EC, HD, AIV, FIV, RAE, A, PR\} \\
PLE_{AF} = \{(HI = y) \land EC \land (RAE = y) \land A \land PR \land \neg FIV \land \neg AIV, \neg A, \neg EC, \neg AIV, \neg A, \neg EC, \neg AIV, \neg A \}
\]

Example variables and domains

- HI (hemodynamic instability) = \{yes, no\}
- HD (structured heard disease) = \{yes, no\}
- RAE (recurring AF episode) = \{yes, no\}
**Key Concepts: Combined Logical Models**

- A *combined logical model* \((CLM_{i,j})\) brings together a pair of logical models and information about adverse interactions between underlying CPGs
  \[
  CLM_{i,j} = <LM_i, LM_j, ILE_{i,j}>
  \]

- \(LM_i \& LM_j\) = individual logical models representing \(AG_i\) and \(AG_j\)

- \(ILE_{i,j}\) = logical expressions that represent indirect adverse interactions between \(CPG_i\) and \(CPG_j\)
  \[
  ILE_{WPW, AF} = \neg (A \land DF = DF_{max})
  \]
Original Assumptions and Extensions

- Only Boolean variables
  - Requires discretizing decisions and “go/no go” actions
  - **Extension:** introduce numeric variables
    - Action variables support finer grained details
      - Previously *Flecainide* := True / False
      - Now *Flecainide* := [0…500]
    - Decision variables no longer need discretization
      - Previously *if Flecainide* == 150
      - Now *if Flecanide* > 150 ∧ *Flecanide* < 300

- Acyclic AG
  - A node can only be traversed once
    - Unable to support repeated actions (re-testing or monitoring)
  - **Extension:** allow for algorithmic expressions and conditions
    - Previously ¬( *A* ∧ *DF*)
    - Now ¬( *A* ∧ *DF* = *DF*_{max}) ∧ ( *DF* = *DF*_1 + *DF*_2 + *DF*_3 + …)
Extensions: Supporting repeated actions

• Find and expand loops in AGs

procedure expand(in AG_i, in WorstCase_i, out AG_i_exp)
begin
  1. Loop_i := identify_loop(AG_i)
  2. MaxIter_i := check_conditions(Loop_i, WorstCase_i)
  3. ForwardPath_i := create_path(Loop_i, MaxIter_i)
  4. AG_i_exp := replace_loop(AG_i, ForwardPath_i)
  5. return AG_i_exp
end

• Loop_i found using a path-based strong component algorithm (Tarjan’s)
  • Assumes a single loop in the AG
• WorstCase_i is defined according to patient information and secondary knowledge (expert's opinion, evidence, literature, …)
• Supports diseases where therapy involves repeated actions
Extensions: Supporting repeated actions

Original AG_{WPW}

Automatically Revised AG_{WPW}
Updated Mitigation Approach

- Transform CPGs into actionable graphs
  - Actionable graphs
  - Expand loops in actionable graphs
  - Create logical models from actionable graphs
    - Logical models of $CPG_1$ and $CPG_2$

**Phase 0:**
- pre-processing

Modified step to handle increased complexity associated with numerical variables

New step introduced to handle loops in actionable graphs

Modified step to handle increased complexity associated with numerical variables
Discussion and Future Work

• **Contribution:** Support complex relationships
• **Contribution:** Identify and expand repeated actions
• **Benefit:** Steps towards a comprehensive alerting system for physicians at the point of care
• **Benefit:** Steps towards wider acceptance on CPGs in clinical practice [Sittig et al. 2008]
• **Future Work:** Refine inference in handling repeating actions
• **Future Work:** Modeling temporal aspects of CPGs
Thank you

martin.michalowski@adventiumlabs.com