Automated Conflict Detection Between Medical Care Pathways

Philip Weber, Bosco Filho, Mark Lee, Ian Litchfield, Ruth Backman

University of Birmingham, UK

School of Computer Science | Institute of Applied Health Research

ICSSP, Paris, 7th July 2017
What is the Problem?

In England > 15 million people have a long-term health condition.

Around 70% of the money spent on health and social care.

In the UK 2.9 million with three or more (Multi-morbidity) – by 2018.

- aging, smoking, diet, inactivity, ...
- cancer, heart disease, lung disease, diabetes, depression, ...

Complex processes for treatment (people, factors, clinical evidence ...)

UK National Institute for Care Excellence (NICE):
- Clinical Guidelines → Care Pathways.
A care pathway is essentially a process for treatment of a disease.
COPD overview

Person over 16 at risk of COPD

- Diagnosis and assessment
- Multidisciplinary team
- See what NICE says on patient experience

- Management
- Managing exacerbations

- Palliative care

© NICE 2017
Two Problems

1. **Informal modelling** – potential for inconsistency,
2. Focus on **single conditions** – potential for conflict.

**Implicit cycle** of retesting.

- What does Metformin conflict with?
- What does HbA1c interact with?

Our work:

1. define a **formal pathway model** to capture clinical pathways,
2. develop automated methods for **conflict detection**,
3. recommend minimal solutions for **conflict resolution**.
1. Modelling clinical guidelines
Modelling Clinical Guidelines

Requirement: formal modelling for analysis.
Many options (YAWL, Petri Nets, Computer Interpretable Guidelines, …)

Business Process Model and Notation (BPMN):

- well-known de facto business process modelling language,
- increasingly prevalent for modelling clinical pathways,
- graphical, intuitive, flexible, ‘subset-able’.

But

- no formal semantics,
- models can be unstructured,
- especially the semantics w.r.t. data are unspecified.
Requirement: formal modelling for analysis.
Many options (YAWL, Petri Nets, Computer Interpretable Guidelines, ...)

Business Process Model and Notation (BPMN):

- well-known de facto business process modelling language,
- increasingly prevalent for modelling clinical pathways,
- graphical, intuitive, flexible, ‘subset-able’.

But

- no formal semantics,
- models can be unstructured,
- especially the semantics w.r.t. data are unspecified.
Data in Care Pathways

Routing may be dependent on and modify data, e.g.

- Is patient already taking medication $m$?
- If test $X > v$, refer for treatment $X$ else retest in $M$ months.
- If patient age $A > a$ prescribe drug $x$ else $y$.
- Record the fact of prescription of drug $Z$.

BPMN has the Data Object element, but

- semantics open to interpretation,
- decoupled from the control-flow.

Literature covering formalisation of

- BPMN integration with data objects, e.g. [Meyer et al., 2013];
- interaction between processes and databases [Sun et al., 2014];
- seems more complex than we need;
- similarly the data semantics of YAWL or Computer Interpretable Guidelines.
Routing may be dependent on and modify data, e.g.

- Is patient already taking medication $m$?
- If test $X > v$, refer for treatment $X$ else retest in $M$ months.
- If patient age $A > a$ prescribe drug $x$ else $y$.
- Record the fact of prescription of drug $Z$.

BPMN has the Data Object element, but

- semantics open to interpretation,
- decoupled from the control-flow.

Literature covering formalisation of

- BPMN integration with data objects, e.g. [Meyer et al., 2013];
- interaction between processes and databases [Sun et al., 2014];
- seems more complex than we need;
- similarly the data semantics of YAWL or Computer Interpretable Guidelines.
Routing may be dependent on and modify data, e.g.

- Is patient already taking medication $m$?
- If test $X > v$, refer for treatment $X$ else retest in $M$ months.
- If patient age $A > a$ prescribe drug $x$ else $y$.
- Record the fact of prescription of drug $Z$.

BPMN has the Data Object element, but

- semantics open to interpretation,
- decoupled from the control-flow.

Literature covering formalisation of

- BPMN integration with data objects, e.g. [Meyer et al., 2013];
- interaction between processes and databases [Sun et al., 2014];
- seems more complex than we need;
- similarly the data semantics of YAWL or Computer Interpretable Guidelines.
Fragments of doctors’ appointments for review of

**OA** : Osteoarthritis.

**COPD** : Chronic Obstructive Pulmonary Disease (Lung Disease).
Fragments of doctors’ appointments for review of

**OA** : Osteoarthritis.

**COPD** : Chronic Obstructive Pulmonary Disease (Lung Disease).
BPMN+V: Data-Enhanced BPMN

Formal execution semantics . . .

- subset of BPMN notation – can be expanded,
- formal semantics of execution,
- based on Workflow Graphs ([Vanhatalo et al., 2007] formalism),

. . . and integration with data,

- semantics of dependence on – and modification of – data,
- based on Coloured Petri Nets.
Workflow Graphs

Effectively a subset of BPMN allowing the main control-flow patterns and imposing some structure on the model [Vanhatalo et al., 2007].

- \( G = (N, E) \), nodes \( N \), edges \( E \), such that \( E \subseteq N \times N \),
- \( N \in \{ \text{Start}, \text{Stop}, \text{Activity}, \text{Fork}, \text{Join}, \text{Decision}, \text{Merge} \} \),
- \( G \) is well-formed by definition,
- allowing for atomic activities, parallel and alternative behaviours.

Semantics of \( G \) is a token game (cf Petri Nets).

- State \( s \) of \( G \) is a mapping \( s : E \rightarrow \mathbb{N} \) assigning tokens to edges in \( E \).
- \( s(e) = k \iff \) in state \( s \), edge \( e \) carries \( k \in \mathbb{N} \) tokens.
- execution of node \( n \) changes the state to \( s' : s \xrightarrow{n} s' \).

This says nothing about data.
BPMN+V: Modelling Data

Flexible approach to model data:

- Fixed set of $d$ variables $X = \{x_1, \ldots, x_d\}$ of types $T(x_i) \in \{T_1, \ldots, T_m\}$,
- valuations $V = (\nu_1, \ldots, \nu_d)$ assigned to $X$ as the process executes.
  - $V$ assigned to token (‘colour’).
- Activity may be guarded by pre- and post-conditions $c(\cdot)$:
  - $c(\cdot)$ is a first-order logic formula over $X$,
  - $c(\cdot) \models V$ if the valuation $V$ satisfies $c(\cdot)$,
  - e.g. $\text{pre}(a) := c(x_1, \ldots, x_d) \triangleq (x_i > 55)$.
- Activity may carry out data modifications:
  - statement $f(\cdot) : V \rightarrow V'$ over variables in $X$,
  - e.g. $x := x + 1$ or $x := \text{False}$,
- data may require synchronisation – managing the control-flow.

Implicit data is referenced but not modified.
- e.g. database of drug-drug-disease interactions (Stockley / BNF).
BPMN+V: Execution Semantics

Extension of Workflow Graphs:

- \( G = (N, I, E, X, pre, post, mod) \),
- \( I : N \rightarrow \{ \text{Start}, \text{End}, \text{Activity}, \text{Exclusive}, \text{Inclusive}, \text{Parallel} \} \),
- \( X = \{ X_1, \ldots, X_d \} \),
- \( \{ pre, post \} : N \rightarrow C \),
- \( mod : N \rightarrow D \) (database),
- allowing for atomic activities, parallel, exclusive or inclusive choice.

Semantics defined in terms of before- and after- conditions and states,

- \( m : E \rightarrow \{ T_1, T_2, \ldots \} \) is a marking describing the state,
- mapping each edge \( e \in E \) to coloured tokens, \( T_i = (t_i, V_i) \),
- execution modifies the state \( m \xrightarrow{n} m' \) and (perhaps) valuation \( V \rightarrow V' \).
e.g. For an Activity $a$ in a well-formed BPMN+V model:

- one input and one output sequence flow $e_{in}$, $e_{out}$.
- $a$ consumes $T = (t, V)$ from $e_{in}$ and returns $T' = (t, V')$ on $e_{out}$,
- if $\exists T = (t, V) \in m(e_{in}) \mid pre(a) \models V$, \hspace{1cm} // $V$ satisfies any pre-condition.
- then $m \xrightarrow{a} m'$, where 

\[
\begin{align*}
1. & \quad post(a) \models V', \text{ and} \\
2. & \quad m'(e) = \\
& \begin{cases} 
    m(e) \setminus \{T\} & \text{if } e = e_{in}, \hspace{1cm} // V' \text{ satisfies any post \textemdash condition.} \\
    m(e) \cup \{T'\} & \text{if } e = e_{out}, \hspace{1cm} // \text{token is 'moved'.} \\
    m(e) & \text{otherwise.}
\end{cases}
\end{align*}
\]

- Similarly for all node types.

\[
\begin{align*}
\text{e.g.} \quad pre(a) & \triangleq \neg \text{NSAIDS}, \\
f(a) & \triangleq \text{corticosteroids} := \text{corticosteroids} + 1.
\end{align*}
\]
**Review core treatments**

- **Data attributes**
  - guard: not breathless
  - data: NSAIDS+1

- **Data annotations**
  - execution dependency on data

- **Prescribe topical capsaicin**
- **Prescribe paracetamol for pain relief**
- **Refer for intra-articular injections**
- **Consider addition of opioid analgesics. Consider risks and benefits, particularly in older people**
- **No further pain relief required**

- **Prescribe topical NSAIDs for pain relief**
- **Prescribe NSAIDs**

- **Prescribe paracetamol for pain relief**
- **Prescribe theophylline**
- **Prescribe Roflumilast**
- **Prescribe Mucolytics**

- **Review theophylline usage and requirement**
- **Review corticosteroid usage and requirement**

- **Has the patient reported any breathlessness?**
- **Why are Corticosteroids needed?**

- **To manage an exacerbation**
- **Prescribe and keep dose as low as possible**
- **Prescribe prophylactics**

- **Arrange plasma level monitoring**

- **End of medication review**

---

**Managing a patient with stable COPD**

- **Has the patient reported any breathlessness?**
- **Why are Corticosteroids needed?**

- **To manage an exacerbation**
- **Prescribe and keep dose as low as possible**
- **Prescribe prophylactics**

- **Arrange plasma level monitoring**

- **End of medication review**
2. Conflict Detection
The problem: to identify conflicts between clinical care guidelines followed concurrently in treating patients with multiple morbidities.

Assume

- two BPMN+V models (care pathways) $M_1, M_2$,
- interacting with database $D$,
- shared set of $d$ variables $X = X_1 \cup X_2$,
- set of $k$ constraints $C = \{C_1, \ldots, C_k\}$.

Constraint $C_r$ is a logical formula over $X$, e.g.

- if $x_i$ and $x_j$ indicate prescription of two medications,
- which must not be taken together,
- then $C_r(x_1, \ldots, x_d) \triangleq \lnot(x_i \land x_j)$.

The problem:

*identify all pairs of execution paths through $M_1, M_2$ which will modify the variables in $X$ so that at least one of the $C$ is violated.*
The problem: to identify conflicts between clinical care guidelines followed concurrently in treating patients with multiple morbidities.

Assume

- two BPMN+V models (care pathways) $M_1, M_2$,
- interacting with database $D$,
- shared set of $d$ variables $X = X_1 \cup X_2$,
- set of $k$ constraints $C = \{C_1, \ldots, C_k\}$.

Constraint $C_r$ is a logical formula over $X$, e.g.

- if $x_i$ and $x_j$ indicate prescription of two medications,
- which must not be taken together,
- then $C_r(x_1, \ldots, x_d) \triangleq \neg(x_i \land x_j)$.

The problem:

*identify all pairs of execution paths through $M_1, M_2$ which will modify the variables in $X$ so that at least one of the $C$ is violated.*
State Space Approach to Conflict Detection

Current process for conflict detection (evaluate BPMN+V):

1. **Model** clinical pathways in BPMN+V,
2. **simple** parallel composition,

**Assumes** patient (potentially) starts both pathways concurrently. **Assumes** no common activities.

**Future:** intelligent model composition.
Current process for conflict detection (evaluate BPMN+V):

1. Model clinical pathways in BPMN+V,
2. simple parallel composition,
3. annotate with constraints (potential conflicts, e.g. meds. dependencies),
4. identify data combinations for which to explore the model,

- Identify $d$ variables $X$ involved in conditions,
- values $V$ checked/assigned.

- Create a ‘covering set’ of $2^d$ initial data settings for validation:
  - e.g. $\{y > 1, y \leq 1\}$ for a condition $y > 1$. 
State Space Approach to Conflict Detection

Current process for conflict detection (evaluate BPMN+V):

1. Model clinical pathways in BPMN+V,
2. simple parallel composition,
3. annotate with constraints (potential conflicts, e.g. meds. dependencies),
4. identify data combinations for which to explore the model,
5. state space exploration for each data combination (via CPN),

- Transform to CPN (take advantage of existing methods).
- Construct $2^d$ reachability graphs $R_i$ (explore state space).
- Conflicting Activities indicated by non-final dead markings linked by common variables.
- Repeat for individual models (detect data-related inconsistencies)
  - and composed models (detect conflicts).
State Space Approach to Conflict Detection

Current process for conflict detection (evaluate BPMN+V):

1. Model clinical pathways in BPMN+V,
2. simple parallel composition,
3. annotate with constraints (potential conflicts, e.g. meds. dependencies),
4. identify data combinations for which to explore the model,
5. state space exploration for each data combination (via CPN),
6. identify ‘non-final dead markings’,
7. visualise and interpret the conflicting activities and data combinations.
Composed OA and COPD Model

Explore all effective values

Potential inconsistency

Simple parallel composition

(Possibly invalid assumption that the patient starts following both models at the same time.)
3. Evaluation
3-stage evaluation:

1. **Artificial** process fragments, e.g.
   - \( \text{guard: not NSAIDS, not corticosteroids} \)
   - \( \text{data:NSAIDS=true} \)

2. **Randomly**-generated models,
   - block-structured expansion,
   - controlled block probabilities and number of conflicts.

3. **Running example** – Osteoarthritis (OA) and COPD pathways,
   - 14 activities,
   - 3 variables,
   - up to 11,000 states in the composed model.
### Report and Visualise Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Activity</th>
<th>Data</th>
<th>Initial Data</th>
<th>Conflict Model</th>
<th>Conflict Activity</th>
<th>Conflict Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>OA</td>
<td>Agree exercise plan [¬breathless]</td>
<td>breathless=True</td>
<td>breathless=True</td>
<td>NSAIDS=1</td>
<td>NSAIDS=0</td>
<td></td>
</tr>
<tr>
<td>OA</td>
<td>Prescribe NSAIDs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| COPD  | Prescribe cortico. and keep ... | cortico.=1 | cortico.=1 | NSAIDS=0 | NSAIDS=1 | |
| COPD  | Prescribe roflumilast [th.lline < 1] | th.lline=1 | th.lline=0 | NSAIDS=0 | NSAIDS=1 | |
| COPD  | Prescribe th.lline ... [rofl. < 1] | rofl.=1 | rofl.=0 | NSAIDS=0 | NSAIDS=1 | |

**data inconsistencies** for OA (top),
- cannot proceed with exercise plan if patient presents with breathlessness,
- must not over-prescribe NSAIDS.

**COPD** (centre),
- must not over-prescribe corticosteroids,
- check/prescribe Roflumilast and Theophylline are mutually exclusive,
- but parallel structure allows both to be executed.

**conflicts** between the models (bottom).
- corticosteroids and NSAIDs are mutually exclusive across both pathways.
Report and Visualise Results

Model inconsistencies (COPD)

“Inconsistent with this {activity} when {data settings}”.
“Conflict with other {activity} when {data settings}”.

Annotation using http://bpmn.io/ and/or Camunda Modeller.
Bespoke BPMN+V and CPN implementation.
Averages over 30 randomly generated models.

(a) Varying complexity models \([p(seq), p(xor), ppll)]\).

(b) Composed models, varying numbers of conflicts.

(a) \#states vs time (seconds) to run the conflict detection process,
   – models generated with varying probability of sequence, alternate or parallelism.
(b) increasing \#conflicts, in models with low probability of concurrent activity.
Conflict Detection: Performance of State Space Method

Bespoke vs **SNAKES** [1] vs **Neco** [2].
Averages over 30 randomly generated models.

![Graphs showing performance comparisons](image)


“SNAKES is a Python library that provides all the necessary to define and execute many sorts of Petri nets”,


“Neco ... takes a Petri net and builds a library that has all the primitives to explore the state space ... optimised in many ways.”
Future Work

Modelling

- user interface, software tool and case study,
- data integration with sources of data and conflict.

Conflict Detection

- model composition – adequacy of simplistic approach,
- conflict detection using logical specification and constraint solvers,
- scheduling constraints.

Conflict Resolution

- recommendation of minimal changes for conflict resolution,
- e.g. bypass activities,
- e.g. reschedule.
Thank you

Phil Weber

http://www.birmingham.ac.uk/mitcon/

http://www.cs.bham.ac.uk/~weberpy/

p.weber.1@cs.bham.ac.uk