Workflow Management for Health Care Processes Meets Formal Verification

Fazle Rabbi and Wendy MacCaull*

*With valuable input & discussions from Wang, Mashiyat, Leyla, Crawford, Graham, Jewers and numerous clinicians and managers from GASHA

Centre for Logic and Information, StFX University, Antigonish, NS
www.logic.stfx.ca
Motivational Problem: Health Services Delivery

- Health care is approaching a crisis (2009-$192B)
- IT is 20-30 out of date - largely for accounting purposes
- People demanding more and more services
- Work to develop EHRs --- little other IT to support it
- Need for Patient-centred and Collaborative Care
- Guidelines/Standards are common --- how to measure and track care and ensure compliance?
- Health care is variable & distributed process – subject to patient needs, resources of Health Authority
- Safety critical process – money and lives are at stake
Over-riding Goal

- **My Slogan:** *Put science into software*
  - Quality assurance
  - Intelligence

- In Healthcare – the product is the “outcome”; studies show that having and following “good” (evidence-based) process leads to improved outcomes.

- So we focus on the process
PcCfMS

- Collaborative R&D project involving
  - Computer scientists & nurses from academia
  - Clinicians & managers from the local health authority (GASHA)
  - Industry partner: Palomino Innovation Systems Inc.
- Goal: a platform for a web-based workflow management system (PcCfMS) for community-based programs
- Two programs: seniors’ wellness and palliative care (easily adaptable to chronic illnesses)
- Develop & pilot communication, documentation and process management system to promote collaborative patient-centred care.
Workflow management system (WfMS)

- Workflow management system: a computer system to promote the management of work and information in dynamic and distributed organizations

  - Provides consistent output
  - Facilitates visualization of actual work
  - Systematic organization of resources
  - Reduces waste and duplication of work
  - Facilitates documentation
  - Improves efficiency
What is Missing??

- Advanced error handling/compensation for changes
- Quality assurance – guarantee that system is designed to satisfy properties every time
- Monitoring
- Generally lacks flexibility/intelligence
Prototype - NOVA WorkFlow

- compeNsable Ontology-driven Verifiable Adaptive Workflow Management System
- (made in Nova Scotia!!)

- Graphical Editor for Compensable Transactions
- Automated Translator to a Verification Engine
- Workflow as a Service to support various client applications
- (working on integration with ontology to guide workflow; first component almost ready involves access control)
Workflow Languages

- Petri-nets
- Workflow nets
- YAWL
- BPEL
- BPMN
Traditional transaction system

Guarantees that database transactions are processed reliably by ACID properties

ACID is an acronym for
- Atomicity
- Consistency
- Isolation
- Durability
Problem of this transactional system:
Cannot handle long running transactions

What is a long running transaction?
- It has lengthy computation
- It requires long time to complete
- It stops for input from users
- It has sub transactions

Why ACID transactions are not good for long running transactions:
- It requires database and resource locking
- For long running transactions it increases the chance for deadlock
- For web service transactions services are generally independent and autonomous- sometimes they belong to different companies
A compensable transaction is a new type of transaction whose effect can be semantically undone even after it has committed.

It consists of two parts:
1. Forward flow
2. Compensation flow

The behavior structure is described as a state transition diagram:

\[ \Sigma = \{idl, act, suc, abt, fal, und, cmp, hap\} \]
\[ \Delta = \{suc, abt, fal, cmp, hap\} \]
**$t$-Calculus**

- $t$-calculus is the transactional composition language of compensable transactions (Jefing He etc.)
- The syntax of $t$-calculus is made up of several operators, each operator can be semantically defined by a series of behavioral dependencies.

<table>
<thead>
<tr>
<th>Sequential Composition</th>
<th>$S ; T$</th>
<th>Parallel Composition</th>
<th>$S \parallel T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Choice</td>
<td>$S \sqsubset T$</td>
<td>Speculative Choice</td>
<td>$S \otimes T$</td>
</tr>
<tr>
<td>Alternative Forwarding</td>
<td>$S \leadsto T$</td>
<td>Backward Handling</td>
<td>$S \triangleright T$</td>
</tr>
<tr>
<td>Forward Handling</td>
<td>$S \rhd T$</td>
<td>Programmable Compensation</td>
<td>$S \triangleright^* T$</td>
</tr>
</tbody>
</table>
Compensable Task

**Definition**  A compensable task ($\phi_c$) is recursively defined by the following well-formed formula:

$$\phi_c = t_c | (\phi_c \odot \phi_c)$$

where $t_c$ is an atomic compensable task, and $\odot \in \{\; ; \; , \; ||, \; \cap, \; \otimes, \; \leadsto, \; \triangleright, \; \triangleright, \; \ast \}$ is a t-calculus operator defined as follows:

- $\phi_{c_1} ; \phi_{c_2}$: $\phi_{c_2}$ will be activated after the successful completion of $\phi_{c_1}$,
- $\phi_{c_1} || \phi_{c_2}$: $\phi_{c_1}$ and $\phi_{c_2}$ will be executed in parallel. If either of them ($\phi_{c_1}$ or $\phi_{c_2}$) is aborted, the other one will also be aborted,
- $\phi_{c_1} \cap \phi_{c_2}$: either $\phi_{c_1}$ or $\phi_{c_2}$ will be activated depending on some internal choice,
- $\phi_{c_1} \otimes \phi_{c_2}$: $\phi_{c_1}$ and $\phi_{c_2}$ will be executed in parallel. The first task that reaches the goal will be accepted and the other one will be aborted,
- $\phi_{c_1} \leadsto \phi_{c_2}$: $\phi_{c_1}$ will be activated first to achieve the goal, if $\phi_{c_1}$ is aborted, $\phi_{c_2}$ will be executed to achieve the goal,
- $\phi_{c_1} \triangleright \phi_{c_2}$: if $\phi_{c_1}$ fails during execution, $\phi_{c_2}$ will be activated to remove the partial effects remaining in the system. $\phi_{c_2}$ terminates the flow after successfully removing the partial effects,
- $\phi_{c_1} \triangleright \phi_{c_2}$: if $\phi_{c_1}$ fails, $\phi_{c_2}$ will be activated to remove the partial effects. $\phi_{c_2}$ resumes the forward flow to achieve the goal,
- $\phi_{c_1} \ast \phi_{c_2}$: if $\phi_{c_1}$ needs to undo its effect, the compensation flow will be redirected to $\phi_{c_2}$ to remove the effects.
Compensable Workflow Modeling Language (CWML)

Graphical Representation | Formula | Graphical Representation | Formula
--- | --- | --- | ---
Uncompensable task | \( t \) | Compensable task | \( t_c \)
\( \phi_1 \rightarrow \phi_2 \) | \( \phi_1 \land \phi_2 \) | \( \phi_1 \lor \phi_2 \) | \( \phi_1 \times \phi_2 \)

Graphical Representation | Formula
--- | ---
Forward Flow | Backward Flow

\( \phi_c_1 ; \phi_c_2 \) | \( \phi_c_1 \land \phi_c_2 \) | \( \phi_c_1 \lor \phi_c_2 \) | \( \phi_c_1 \times \phi_c_2 \)

\( \phi_c_1 \triangleright \phi_c_2 \) | \( \phi_c_1 \triangleright \phi_c_2 \) | \( \phi_c_1 \triangleright \phi_c_2 \) | \( \phi_c_1 \triangleright \phi_c_2 \)
DiVinE (Distributed Model Checker)

- LTL model checker
- Can effectively handle the well-known "state explosion problem"
- Open source

But:

- Modeling in DVE (or modeling language of any model checker) is tedious and error prone
Translation of And Split-Join block

Petri net diagram of And Split-Join block and its transcription to DVE

```
process Proc_T1{
    tate tr;
    init tr;
    trans
    tr -> tr{ guard var_P1 >= 1;
        effect var_P1 = var_P1 - 1,
                var_T1 = var_T1 +1;
    };
}
```
Automated Verification of Workflow

- NOVA Translator
- Reduced Model (System Requirements)
- Specification (System Property)
- DiVinE (Model Checker)

Answer:
Yes, if the model satisfies the specification
Counterexample, otherwise
\begin{itemize}
  \item Let $F$ be an LTL formula, let $PM$ be a Petri net model and let $DM$ be its translation to DVE; then:
    \begin{align*}
      PN \models F & \quad \text{iff} \quad DM \models F
    \end{align*}
  
  Let $W$ be a workflow model and let $W'$ be its reduction and let $F$ an LTL$_X$ property; then
    \begin{align*}
      W \models F & \quad \text{iff} \quad W' \models F
    \end{align*}
\end{itemize}
Nova Workflow Architecture
Case Study: 2002 National Model for HPC

Overview of the Model to Guide Hospice Palliative Care

Background
- Changing Illness and Bereavement Experiences
- Multiple Domains of Issues
  - Cause Suffering and Quality of Life
  - Disease Management
  - Physical
  - Psychological
  - Social
  - Spiritual
  - Practical
  - End-of-life Care/Death Management
  - Loss, Grief

Definition of Hospice Palliative Care
- Values
- Principles
- Effective Communication
- Effective Group Function
- Change Strategies

Guide to Patient & Family Care
- Square of Care
- Principles & Norms of Practice
- Preferred Practice Guidelines
- Data Collection/Documentation Guidelines

Applications
- Clinicians/administrators
- Quality managers
- Policy makers
- Researchers (clinical)
- Advocates (consumer)

Guide to Organizational Development & Function
- Mission & Vision
- Square of Organization
- Principles & Norms of Practice

Applications
- Administrators/clinicians building organizations
- Careteam leaders
- Quality managers
- Policy makers
- Researchers (systems)
- Advocates (systems)
Palliative Care Workflow

(150 unique tasks, 20 subnet flows, 40 decision points)
Intake
Care Planning
Properties

- **Norms** - Generic principles of care meant to be interpreted locally

- **Prop1 (N3.5, N4.1, N4.4, N5.3)** - If patient is at home and has no family, then there must be a home service provided for his/her care. Otherwise, the patient must move to the hospital.

- **Prop2 (N3.5, N4.1, N4.4)** - If the patient is evaluated and assigned a PPS of 50% or lower then s/he must be moved to the hospital.

- **Prop3 (N3.5, N4.1, N4.4)** - If the patient is evaluated and assigned a level of 3 or lower on `Nursing Documentation' then s/he must be moved to the hospital.

- **Prop4 (N5.1 N1.1, N1.3)** - If the patient's mobility changes and is checked off in `Rounds Report', then a Physiotherapist will be notified.

- **Prop5 (N2.1)** - If the field `Consent to Contact Other Team Member'' on `Issues Log' is set to YES, then Consent to Share Information must be filled out.
### Verification Results

<table>
<thead>
<tr>
<th>Property</th>
<th>Accepting Cycle</th>
<th>States</th>
<th>Memory (MB)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prop1</td>
<td>No</td>
<td>126188210</td>
<td>88619.1</td>
<td>454.4</td>
</tr>
<tr>
<td>Prop2</td>
<td>No</td>
<td>128013744</td>
<td>88920.0</td>
<td>397.9</td>
</tr>
<tr>
<td>Prop3</td>
<td>No</td>
<td>127934841</td>
<td>88894</td>
<td>396.1</td>
</tr>
<tr>
<td>Prop4</td>
<td>No</td>
<td>132038485</td>
<td>90285.3</td>
<td>315.0</td>
</tr>
<tr>
<td>Prop5</td>
<td>No</td>
<td>119611390</td>
<td>85030.1</td>
<td>359.7</td>
</tr>
</tbody>
</table>

All experiments were executed on the Mahone2 cluster of ACEnet, the high performance computing consortium for universities in Atlantic Canada. The tests were performed using DiVinE with 64 CPU's and 3G v memCPU.
## Comparison with other tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>YAWL</th>
<th>ADEPT2</th>
<th>IBM (WebSphere Product Line)</th>
<th>NOVA WorkFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling Language</td>
<td>Unstructured</td>
<td>Structured</td>
<td>Unstructured</td>
<td>Structured</td>
</tr>
<tr>
<td>Workflow Patterns</td>
<td>Yes (most)</td>
<td>Not all</td>
<td>Not all</td>
<td>Not all</td>
</tr>
<tr>
<td>Compensation</td>
<td>No</td>
<td>No</td>
<td>Yes (some)</td>
<td>Yes</td>
</tr>
<tr>
<td>LTL-Verification</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Collaborative Development</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Default Form for Simulation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Dynamic change during execution</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Language Support</td>
<td>XML, XQuery</td>
<td>Java</td>
<td>BPEL, XPDL, Java</td>
<td>Java</td>
</tr>
<tr>
<td>Web Service Integration</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Graphical Editor: Regular Evaluation

Double click to open the workflow

Pre-selection and Post-selection tool; use these to insert a new task

Click on this action to translate the model to DVE

Click on this action button to generate java service classes
Workflow Execution

- NOVA Workflow generates Java code;
- The user provides details of the implementations for the specific task; that is, business logics;
- NOVA Workflow engine executes these tasks in a web server (such as websphere, tomcat, etc.)

Visit CLI website (user manual, examples, etc.)
Client Application that communicates with Nova Workflow Engine
Client Application ctd.
Ongoing Research:

- Incorporate Time into CWML
- Ontology Integration with NOVA Workflow
- Personalized access control framework for workflow based healthcare information.
- Scheduling and Monitoring healthcare processes
- Default Forms for Simulation
- Dynamic Change of Workflow Schema


Acknowledgements

Funding from the following sources is gratefully acknowledged:

- NSERC
- ACOA
- CFI
- ACEnet
- NSRIT
- CIHR