Flexible Behaviour of Human Actors in Distributed Workflows

Adwoa Donyina & Reiko Heckel
(add7@le.ac.uk)        (reiko@mcs.le.ac.uk)
Department of Computer Science
University of Leicester
United Kingdom
Motivation

- Requirement:
  - Test: scheduling protocols, polices and regulation
  - Goal: \( \uparrow \) business productivity

- Possible Solution:
  - Model and simulate business processes to gather data
Problem

- Human behaviour is only predictable to a degree of probability

- It is difficult to accurately model and simulate the dynamic behaviour of humans in business processes, while clearly defining participants, roles and responsibilities.
Key Modelling Language Requirements

1. Dynamic (re)-allocation of roles
2. Temporal Escalation Handling
3. Scheduling and Load Balancing
4. Human Error and Recovery (Backtracking)
Feature Diagram: BP for Flexible Human Actors (BPFHA)

- Access Control
- Dynamic (re)-assignment
- Human Error & Recovery
- Role promotion and demotion
- Scheduling
  - Deadline
  - Priority
- Assignment Policy
- Load Balancing
- Escalation
Basic Methodology Steps

1. **Describe**: Business requirements
   **Formulate**: Performance questions

2. **Model**: Business process

3. **Define**: Tests to answer performance questions

4. **Assign**: Probability distribution to actions in the business process

5. **Perform**: Simulation

6. **Analyze**: Simulation results
Outline

✓ Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach

- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Case Study: Performance Question

Does **escalation** and/or **load balancing**:  
- increase the percentage of prescriptions that are completed within a given deadline,  
  or  
- Reduce the time that prescription cases run past their deadline?
Pharmacy Case Study: Actors, Roles, Responsibilities

Actors (Job Position)

- Registered Pharmacist
- Technician
- Pharmacy Student
- Cashier
- Patient

Roles

- Dispensing Pharmacist
- Entry Technician
- Filling Technician
- Pharmacy Cashier
- Customer
Determine the effectiveness of Escalation Handling & Load Balancing

- Escalation Handling:
  - Level 1 pharmacy cashiers - entry technician
  - Level 2 pharmacy cashiers - filling technicians
  - Level 3 untrained pharmacy students - filling technicians and/or entry technicians

- Load Balancing:
  - The option to transfer prescriptions
Typical:
- Typing Prescription
- Printing Prescription Label
- Filling Prescription
- Checking Filled Prescription
- Receive Payment
- Counsel Customer

Occasional:
- Scenario
Finite State Machine
Outline

- Overview
- Case Study
  - Domain Specific Language (DSL)
  - Application Scenario
  - Rule-based Approach
- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Metamodel

- Linguistic: ontological instance-of relationships

- Elements:
  - Actor (Person)
  - Role (RoleInstance)
  - Process (Case)
  - Escalation
  - Capability
  - ArtifactType (Artifact)
  - AttributeDeclaration (AttributeValue)

- Evolved from analysis of other approaches:
  - Role Based Access Control (RBAC)
  - Organisational Metamodel
M1-O1 Concrete Syntax
(Part 1 of 2)
M1-O1 Concrete Syntax  
(Part 2 of 2)
DSL Syntax (M1-O0)

1) level
2) priority
3) startTime → deadline
4) DateTime
5) roleA:
   CaseA:
6) :RoleInstance
7) :RoleInstance
8) :Person
   capability={c1,c2}
9) :Person
   :RoleInstance
   :Case
10) :Person
    :RoleInstance
    :Case
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
  - Application Scenario
  - Rule-based Approach
- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Initial State: 1 Case

Requires a Pharmacist

Escalation level 1

1 min to deadline

“check” state
Minute Later:
Arrival of new high priority case

- Pharmacist assigned
- Escalation level raised
- Backtrack to “fill” state
- Missing filled prescription
- At “type” state
2 minutes later: priority vs. escalation

Cashier temp capability

Pharmacist assigned to entry technician role

Request filling technician
Minute Later: temp assignment and accomplished action

Temp assignment

Prescription is typed

ready to print
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach
- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Graph Transformation System

Background Information

- **Type Graph**
  - Models the conceptual structure and provides types for the instance graphs
Graph Transformation (GT) Rules

- Composed of pair of instance graphs
  - Left-hand side (L): precondition of the rule
  - Right-hand side (R): postcondition of the rule

- Used for rule-based modification on instance graphs
# Graph Transformation Rules

<table>
<thead>
<tr>
<th>Domain Specific</th>
<th>Managerial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type Prescription</td>
<td>Role request</td>
</tr>
<tr>
<td>Print Label</td>
<td>Role Assignment</td>
</tr>
<tr>
<td>Receive Payment</td>
<td>Role Unassignment</td>
</tr>
<tr>
<td>Fill Prescription</td>
<td>Clock tick</td>
</tr>
<tr>
<td>Check Prescription</td>
<td></td>
</tr>
<tr>
<td>Counsel</td>
<td></td>
</tr>
<tr>
<td>Distribute</td>
<td></td>
</tr>
<tr>
<td>Skip action</td>
<td></td>
</tr>
<tr>
<td>Backtrack action</td>
<td></td>
</tr>
<tr>
<td>Escalation Trigger</td>
<td></td>
</tr>
</tbody>
</table>
Examples of GT Rules applied in scenario

- Escalation Trigger
- Assignment
- Backtrack
- New case
- Request role
- Domain specific action
GT Rule: Escalation Trigger

![Diagram showing current time and deadline conditions for Case A with trigger 2]

**currentTime**

**CaseA**

**currentTime <= deadline**

**trigger2**

**currentTime**

**CaseA**

**deadline**
GT Rule: Assign Pharmacist
GT Rule: Backtrack check state
New Case:
Delivery type and high priority
Request FillingTechnician
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach

- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Stochastic Graph Transformation

Normal Distribution

Exponential Distribution
Graph-based Stochastic Simulation (GraSS)

- An extension of the Viatra Eclipse-based model transformation tool
- Define metamodel, and models in Viatra model space
- Translate GT Rules in Viatra textual syntax
Metamodel in Viatra

- **Artifact**: Artifacts can be modeled in Viatra, including attributes like `actorName`, `tempRole`, `level`, `tempPermitted`, and `actorName`, which are represented as `String` properties.

- **Case**: Cases in the model include attributes such as `deadline`, `priority`, `startTime`, `state`, `completion`, `deadline`, `groupNo`, `priority`, `startTime`, and `state`, which are represented as `String` properties.

- **Clock**: A clock within the model includes attributes `Time` and `DateTime` properties.

- **Escalation**: Escalations are modeled with attributes `actorName`, `tempRole`, `level`, `tempPermitted`, `actorName`, which are represented as `String` properties.

- **Person**: Persons are represented as `Actor` nodes with attributes `actual`, `attr`, `group`, `free`, `groupNo`, which are represented as `String` properties.

- **Process**: Processes are represented as `Node` with attributes `actual`, `attr`, `group`, `free`, `groupNo`.

- **Role**: Roles are represented as `Role` nodes with attributes `assignedTo`, `access`, `presence`.

- **RoleInstance**: Role instances are represented with attributes `assignedTo`, `canAccess`, `presence`.

- **Time**: Time is represented with attributes `Time` and `DateTime`. 

- **M2**: The M2 model includes a `metaModel` and `BPMN2Editor`.
Translate GT rules in DSL into Viatra textual syntax
Example GT rule in Viatra textual syntax (VTCL)

gtrule BacktrackRule_checkState() = {
  precondition pattern lhs(Case_,State_) = {
    Case(Case_);
    AttributeValue(AttributeValue_);
    find RequiresChecked(Case_,AttributeValue_);
    find DPassigned (Case_,RoleInstance_,Role_,Person_);
    neg find FilledPrescriptionExist(Case_,Artifact_,ArtifactType_);
    Case.state(State_);
    Case.attr4(R1,Case_,State_);
    check (value(State_)== "check");
  }
  action {
    setValue(State_,"fill");
    println("error (backtrack to fill state)");
  }
}
Simulation

- 2500 Simulation Steps
- Batch size 3
- Represents 2.77 hours
Start Graph in DSL
Start Graph in Viatra
Simulation Results

Title: Percentage of States Completed

% of Cases from Total Received

<table>
<thead>
<tr>
<th>State</th>
<th>Type</th>
<th>Print</th>
<th>Fill</th>
<th>Check</th>
<th>Payment</th>
<th>Counsel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100</td>
<td>90</td>
<td>70</td>
<td>50</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>80</td>
<td>60</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>70</td>
<td>50</td>
<td>30</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>60</td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>50</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Legend:
- V1: Escalation and Load Balancing
- V2: Escalation
- V3: Load Balancing
- V4: Basic
Simulation Results

Title: Time Comparison

% of Cases from Total Received

- Checked on time
- Counsellled on time
- Checked less than 5 mins late
- Checked more than 5 mins late
- Counsellled late

State Completion Time

- V1: Escalation and Load Balancing
- V2: Escalation
- V3: Load Balancing
- V4: Basic
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach
- Stochastic Graph Transformation Simulation
  - Related Works
  - Current Work
  - Future Work
## Related Work vs. Requirements

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a) BPMN</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>b) WS-Humantask</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>c) MILANO</strong></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>d) FlowMark</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>e) InConcert</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>f) Little-Jil</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>g) ADONIS</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach
- Stochastic Graph Transformation Simulation
- Related Works
  - Current Work
  - Future Work
Current Work
Outline

- Overview
- Case Study
- Domain Specific Language (DSL)
- Application Scenario
- Rule-based Approach
- Stochastic Graph Transformation Simulation
- Related Works
- Current Work
- Future Work
Future Work: Evaluate the method

- Usability:
  - usability testing: ease-of-use

- Expressiveness:
  - Check completeness with respect to requirements

- Scalability:
  - larger models and longer periods of simulation
Thank You