



THE INTELLECTUAL CENTER OF THE JERSEY SHORE
MONMOUTH
UNIVERSITY

An Intuitive Formal Approach to Dynamic Workflow Modeling and Analysis

Jiacun Wang, Daniela Rosca,
William Tepfenhart, Allen Milewski, Michael Stoute

Department of Software Engineering
Monmouth University

Outline

- Motivation
- An intuitive and formal model
- Modeling power
- Well-formed workflows
- Tool support
- Future work

Motivation

- Motivated by incident command system (ICS) modeling
- An ICS allocates people, resources, and services for a major natural or terrorist incident
 - Frequent changes of the course of action dictated by incoming events
 - Predominantly volunteer-based workforce
 - Needs to integrate various software tools and organizations
 - Characterized by a highly distributed workflow management

Challenges

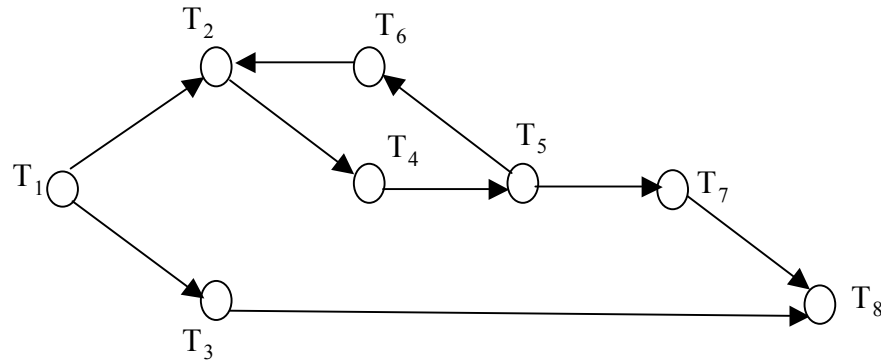
- Ad-hoc changes call for on-the-fly verification of the workflow correctness
- We need a formal approach:
 - No ambiguity
 - Allows analysis
- We need intuitive features for the description and modification of the WF
- We introduced the Workflow Intuitive Formal Approach (WIFA) to meet the needs

WIFA Workflow Definition

A workflow is $WF = (T, P, C, A, S_0)$, where

- $T = \{T_1, T_2, \dots, T_m\}$ is a set of *tasks*, $m \geq 1$.
- $P = (p)_{m \times m}$ is the *precedence matrix* of the task set. If T_i is the direct predecessor of T_j , then $p_{ij} = 1$; otherwise, $p_{ij} = 0$.
- $C = (c)_{m \times m}$ is the *conflict matrix* of the task set. $c_{ij} \in \{0, 1\}$ for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, m$.
- $A = (A(T_1), A(T_2), \dots, A(T_m))$ defines *pre-condition set* for each task. $\forall T_k \in T, A(T_k): {}^*T_k \rightarrow 2^{*T_k}$. Let set $A' \in A(T_k)$. Then $T_i \in A'$ implies $p_{ik} = 1$.
- $S_0 \in \{0, 1, 2, 3\}^m$ is the *initial state* of the workflow.

Example



$$T = \{T_1, T_2, \dots, T_8\},$$

$$A(T_1) = \emptyset, A(T_2) = \{\{T_1\}, \{T_6\}\}, A(T_3) = \{\{T_1\}\},$$

$$A(T_4) = \{\{T_2\}\}, A(T_5) = \{\{T_4\}\},$$

$$A(T_6) = A(T_7) = \{\{T_5\}\}, A(T_8) = \{\{T_3, T_7\}\}.$$

$$S_0 = (1, 0, 0, 0, 0, 0, 0, 0).$$

$$P = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Individual Task State Values

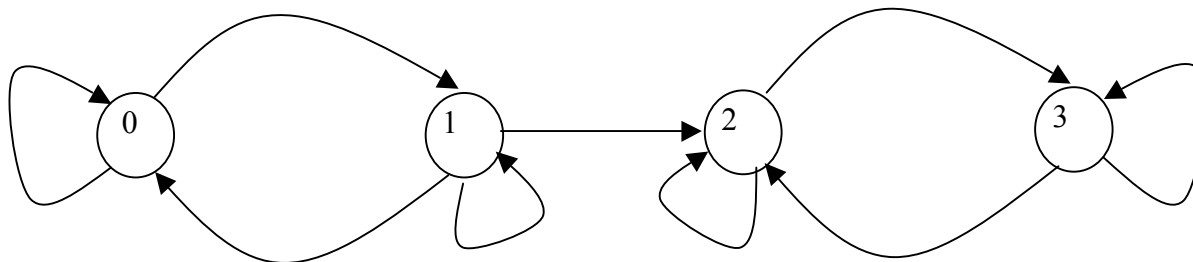
- $S(T_i) = 0$ means T_i is **not executable** at state S and **not executed previously**.
- $S(T_i) = 1$ means T_i is **executable** at state S and **not executed previously**.
- $S(T_i) = 2$ means T_i is **not executable** at state S and **executed previously**.
- $S(T_i) = 3$ means T_i is **executable** at state S and **executed previously**.

State Transition Rules

- A set of rules to guide workflow execution
- Denote by $S_a(T_i)S_b$ that task T_i is executed under state S_a , and the new state after the execution is S_b .
- Rules: $\forall T_j \in T$,
 - If $T_j = T_i$ then $S_b(T_j) = 2$. (T_j is just executed)
 - If $S_a(T_j) = 0$:
 - If $p_{ij} = 1$ and $\exists A' \in A(T_j)$ such that $S_b(T_k) = 2$ for any $T_k \in A'$, then $S_b(T_j) = 1$;
 - otherwise $S_b(T_j) = 0$.

State Transition Rules

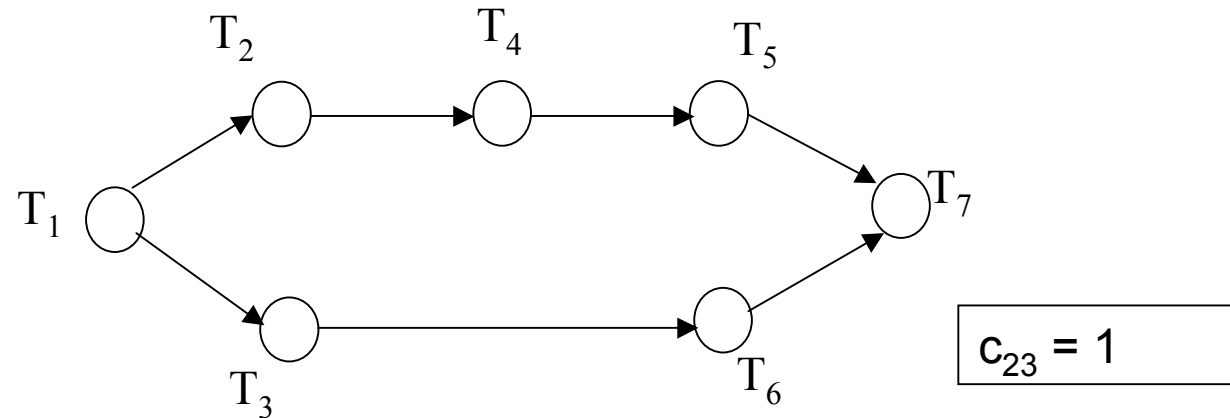
- $S_a(T_j) = 1$
If $c_{ij} = 0$ then $S_b(T_j) = 1$; otherwise $S_b(T_j) = 0$.
- $S_a(T_j) = 2$
If $p_{ij} = 1$ and $\exists A' \in A(T_j)$ such that $S_b(T_k) = 2$ for any $T_k \in A'$,
then $S_b(T_j) = 3$; otherwise $S_b(T_j) = 2$.
- $S_a(T_j) = 3$
If $c_{ij} = 0$ then $S_b(T_j) = 3$; otherwise $S_b(T_j) = 2$.



State value
change of a task

State Transition Rules

- Example



$$S_0 = (1, 0, 0, 0, 0, 0, 0)$$

$$S_1 = (2, 1, 1, 0, 0, 0, 0)$$

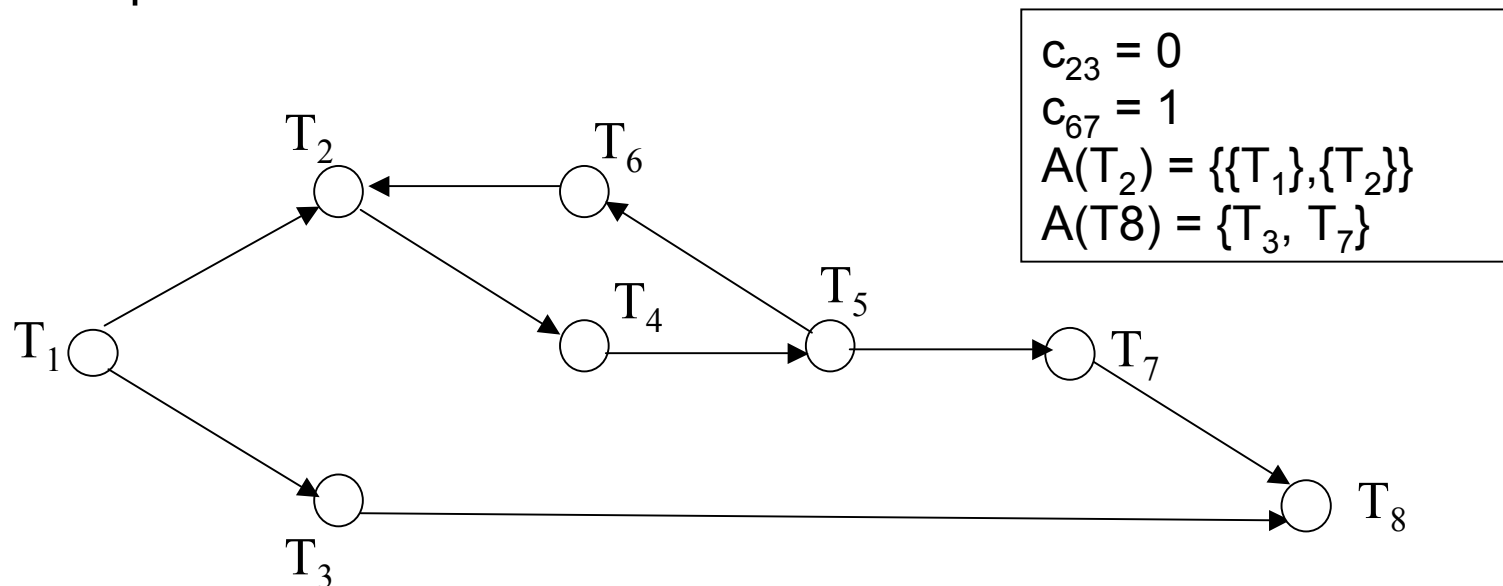
$$S_2 = (2, 2, 0, 1, 0, 0, 0)$$

$$S_3 = (2, 0, 2, 0, 0, 1, 0)$$

... ..

Modeling Power

- Sequential execution
- Conflict (decision making)
- Concurrency
- Synchronization
- Loop



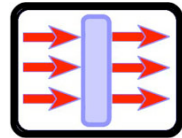
Well-Formed Workflows

- All reachable states form *reachable set* \mathcal{R}
- A workflow is *well-formed* if and only if the following two *behavior conditions* are met:
 - $\forall T_i \in T, \exists S \in \mathcal{R}$ such that $S(T_i) = 1$. (i.e. there is no dangling task.)
 - $\exists S \in \mathcal{R}$ such that $S(T_i) \in \{0, 2\}$ for $\forall T_i \in T$. (i.e. there is at least one ending state.)
- Validation of a WF being well-formed requires the reachability analysis of the WF

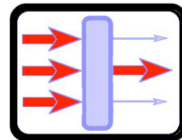
Confusion-Free Workflows

- To simplify workflow modeling and verification
- A confusion-free workflow
 - Is a well-formed workflow
 - Either all tasks triggered by the same task are in conflict, or no pairs of them are in conflict
 - A task becomes executable either when all of its predecessor tasks are executed, or when any one of them is executed

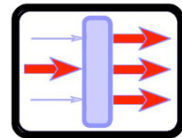
Four types of tasks in a confusion-free workflow



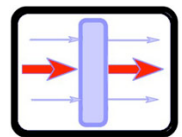
And-In-Parallel-Out



And-In-Conflict-Out



Or-In-Parallel-Out



Or-In-Conflict-Out

Workflow Dynamics

- Theorem for **adding** new tasks to a WF, such that the new WF can preserve the confusion-free, well-formed properties (in the paper).
- Theorem for **deleting** a task from the WF such that the new WF can preserve the confusion-free, well-formed properties.
- Theorem for **changing** business rules that express task dependencies

Tool Support for Editing, Validation and Enactment of WFs

The screenshot displays the 'Work Flow Editor 1.0' application window. The main workspace contains a workflow diagram with tasks T1 through T8. Task T1 is the starting point, branching into T2 and T3. T2 leads to T4, which then leads to T5. T3 leads to T6. T4 and T5 both lead to T6. T6 leads to T7, which finally leads to T8. A 'TaskProperties' dialog box is open over task T6, showing configuration options for triggers, conflicts, and business rules.

TaskProperties

T6 -

Triggered By: (T5)

Triggers: (T2)

Conflicts: T7

Business Rules:

OK

Properties

Appearance	
Color	Black
Description	
Text	T6
Layout	
X	240
Y	73
Workflow	
State	Wait
TaskType	Or_In_Parallel_Out

Toolbox

- Task: Four task icons with different arrow patterns.
- Connection: Arrow icon.
- Note: [Text]

Output

Zoom (+, -)

Properties

X The x-coordinate of the upper-left corner of the task.

Tool Features

- Saving workflow in XML or as an image
- Drag and Drop interface
- Dynamically change tasks/workflow properties
- Zooming in and out to focus on specific sections of the workflow
- Validate workflow
- Visually step through workflow in design window
- Step forward/backward through the simulation
- Auto-play speed adjustment.
- Audit log – for post incident analysis

Tool Usability Analysis

- Usability experiment conducted on 15 people who have no knowledge of workflow, come with different backgrounds (mostly non-technical)
- Proceeds with educating and testing (building a simple model), 45 minutes each
- Three items asked to rate:
 1. “Overall, I am satisfied with how easy it is to use this system”
 2. “I feel comfortable using this system”
 3. “I found the Four Task Types useful in creating workflows”
- Rating scale: 1-7, with 7 the best
- In general, the workflow editor was considered useful and that its usability was acceptable.
 - Item 1 average rate: 5.20
 - Item 2 average rate: 5.47
- But the four types of tasks are not “straightforward enough”
 - Item 3 average rate: 4.00

Future Work

- Interorganizational workflows
- Hierarchical workflows
- Tool improvement