Analysis of Artifact-Centric Workflow

Kamal Bhattacharya, Cagdas Gerede, Jianwen Su
The Analysis Problem: State Machines

- Given a workflow and a goal, do all executions of the workflow satisfy the goal

Artifact types with state machines

Repositories

Work Descriptions: Guarded state machines with actions

Artifact behavior specification

[SOCA 07 ….. ICSOC 07]
Artifact Types

Similar to object-oriented classes augmented with states

Related Work

- Typestate [StromYemini-TSE86]
- Method schemas [Abiteboul et al–PODS90]
- Method structures [Hull et al–FODO88]

Guest Check

- data
  - customerName
  - tableNo
  - paymentDate

- references
  - Receipt
  - Order

- methods
  - addOrder(…)
  - setPaymentDate(…)

- states
  - Pending
    - addOrder
  - Active
    - setCancelDate
    - receiptRequest
  - Payment
    - setPaymentDate
  - Canceled
  - Completed
Work Descriptions

- Describes the business logic
  - Artifact check out / check in
  - Read/Update artifact data via artifact methods
  - Read from external environment
An Operational Model

- **Operational Model** $O = (A, R, W)$
  - $A$: set of artifact types
  - $R$: set of repositories
  - $W$: set of work descriptions
Example: A Restaurant

Artifacts

- Guest Check
- Kitchen Order
- Receipt
- Cash Balance

- Create Guest Check (GC)
- Open GCs
- Add Item
- Pending KOs
- Prepare & Test Quality
- Ready KOs
- Payment
- Update Cash Balance
- Deliver
- Closed GCs
- Archived Receipts
- Archived GCs
- Archived KOs
- Recalculate Receipt
- Pending Receipts
- Disagreed Receipts
Operational Semantics

- Configuration (or snapshot):
  - A finite set of work for each work description
  - A finite set of artifacts spread among repositories and work

- Root configuration: no artifacts checked out

- Note:
  - If there are 5 work for Waiter in a configuration, each derived configuration also has 5 work for Waiter
  - No artifact creation
The Analysis Problem: State Machines

- Given a workflow and a goal, do all executions of the workflow satisfy the goal

Artifact types with state machines + Repositories + Work Descriptions: Guarded state machines with actions

|= satisfies ⪗

[SOCA 07 ...... ICSOC 07]
Artifact Behaviors Specifications

- Every guest check in the *Open Guest Checks* repository has *table#* defined: \( \forall g \text{OpenGCs}(g) \rightarrow \text{Defined}_{\text{table#}}(g) \)
- When a guest check is inserted in the *Closed Guest Check* repository, it can be checked out next by *update cash balance*: \( \forall g \text{ClosedGCs}(g) \rightarrow \text{EN}_{@g} \text{UpdateCB}(g) \)
- For every guest check, there is a way to place it in the *archived repository*: \( \forall g \text{EF}_{@g} \text{ArchivedGCs}(g) \)
- Guest checks are not closed until their kitchen orders are delivered:
  \[ \forall g \neg \exists k \text{Related}(g,k) \land \neg \text{ArchivedKOs}(k) \cup @g \text{ClosedGCs}(g) \]

Related \((g,k) \equiv \text{Equal}(\text{ID}(g), \text{CorrespondingGC}(k))\)
Verification Problems

\((W, S) \models \beta\)
- \(W\) is an operational model
- \(S\) is a root configuration
- \(\beta\) is a behavior specification

\(W \models \beta\)
- For every root configuration \(S\), \((W, S) \models \beta\)
Decidability Results

$\forall (W, S) \models \beta$ is decidable if
- domains are bounded, or
- $\beta$ has only variables over artifacts and domains are unbounded