Analysis of Data-Centric Workflows

Victor Vianu
UC San Diego & INRIA-Saclay
Workflows centered around data

Increasingly common:

• E-commerce
• Supply chain management
• Case management in health-care
• Scientific workflows
• Business processes

....

Complex logic → need for analysis tools

Cautionary tale: healthcare.gov !
Analysis tasks

• Automatic verification of desirable properties
  “no product can be shipped before being paid”
  “if a product is paid, it is eventually delivered or the customer is refunded”

• Comparison, customization, optimization of workflows
  “is my workflow equivalent to yours ?”

• Runtime analysis
  “what can I infer from my local view of the workflow?”
  “what should I do next?”
Long collaboration with INRIA

- Automatic Verification
  - PODS 2008, ICDT 2009, TODS 2009,
  - ICDT 2011, BPM 2011, TODS 2012
- Comparison, customization, optimization of workflows
  - ICDT 2011, TODS 2012
- Runtime analysis for collaborative workflows
  - PODS 2013
What is a data-centric workflow?

- States (data)
- Events (with data)
- Transitions
DB transition system

Infinite-state system
Specifications of data-centric workflows

Two frameworks used in our work:

- Active XML (INRIA Gemo):
  - XML with embedded functions
  - events: function calls

- Business Artifacts (IBM Yorktown)
  - relational database, evolving “artifact” tuple
  - events: pre/post conditions
AXML workflow by example
AXML workflow by example
AXML workflow by example

Mail-Order-Center

Catalog

Customer

Order

Product

"Joe"

"ipod80G"

guard: product available?

Order

Invoice

Customer

Amount

"Joe"

"$399"

!Bill

!Credit

check

!Deliver

Amount

"Joe"

"$399"
AXML workflow by example
AXML workflow by example

Mail-Order-Center

Catalog

Order

Customer: "Joe"

Product: "iPod80G"

Payment: "VISA" "$399"

Request

Customer: "Joe"

Amount: "$399"

Guard: paid correct amount?

Credit check

Deliver

AXML workflow by example
AXML workflow by example

Mail-Order-Center

Catalog

Customer • “Joe”

Order

Product • “ipod80G”

Payment • “VISA” “$399”

Rating • “Good”

!Deliver
AXML workflow by example

guard: rating good or excellent?
AXML workflow by example

guard: rating good or excellent?

Mail-Order-Center

Catalog

Customer

Order

Product

Payment

Rating

Customer

Product

"Joe"

"ipod80G"

"VISA" "$399"

"Good"

Delivery

"Joe"

"ipod80G"
AXML workflow by example

Mail-Order-Center

Catalog

Customer

Order

Product

Payment

Rating

Delivered

“Joe”

“ipod80G”

“VISA” “$399”

“Good”
IBM’s Business Artifacts

Tuple artifacts:

• evolving tuples of data values
• fixed underlying database
• services (pre/post conditions on tuple and a global relational database)
Example

Single artifact record:

<table>
<thead>
<tr>
<th>Status</th>
<th>Prod_id</th>
<th>Ship_type</th>
<th>Coupon</th>
<th>Amount owed</th>
<th>Amount paid</th>
<th>Amount refunded</th>
</tr>
</thead>
</table>

Underlying database:

```
PRODUCTS(id, price, availability, weight)
COUPONS(code, type, value, min_value, free_shiptype)
SHIPPING(type, cost, max_weight)
OFFERS(prod_id, discounted_price, active)
```
services, by example

Current snapshot:

<table>
<thead>
<tr>
<th>Status</th>
<th>Edit shiptype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product id</td>
<td>23</td>
</tr>
<tr>
<td>Ship type</td>
<td></td>
</tr>
<tr>
<td>Coupon</td>
<td></td>
</tr>
</tbody>
</table>

Run (infinite sequence of successive artifacts):

Every precondition is evaluated on the current snapshot.

One is chosen non-deterministically.

<table>
<thead>
<tr>
<th>Status</th>
<th>Edit coupon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product id</td>
<td>23</td>
</tr>
<tr>
<td>Ship type</td>
<td>Fedex 2-day</td>
</tr>
<tr>
<td>Coupon</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
</tr>
<tr>
<td>Fedex 2-day</td>
</tr>
<tr>
<td>$50 coupon</td>
</tr>
</tbody>
</table>

DB
Specifying temporal properties of runs

LTL + statements about data

Linear-time temporal logic

always ( p \rightarrow \text{eventually } q)
LTL(\(\mathcal{L}\))

propositions \(\rightarrow\) formulas in a db language

\(\mathcal{L}\)

\(p\) \(q\)

\(\varphi_p\) \(\varphi_q\)

\[\varphi\]

...
LTL(FO) property:

∀x  always ( p  →  eventually q)

∃y (pay(x,y) ∧ price(x,y))  delivered (x)
Variant for AXML workflows

\[ \forall \overline{X} \text{ always } (p \rightarrow \text{ eventually } q) \]

\[ T_p(X) \]
\[ T_q(X) \]

tree patterns with free variables \( \overline{X} \)
∀ X  \text{always}(p \implies \text{eventually } q)
The verification problem

The diagram shows a verification process where a verifier (VERIFIER) checks an input $A$ and a formula $\varphi$. The verifier outputs 'YES' if $A$ satisfies $\varphi$, and 'NO, counterexample' if not.
Results on verification

Restrictions that guarantee decidability

- Business artifacts: UCSD + INRIA + IBM
- Active XML: UCSD + INRIA

Boundary of decidability, complexity
Results on verification

Restrictions that guarantee decidability

• Tuple artifacts: UCSD + INRIA + IBM
• Active XML: UCSD + INRIA

Efficient implementation: WAVE verifier

Symbolic model checking: marriage of database and CAV techniques
Long collaboration with INRIA

• Automatic Verification
• Comparison, customization, optimization of workflows
  ICDT 2011, TODS 2012
• Runtime analysis for collaborative workflows
  PODS 2013
How to compare different workflows?

Building a beehive

Building an ant nest
Use abstraction!

bee → insect → ant
hive → dwelling → nest

Building a beehive
Building an ant nest
Use abstraction!

- Map to a common abstraction
- Define a notion of simulation
Use abstraction!

Allows to compare very different workflows

Building a beehive
Building an ant nest
AXML vs Artifacts

- AXML workflows can simulate Artifacts with respect to appropriate abstractions

- Artifacts cannot simulate AXML
Long collaboration with INRIA

• Automatic Verification
  
  PODS 2008, ICDT 2009, TODS 2009,
  ICDT 2011, BPM 2011, TODS 2012

• Comparison, customization, optimization of workflows
  
  ICDT 2011, TODS 2012

• Runtime analysis for collaborative workflows
  
  PODS 2013
Collaborative workflows are pervasive

Business processes, health care, scientific workflows, government …
Some possible actions

Author: submit, respond, final, copyright
PC chair: assign, discuss, decide, notify, remind
PC member: enter review, invite referee, discuss
The collaborative workflow model

Simple peer views

actions & data
actions & data
actions & data
actions & data

Author
PC chair
PC member
Referee

virtual database

virtual database

virtual database

virtual database
Peer actions

\[ \text{Update} \quad :\quad \text{Condition} \]

- sequence of insertions/deletions into the view
- conjunction of literals in the view

**PC Chair**: \[ \text{Assign}(\text{Id, PCmember}), \quad \text{Status}(\text{Id, assigned}) \]
\[ \quad :\quad \text{Submitted}(\text{Id, title, author, abstract}), \quad \neg \quad \text{Status}(\text{Id, assigned}) \]
\[ \quad \quad \text{PC(PCmember)}, \quad \neg \quad \text{Conflict}(\text{author, PCmember}) \]
Focus:
runtime peer reasoning about global run based on local observations

• Monitoring events
  Has my paper been accepted?

• Diagnosing anomalous behavior
  A paper was assigned to a PC member with a COI: could the assignment have been made by the PC chair, or was it necessarily made by a track chair?

• Competitive advantage
  Is my own submission as a PC member still under discussion?
Runtime peer reasoning about global run based on local observations

Trace $\rho$

Infinitely many compatible global runs
Specifying properties of global runs

PLTL-FO

PLTL:
past linear-time propositional temporal logic

PLTL-FO:
PLTL with propositions interpreted as FO statements
Specifying properties of global runs

PLTL-FO

• My paper (submission 12) has not yet been accepted nor rejected:

\[
\text{always } ^{-1} \rightarrow \exists \text{ review } (\text{Accept}(12, \text{ review}) \lor \text{Reject}(12, \text{ review}))
\]
Specifying properties of global runs

**PLTL-FO**

- My paper (submission 12) has not yet been accepted nor rejected:

  \[
  \text{always}^{-1} \rightarrow \exists \text{ review } (\text{Accept}(12, \text{ review}) \lor \text{Reject}(12, \text{ review}))
  \]

- Semantics of a formula \( \phi \): relative to a trace \( \rho \)

  \[
  \text{Cert}(\phi) : \ \phi \text{ holds in every global run compatible with } \rho
  \]

  \[
  \text{Poss}(\phi) : \ \phi \text{ holds in some global run compatible with } \rho
  \]
Goal

Given a workflow specification $W$, a trace $\rho$ and an PLTL-FO property $\phi$ of global runs evaluate $\text{Cert}(\phi)$ and $\text{Poss}(\phi)$

Theorem: $\text{Cert}(\phi)$ and $\text{Poss}(\phi)$ can be evaluated in $\text{PSPACE}$ with respect to $\phi$ and $\rho$. 
Proof idea

Trace $\rho$

Infinitely many compatible global runs

Symbolic transition system
Proof idea

Trace $\rho$

symbolic instances
variables + constraints
Proof idea

Trace \( \rho \)

Transitions among symbolic instances each computed in PSPACE
Proof idea

Finite transition system on which Poss(φ) can be evaluated by a nondeterministic PSPACE search

Cert(φ) = ¬ Poss(¬ φ)
Other results

- Incremental evaluation
- Pre-emptive monitoring

\[ \varphi \text{ is not known a priori} \]

Can be done for classes of properties sharing a temporal type: underlying propositional PLTL formula

Example: \textcolor{red}{always}^{-1}(p \rightarrow \textcolor{red}{sometime}^{-1} q)
Other results

- Incremental evaluation
- Pre-emptive monitoring
- Using knowledge about global run in workflow specifications

**PC member**

As long as I don’t know that my paper is accepted, I will reject every other paper

\[
\text{Reject(ID) :- paper(ID, author), author \neq self, \neg cert ("my paper is accepted")}
\]

epistemic atom
Summary

• Models for data-centric workflows
• Automatic verification
• Framework for comparing workflows
• Runtime analysis for collaborative workflows

Better tools for the design and analysis of data-centric workflows
Summary

• Models for data-centric workflows
• Automatic verification
• Framework for comparing workflows
• Runtime analysis for collaborative workflows

Exciting area, much more to be done!
Thank you!

Merci!