Mining and visualizing Behavioral Software Processes

TU/e research - current status

Nour Assy Cong Liu Bart Postma

> TUe Technische Universiteit Eindhoven University of Technology

oniversity of rectinio

EXPERIENCE.

1. 334 E.

Where innovation starts

BSR meeting - 4/12/15

TU/e: bigger picture



General idea



Documenting software architecture – Different views





Modeling structure:

- Components: processing units and data stores
- **Connectors:** form of interactions among components (e.g. call-return)
- Ports: point of interaction on a component





Taxonomy of connectors





Modeling behavior:

- How the system is actually working?
- Is it working as intended?
- Using behavioral models such as Activity diagrams



General idea



General idea



- Software open source
- Components are known → plugins in ProM
- Connector type: call-return (Peer to peer)
 - Internal calls: calls within a component
 - Inter-calls: calls between components
 - Intra-calls: calls to shared libraries

1. Code instrumentation and generation of software event data

 Log information on components' method calls and connector type

A snapshot of event log in CSV

Case1	<init>()</init>	OpenLogFilePlugin	org.processmining.plugins.log	Open XES Log File (Naive)	Open XES Log File (Naive)	internal
Case1	<init>()</init>	OpenNaiveLogFilePlugin	org.processmining.plugins.log	Open XES Log File (Naive)	Open XES Log File (Naive)	internal
Case1	getInputStream()	OpenLogFilePlugin	org.processmining.plugins.log	Open XES Log File (Naive)	Open XES Log File (Naive)	internal
Case2	extractName()	XConceptExtension	org.deckfour.xes.extension.std	Mine Fuzzy Model	XES Library	intra
Case2	toString()	XExtension	org.deckfour.xes.extension	Mine Fuzzy Model	XES Library	intra
Case2	instance()	XLifecycleExtension	org.deckfour.xes.extension.std	Mine Fuzzy Model	XES Library	intra

2. Our XES log structure

```
<event>
    <string key="concept:name" value="&lt;init&gt;()\\XParser\\org.deckfour.xes.in"/>
    <string key="lifecycle:transition" value="complete"/>
    <string key="Activity" value="&lt;init&gt;()"/>
    <string key="Class" value="XParser"/>
    <string key="Package" value="org.deckfour.xes.in"/>
    <string key="Runtime_Component" value="Open XES Log File (Naive)"/>
    <string key="Belonging Component" value="XES Library"/>
    <string key="Interaction_Type" value="intra"/>
    <string key="Start_Time" value="1449067337846567315"/>
    <string key="End_Time" value="1449067337846573302"/>
    <string kev="Parameter_Types" value="Null"/>
    <string key="Return_Type" value="Null"/>
    <string key="Parameter Value" value="Null"/>
    <string key="Return Value" value="Null"/>
</event>
<event>
    <string kev="concept:name" value="&lt;init&gt;()\\XsDateTimeConversion\\org.deckfour.xes.util"/>
    <string key="lifecycle:transition" value="complete"/>
    <string key="Activity" value="&lt;init&gt;()"/>
    <string key="Class" value="XsDateTimeConversion"/>
    <string key="Package" value="org.deckfour.xes.util"/>
    <string key="Runtime Component" value="Open XES Log File (Naive)"/>
    <string key="Belonging Component" value="XES Library"/>
    <string key="Interaction Type" value="intra"/>
    <string key="Start Time" value="1449067337847406797"/>
    <string key="End Time" value="1449067337847594109"/>
    <string key="Parameter_Types" value="Null"/>
    <string key="Return_Type" value="Null"/>
    <string key="Parameter_Value" value="Null"/>
    <string key="Return Value" value="Null"/>
</event>
```

3. Mining algorithm output: Activity diagram



Our next steps...

Short term

Developing the mining algorithm

• Mid term

- Generalizing our 1st scenario:
 - Components are not known
 - Discovering different interaction types (connector types):
 - Peer to peer, client-server, Datastream, Data access

Our next steps...

• Long term:

- Conformance checking and Delta analysis with an existing C&C view
 - How to detect violations?
 - How to visualize anomalies?
- How to discover the architectural style (C&C diagram) from the discovered behavioral model?

And now visualization...



Events

Software events

Event visualizations

Software visualizations





Experimenting



Collaborations and contacts

TUE's AIS group on Process Mining

UT's FMT group on Parallel Systems

TUE colleagues for datasets

Industry partners



Taxonomy: The science of classification



Single events



Single events

Sequences of events



Single events

Sequences of events

Sets of event sequences



Single events

Sequences of events

Sets of event sequences

Their corresponding visualizations

Software events

Software processes



Taxonomy

Future work/hopes

Finish draft taxonomy, iteratively refine

Increase understanding of software events and big software

Acquire and analyze datasets

Develop prototypes, in close collaboration

Thank you for listening

Nour Assy n.assy@tue.nl Cong Liu c.liu.3@tue.nl

Bart Postma b.j.j.postma@tue.nl