State Dependence in Performance Evaluation of Component-Based Software Systems

Barbora Bühnová and Lucia Kapová

Masaryk University, Czech Republic
Universität Karlsruhe, Germany

DSRG Seminar, Prague

October 7, 2009
Aim of the talk

To answer the following questions:

- What is the state of a component-based system (CBS)?
- What types of states can be identified in CBSs?
- How to extend an existing comp. model to reflect the state?
- What accuracy does the state bring to our predictions?
- What size increase does the state mean for our model?

Note: Joint work of Lucia Kapova, Barbora Buhnova, Anne Martens, Jens Happe, and Ralf H. Reussner. To be presented at WOSP 2010.
Introduction
Motivation and challenges

State Characterization
State definition
State categorisation

State Dependency Analysis
Prediction framework
Analysis of state categories
Validation on a case study

Conclusion
Summary and discussion
Future directions
Motivating example

Messaging system with transactions

- Guaranteed order of delivered messages
- Messages collected first, and then delivered
- A measurement for 1000 messages/transaction below
Challenges of stateful analysis

Challenges of stateful analysis

- State definition
- Performance impact
- Prediction difficulty
- State support in component models

Overview of contributions

(i) State definition and categorisation
(ii) Extension of an existing performance prediction model
(iii) State-dependency analysis
State definition

How do we understand a state?

- Information remembered inside the system
- Typically context or history-dependent
- Used to navigate system behaviour

In literature

- **Explicit state** – an additional information in the model
- **Implicit state** – an information about the current position in system behaviour
State categorisation for CBSs

(i) **Place dimension** answers the question: *Is the state proprietary to a component/system/user?*

(ii) **Time dimension** answers the question: *Is the state initialised or changed at run/deployment/instantiation time?*

<table>
<thead>
<tr>
<th>Component</th>
<th>Run time</th>
<th>Deployment time</th>
<th>Instantiation time</th>
</tr>
</thead>
</table>
| Component | (a) Protocol state  
(b) Internal state  | (c) Allocation state  | (d) Configuration state |
| System    | (e) Global state  | (f) Allocation state  | (g) Configuration state |
| User      | (h) Session state  
(i) Persistent state |  |  |
Prediction framework

Palladio Component Model (PCM)

- Performance prediction in CBSs

Existing state-relevant constructs:

- Usage profile propagation and parameter dependencies
- Static component parameters (not modifiable at run time)
- Limited passive resources (semaphores, threads from a pool, or memory buffers)

New internal state construct:

- Component-internal state (with get/set methods)
Diversity among state categories

Similarities in state classes

- **Allocation vs. Configuration state**: both fixed before run-time
- **System vs. Component-specific states**: similar when we abstract from component boundaries
- **Session vs. Persistent state**: differs only in the length of a session

Classes to be analysed

- Protocol state
- Internal state
- Allocation state
- Session state
Protocol state

Performance impact

- Usage-profile knowledge
- Predictability of its propagation

Model-size costs

- Store of the actual state value

B. Bühnová and L. Kapova
State Dependence in Performance Evaluation
October 7, 2009 10 / 17
Internal state

Performance impact

- Correlation of subsequent branches

Model-size costs

- Connection of comp. behaviour with state value
- Independence of state-guarded branches
- Number of state updates
Allocation state

Performance impact

- Independent on usage-profile knowledge
- Dependent on deployment-details knowledge

Model-size costs

- Typically smaller than without a state
Session state

Performance impact

- Dependent on input-values knowledge
- Correlation of branches (always high)

Model-size costs

- Connection of comp. behaviour with state value
- Lower increase implied by high correlation
- No update of state values
Validation on a case study

Messaging System of SPECjms2007 Benchmark

- Recall the motivating example
- Modelled in extended PCM
- Analysed and compared to measurements

<table>
<thead>
<tr>
<th>Transaction Size</th>
<th>Measurement (Median)</th>
<th>Prediction (Median)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,665 917ms</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2,506 566ms</td>
<td>2,609 999ms</td>
</tr>
<tr>
<td>4</td>
<td>4,157 104ms</td>
<td>4,619 999ms</td>
</tr>
<tr>
<td>10</td>
<td>9,145 595ms</td>
<td>9,050 000ms</td>
</tr>
<tr>
<td>20</td>
<td>17,012 373ms</td>
<td>17,079 999ms</td>
</tr>
<tr>
<td>100</td>
<td>82,752 583ms</td>
<td>85,440 000ms</td>
</tr>
<tr>
<td>400</td>
<td>356,843 626ms</td>
<td>360,980 000ms</td>
</tr>
<tr>
<td>1000</td>
<td>943,539 863ms</td>
<td>943,370 000ms</td>
</tr>
</tbody>
</table>
We discussed the following questions:

• What is the state of a component-based system (CBS)?
• What types of states can be identified in CBSs?
• How to extend an existing comp. model to reflect the state?
• What accuracy does the state bring to our predictions?
• What size increase does the state mean for our model?
Future directions

To be discussed here :)
Thank you for your attention!

Any questions?