Software Process Simulation Modeling

HyeonJeong Kim
2009. 2. 25
Part I: Definition of Software Process Simulation Modeling (SPSM)
Part II: Overall procedure to build SPSM
Part III: Research issues
Definition of SPSM (1/3)

- **Software Process**
  - A set of *procedures* and *methods*
    - Define the relationship of tasks[^1]
  - Also a part of *process, people, technology* triad[^2]

[^1]: Define the relationship of tasks
[^2]: Process, people, technology triad
Definition of SPSM(2/3)

- Simulation modeling
  - A method to specify real system at some abstraction level

`<Real world>`
- Characteristics
  - System uncertainty
  - Change over time
  - Learning mechanism

Mapping

`<Simulation model>`
- Corresponding concept
  - Stochastic simulation
  - Dynamic behavior
  - Feedback loop

© KAIST SE LAB 2009
Software Process Simulation model (SPSM)

- System specification describes how the process works
  - SPSM should contain all three elements of triad

Process

People

Technology

<Project environments in SPSM>
Overall procedures to build SPSM

- Requirements analysis
  - Define the objectives of simulation
  - Gather model elements

- Design
  - Deriving qualitative information
  - Formal modeling

- Implementation
  - Implement the simulator

- Testing (V&V)
  - Verification
  - Calibration
  - Validation

- Analysis
  - Collect the data from simulator
  - Analyze the collected data
Define the objectives (1/2)

- Determine why the simulator is needed
  - Main categories of the objectives in SPSM\(^3\)
    - Most simulators are made for supporting the manager’s decision making

Strategic management
Process improvement and technology adoption
Planning
Control and operational management
Define the objectives (2/2)

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Strategic management</th>
<th>Planning</th>
<th>Control &amp; operational management</th>
<th>SPI &amp; tech. adoption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area that objective deals with</td>
<td>Deciding on the organizational strategies or policies</td>
<td>Predicting/forecasting some project parameters across time</td>
<td>Project tracking and oversight on software project during enactment</td>
<td>Forecasting the impact of a potential process (tech) change before enactment</td>
</tr>
<tr>
<td>Example</td>
<td>• Decide development site (cross sites vs. centralized at one site)</td>
<td>• Predicting staffing levels across time and periodical re-planning</td>
<td>• Predicting which activity threatens product quality</td>
<td>• Analyzing impacts of the inspection on product quality</td>
</tr>
<tr>
<td></td>
<td>• Decide staff strategies (in-house vs. outsource)</td>
<td></td>
<td></td>
<td>• Analyzing impacts of automation of test design on effort</td>
</tr>
</tbody>
</table>
Gather the model elements (1/2)

- Model elements
  - Identified information elements
    - To be needed for the defined objectives in simulation

![Diagram of Simulation model and Model scope]

© KAIST SE LAB 2009
Gather the model elements (2/2)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Model scope</th>
<th>Result variables</th>
<th>Model abstraction</th>
<th>Input parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Boundary of the simulation model in organizational view and time-span</td>
<td>Information elements needed to answer the key questions related to the objectives</td>
<td>Key elements of software process and project environments</td>
<td>Information elements related to result variables and model abstractions</td>
</tr>
<tr>
<td>Example</td>
<td>• Organizational view - less than one, one, multiple projects(teams)</td>
<td>• Effort/cost • Schedule • Quality(# of defects) • Productivity • ROI(Return on Investment) • Staff utilization rate (staffing needs)</td>
<td>• Key process components and their level • Sequence of the activities • Available resources(staff, hardware) • Iteration loops</td>
<td>• Software size(KSLOC) • Defect detection efficiency • Defect removal and injection rates • Requirements creeping pattern • Resource constraints</td>
</tr>
</tbody>
</table>
What to do in design

- Deriving the qualitative information
  - Define a cause-effect relationship from input to result variables
- Formal modeling for the qualitative information

"Roof size depends on size of dog house"

Roof size $a = 1.2 \times$ dog house size $b$
Deriving the qualitative information (1/2)

Qualitative information

- Contain a cause-effect relationship between parameters
  - It describes the atomic behavior of real system (process)
- Causal diagram (cause-effect diagram)\(^4\)
  - Useful diagram to describe qualitative information

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="X" alt="Cause" /> <img src="Y" alt="Effect" /> (\rightarrow) (\uparrow)</td>
<td>If X increases (decreases), then Y increases (decreases)</td>
<td><img src="+" alt="Effort" /> <img src="+" alt="Schedule" /></td>
</tr>
<tr>
<td><img src="X" alt="Cause" /> <img src="Y" alt="Effect" /> (\rightarrow) (\downarrow)</td>
<td>If X increases (decreases), then Y decreases (increases)</td>
<td><img src="-" alt="Productivity" /> <img src="+" alt="Effort" /></td>
</tr>
</tbody>
</table>
Example of a causal diagram\[4\]

- Useful to identify the important dependencies for entire model
- Feedback loop
  - Represent learning mechanism from previous value
Simulation model needs a formal description of system behavior

- *Mathematical representation* is required
  - Based on observation of real projects or literatures

<table>
<thead>
<tr>
<th>Qualitative information</th>
<th>Formal description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort – Schedule</td>
<td>[ \text{Schedule} = \int (\text{Effort} - \ldots) ds + \text{Schedule}_{t_0} ]</td>
</tr>
<tr>
<td>Productivity + Effort</td>
<td>[ \text{Effort} = \int (\text{productivity} + \ldots) ds + \text{Effort}_{t_2} ]</td>
</tr>
</tbody>
</table>
Implement a simulator

- Simulator
  - Computer program to interpret the simulation model’s instruction

- Main simulation methods to build simulator for software process
  - System Dynamics
  - Discrete Event-based Simulation
  - Hybrid simulation
System dynamics (SD)

- A method to enhance *understanding dependencies between factors* in complex system[^4][^5]

Example

- Project environments which change dynamically and continuously over time and their changes affect other variables
  - In SD, all the parameters are automatically re-calculated on every single time unit
System dynamics (Cont’d)

- Result of simulator depends on single time unit

<table>
<thead>
<tr>
<th>Calculation # in one run</th>
<th>Total time Simulation</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 times</td>
<td>Long</td>
<td>High</td>
</tr>
<tr>
<td>4 times</td>
<td>Short</td>
<td>Low</td>
</tr>
</tbody>
</table>
Method to build simulator (3/5)

- Discrete Event-based Simulation (DES)
  - A method to enhance understanding interactions between components of complex system[6]
    - States of software process are changed by
      - Start/end of an activity or reception/release of an artifact
        » Irrelevant to time
        » Events represent interactions between process activities
Method to build simulator (4/5)

**SD vs. DES for software process**

<table>
<thead>
<tr>
<th></th>
<th>SD (System Dynamics)</th>
<th>DES (Discrete Event-based Simulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>- Dependency between parameters</td>
<td>- Easily describe discrete process steps</td>
</tr>
<tr>
<td></td>
<td>- Feedback loops</td>
<td>- Attributes of each process entity</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>- Difficult to model discrete process steps, and attributes of each process entity</td>
<td>- Difficult to make feedback loops</td>
</tr>
</tbody>
</table>

- **Project environments based on feedback loop**
  - Provide a hybrid simulation method to combine SD and DES
- **Process based on events**
Hybrid simulation method
- Combine DES and SD
  - DES is responsible for describing software process
  - SD is responsible for describing feedback loop
- Example
  - Choi’s work[7]
    - Combine SD into DES framework
Verification and validation

Real project

Validation

Simulation model

Verification

Simulator

Validation

Simulator

Validation

Simulation model spec

Verification

Simulator

Calibration

Similar result?

Real project

Inputs

Simulation model

Calibration

Similar result?

Implementation

Compare

Result

Calibration

Produce

© KAIST SE LAB 2009
Sensitivity analysis

- **Purpose**
  - Determining which input parameters have a significant impact on the result\(^8\)
    - Identified main parameters should be carefully treated in management

- **Method**
  - For each input parameter
    - Its minimum and maximum value are simulated under setting other parameters to have a fixed value
Sensitivity analysis (Cont’d)

Example

- Effect of input parameters of GSD* on duration to complete software project[9]

- Top 3 significant elements on duration for GSD
  - Overlapping working hour
  - Distribution overhead
  - Distance

*GSD: Global Software Development
Research issues (1/2)

Define the objectives of simulation

Gather model requirements and analyze them

Build qualitative model
Collect quantitative information

For developing a general simulation model
- ‘plug and play’ model components

For new development process
- GSD, business process with SW process, ...

For deciding managerial policies
- Req. Creeping

Re-planning
- Resource allocation

Implement the simulator

Collect the data from simulator

Analyze the collected data

Verification
Calibration
Validation
Research issues (2/2)

- Define the objectives of simulation
- Gather model requirements and analyze them
- Build qualitative model
- Collect quantitative information
- Implement the simulator

V&V for SPSM with limited data
- With minimum data sets, provide the useful, proven SPSM

Focus on analyzing the results
- Combine with statistics in stochastic simulation model

Define the objectives of simulation
Gather model requirements and analyze them
Build qualitative model
Collect quantitative information
Implement the simulator

Verification
Calibration
Validation

Collect the data from simulator
Analyze the collected data

© KAIST SE LAB 2009
References


Thank You.