Trustworthy Software Process Modeling Methods and Key Technologies

Software Process Modeling Research Group

Lab for Internet Software Technologies, ISCAS
Agenda

- Overview of the Research Group
- Research Background
- Research Areas and Framework
- Research Progress
- Publications
Agenda

- Overview of the Research Group
- Research Background
- Research Topics and Framework
- Research Progress
- Publications
Overview of the Research Group

- Research Focus: software process modeling methods and technologies
- Faculty
  - Prof. Mingshu Li
  - Prof. Qing Wang
  - Prof. Yongji Wang
  - Associate Prof. Feng Yuan
  - Assistant Prof. Junchao Xiao
  - Assistant Prof. Qiusong Yang
Overview of the Research Group

- Students
  - Ph. D. (4): Shen Zhang, Jian Zhai, Dapeng Liu, Lizi Xie

- Supports
  - “Research on Trustworthy Software Process Management and Risk Control Models and Methods”, National Natural Science Foundation of China under grant No. 90718042
  - “Production Line Supporting Trusted Software Process Management”, National Hi-Tech Research and Development Program of China under Grant No. 2007AA010303
  - “Requirement Evolvement Modeling, Requirement Engineering – Basic Research of Software Engineering for Complex System”, National Basic Research Programs of China under Grant No. 2007CB310802
Agenda

- Overview of the Research Group
- Research Background
- Research Areas and Framework
- Research Progress
- Publications
Research Background

- **Trustworthy**
  - People trust quality of the software they used
  - Software process is a key factor in software projects which ensures the software quality
  - Trustworthy software process is necessary for trustworthy software
  - Behaviour, performance and risks of trustworthy software processes are predictable and controllable
Research Background

● Our Goal: trustworthy software processes

● For the goal, we need:
  ● Propose new methods and technologies to improve the trustworthiness of activities, such as risk management, and process measurement
  ● Investigate the mechanism integrating different activities, approaches and technologies into trustworthy software processes
  ● Develop effective software tools to ensure the strict execution of trustworthy software processes
Agenda

- Overview of the Research Group
- Research Background
- *Research Areas and Framework*
- Research Progress
- Publications
Our Research Areas

- **Software Process Modeling**
  - How to model software processes formally?
  - How to validate and verify software process models?

- **Human Resources Scheduling**
  - How to assign proper persons to tasks?
  - How to maximize an organization’s benefits by scheduling human resources among multiple projects?
  - How to schedule when human resources are not enough?

- **Software Process Simulation**
  - How to deal with risks when software projects are ongoing?
  - How to predict the performance of software processes?
Agenda

- Overview of the Research Group
- Research Background
- Research Areas and Framework
- Research Progress
- Publications
Software Process Modeling

- Component-based software process modeling method
- Software Process Verification
Component-based Software Process Modeling Method

- Software process reuse receives broad attention in software process modeling field
  - Benefit (high quality/productivity, low cost)
  - Weakness
    - Different description method blocks communication
    - Different granularity
    - Lack of formalization, depends on user’s comprehension
    - Lack of operation rules, tends to be error used
Method Overview

- Our Method
  - Define Software Process Component (SPC) formally
    - Uniforms the description and granularity
  - Define SPC operational rules
    - Ensures the right usage of SPC
    - Ensures modeling process can be strictly recorded and traced
Definition of SPC

- **Structure of SPC**
  - Interface, input, output, boundary, name, behavior
Definition of SPC operation

- Connection
- Adding
- Removing
- Extension
- Shrinking
An Example
Software Process Modeling

- Component-based software process modeling method
- *Software Process Verification*
Software Process Verification

- To ensure the validity of process models
- Verify the properties of process models
  - deadlock
  - resource racing
  - existence of specific action
  - temporal relationship among specific actions
- Results feedback to correct models
Human Resource Scheduling

- Optimized Task-Human Resource Matching
- Value-based Multiple Software Projects Scheduling with Genetic Algorithm
- Preemptive Human Resource Scheduling
Human is the core resource in software projects.
The productivity varies hugely among different persons.
The work efficiency of the human can be affected by many factors.
We try to develop an optimized Task-Human Resource matching method which takes all the technical skills, personality and career plan into consideration.
We adopt the skill description in OEC-SPM

We have designed a questionnaire to get the relation between capability and personality

In most of the software companies, the two career paths are Management and Technique
Human Resource Scheduling

- Optimized Task–Human Resource Matching
- Value-based Multiple Software Projects Scheduling with Genetic Algorithm
- Preemptive Human Resource Scheduling
Value-based Multiple Software Projects Scheduling with Genetic Algorithm

- Multi-project environment
  - Contention resource requirements among multiple projects
  - The projects may have different stakeholders who bear different requirements and preferences
  - Each project holds different constraints and different value objectives

- One of the goals of an organization
  - Achieve the maximum value from the projects and response to the changing market timely
Value-based Multiple Software Projects Scheduling with Genetic Algorithm

● An example

- Benefit
  - Increase in customer satisfaction
  - More money earned by the organization

- Penalty
  - Compensation asked for by the customer
  - Decrease in customer satisfaction

Annotations:
RA  - Requirement Analysis;
AD  - Architecture Design;
IMP - Implementation;
WTC - Write Test Case;
TST - Testing;
URA - Upgrading
  Requirement Analysis;
DD  - Detailed Design;
COD - Coding
## Value-based Multiple Software Projects Scheduling with Genetic Algorithm

### Projects Constraints

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost constraint</td>
<td>2*10^5</td>
<td>5*10^4</td>
<td>3.5*10^5</td>
</tr>
<tr>
<td>Preference</td>
<td>Cost preference</td>
<td>Schedule preference</td>
<td>Cost preference</td>
</tr>
<tr>
<td>Schedule ahead benefit ($/day)</td>
<td>400</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Schedule postpone penalty ($/day)</td>
<td>400</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Cost saved benefit ($)</td>
<td>Equal to saved</td>
<td>Equal to saved</td>
<td>Equal to saved</td>
</tr>
<tr>
<td>Cost exceeded penalty ($)</td>
<td>Equal to exceeded</td>
<td>Equal to exceeded</td>
<td>Equal to exceeded</td>
</tr>
<tr>
<td>Project failure penalty($)</td>
<td>10^6</td>
<td>10^6</td>
<td>10^6</td>
</tr>
<tr>
<td>Project importance preference</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Problems in multi-project scheduling

- Define the value obtained by scheduling according to constraints, value objectives and possible scheduling results in projects.
- Provide a multi-project scheduling method which can obtain the (near-) maximum value for the organization.
Our Method

- Value-based multiple software projects scheduling method using a genetic algorithm
  - Value function in multi-project environments is defined to guide the scheduling
  - Genetic algorithm (GA) is adopted to tackle this problem to get optimal or near-optimal solutions with high efficiency
    - Encode the scheduling problem as the chromosome
    - Use value function as fitness to direct the evolution
Multi-project Value Function

Value of an organization

Value of project P1
  Schedule value of P1
  Cost value of P1

Value of project P2
  Schedule value of P2
  Cost value of P2

......
Structure of the Chromosome

- Encode

One Activity

One Capable Resource

One Capable Resource

Code

Code

......

......

Priority for A_1

Priority for A_N

Priority genes

Human resource genes

\[
\begin{array}{cccccccccccc}
0/1 & 0/1 & ... & 0/1 & 0/1 & ... & 0/1 & ... & 0/1 & 0/1 & ... & 0/1 & 0/1 & ...
\end{array}
\]

\[
\begin{array}{cccccccccc}
HR_{1,1} & HR_{1,2} & ... & HR_{1,t_1} & HR_{2,1} & HR_{2,2} & ... & HR_{2,t_2} & ... & HR_{N,1} & HR_{N,2} & ... & HR_{N,t_N}
\end{array}
\]

Size = g

Size = g
Structure of the Chromosome

- Decode

Human resource genes

\[
\begin{array}{cccccccc}
A_1 & A_2 & \cdots & A_N \\
0/1 & 0/1 & \cdots & 0/1 & 0/1 & 0/1 & \cdots & 0/1 \\
HR_{1,1} & HR_{1,2} & \cdots & HR_{1,N_1} & HR_{2,1} & HR_{2,2} & \cdots & HR_{2,N_2} \\
\end{array}
\]

Size = g

Priority genes

\[
\begin{array}{cccccccc}
\text{Priority for } A_1 & \cdots & \text{Priority for } A_N \\
0/1 & \cdots & 0/1 & 0/1 & \cdots & 0/1 & \cdots & 0/1 \\
HR_{N_1,1} & HR_{N_1,2} & \cdots & HR_{N_1,N_N} & HR_{N_2,1} & HR_{N_2,2} & \cdots & HR_{N_2,N_N} \\
\end{array}
\]

Size = g

activities that do not have precedent activities or whose precedent activities have been assigned

Priority for \( A_i \)

Priority for \( A_j \)

Scheduled Resources for \( A_i \)

Scheduled Resources for \( A_j \)
Three simulation runs of the algorithm
Human Resource Scheduling

- Optimized Task-Human Resource Matching
- Multiple Software Projects Scheduling
- Preemptive Human Resource Scheduling
Task Priority Based Preemptive Human Resource Scheduling Method

- Software process is of high uncertainties.
- The human resource scheduling must be flexible.
- As the project environment changes all the time, resource re-plan must be supported.
- To get the maximal utilization of limited human resource, A Task Priority Based Preemptive Human Resource Scheduling Method was suggested.
Structure of the Scheduling Method

- Task priority sequence
- Value based task priority model
- Tasks

Negotiation Protocol

Process Agents (When resource is not enough, preemptive resource scheduling will occur)

Human resource

Resource schedule result
Scheduling Rule

- When searching suitable resource combination, if the available free human resource is not enough, Task priority model will be used by PA to do preemptive human resource scheduling.
  - First, when searching suitable human resource combination for the given task, free human resources and human resources occupied by lower priority tasks compared with the target task are both considered.
  - Second, if there is more than one tasks whose resource can be robbed, PA will choose the task of lowest priority.
The Extension of Process-Agent

- Process agent
  - Engines
  - Knowledge
    - Experience lib
    - Process knowledge
  - Description knowledge
    - Capability
    - Resource

- Human resource
  - Cost
  - Work calendar
  - Human resource
  - Human resource
  - Human resource
Software Process Simulation

- Stochastic Process Algebra Based Software Process Simulation
- Risk Driven Project Buffer Allocation and Simulation
- Requirements Volatility Simulation
- Employee Turnover Solution Simulation
Stochastic Process Algebra Based Software Process Simulation

- Two models
  - Software process model
    - Software design notation
    - Formal method
    - Multi-agents system
    - ......
  - Software process simulation model
    - System dynamic (SD)
    - Discrete event simulation (DES)

- A big gap between them

- Our goal: derive simulation models from traditional descriptive process models
To Uniform the Two Models

- s-TRISO/ML language has been introduced
  - Derive simulation model from process model
  - Only one model needs to be built
  - Stochastic process algebra based approach
  - Simulate uncertainties on time of a process
  - Give time related simulation results
Framework

Analysis

Descriptive Process Model

Workflow Chart

Simulation

Stochastic parameters assignment

s-TRISO/ML Model

Support

PEPA

Support

Stochastic π-calculus
Simulation: Step by Step

- Describe process by s-TRISO/ML
- Transform s-TRISO/ML into stochastic $\pi$ -calculus by the mapping rules
- Assign the stochastic parameters
- Input model into PEPA and simulate
- Analyze the result
Software Process Simulation

- Stochastic Process Algebra Based Software Process Simulation
- Risk Driven Project Buffer Allocation and Simulation
- Requirements Volatility Simulation
- Employee Turnover Solution Simulation
Risk Driven Project Buffer Allocation and Simulation

- Software projects is of great uncertainties
- In general, project managers will keep some time buffer in the tail of project plans
- However, buffer often causes too many plan changes, bringing more cost and delay
- The core question is: how should project managers use the time buffer properly?
- Our solution: risk driven project buffer allocation and simulation
  - Allocate buffer to tasks based on risks happened to them
Structure of the Buffer Allocation Method
The Project Buffer Allocation Process

1. Input Info.
2. Set task relation
3. Get ideal schedule
4. Critical chain analyze
5. Buffer Allocation Process
6. Process Non-Critical tasks
7. Output allocation result
An Example

Traditional project buffer allocation

Risk driven project buffer allocation
Simulation Results

• Project plan changes
Software Process Simulation

- Stochastic Process Algebra Based Software Process Simulation
- Risk Driven Project Buffer Allocation and Simulation
- Requirements Volatility Simulation
- Employee Turnover Solution Simulation
Requirements volatility/change is one of the most frequent and severe risks in software projects.

Requirements volatility is a complex problem due to the relationship between requirements and many other factors, such as productivity, cost, defects, and so on.

Software project managers need methods and tools to estimate the impacts of requirements volatility accurately and
Requirements Volatility Simulation

- Our method: RVSim

Fig. 1. The RVSim simulation model
Requirements Volatility Simulation

● We developed a tool RVSimulator, and put it into actual software projects.

● The feedback from projects showed the following shortcomings:
  ● The current RVSim is only for waterfall process
  ● The relationship between requirements and tasks is fixed
  ● Maintenance work is not considered in calculating the impacts
  ● Only the impacts on project duration is supported
Requirements Volatility Simulation

- To address above limitations, we proposed an improved version RVSim2
  - Support more software processes
  - The relationship between requirements and tasks are more flexible
  - Maintenance work is taken into consideration
  - The estimate of impacts on project efforts is supported
Requirements Volatility Simulation

- **RVSim2**

Fig. 1 The Framework of the RVSim2 Approach
Software Process Simulation

- Stochastic Process Algebra Based Software Process Simulation
- Risk Driven Project Buffer Allocation and Simulation
- Requirements Volatility Simulation
- Employee Turnover Solution Simulation
Employee Turnover Solution Simulation

- Employee turnover is another frequent and severe risk in software projects.
  - Frequent: Employee turnover rate is high for IT companies, about 15%~25%.
  - Severe: The cost of hiring a new employee is about 25%~100% of annual salary of an employee.

- The key difficulty is that human resources cannot be substitute each other easily.
  - Different persons have different skills and capabilities.
  - Skills and capabilities need learning and practicing to gain.
Employee Turnover Solution Simulation

- Our method for human resources substitution
  - building a fine-grained human resource productivity model, considering learning and forgetting, which is the natural phenomenon for human being.
Human resource model:
- building fine-grained human resource productivity model, considering learning and forgetting, which is the natural phenomenon for human being.
Learning Model

- Scenario 1: a person carries out a kind of task for the first time. He/She will need some time to learn the skill for the task.

\[ P_{new}(t) = P_{max} \left( \frac{1}{1 + b \cdot e^{-a \cdot t}} \right), a > 0, b > 1 \]
Learning Model

- **Scenario 2:** a person carries out a kind of task again. He/She will need some time to review the skill for the task.

\[ P_{\text{experienced}} = P_{\text{startup}} + (P_{\text{max}} - P_{\text{startup}})(1 - e^{-d \cdot t}), \ a > 0 \]
Forgetting Model

- Forgetting is mainly related to two factors: productivity before forgetting, and the interval time.

\[ P_{\text{forgetting}} = F(P_{\text{beforeforgetting}}, T_{\text{break}}) \]

- We choose a forgetting model as follows:

\[ P_{\text{forgetting}} = \frac{P_{\text{beforeforgetting}}^{c_1}}{C_0 T_{\text{break}}^{c_2}} \]
Human Resource Productivity Model

Based on the learning and forgetting model, we get the productivity model:

\[
Productivity(t) = \begin{cases} 
\frac{P_{new}(t)}{C_0(t - TF_1)^{C_2}} & TS_1 < t \leq TF_1 \\
\frac{P_{new}(TF_1 + 1)^{C_1}}{C_0(t - TF_1)^{C_2}} & TF_1 < t \leq TS_2 \\
Productivity(TS_2) + (P_{max} - Productivity(TS_2))(1 - e^{-d(TS_2 - t)}) & TS_2 < t \leq TF_2 \\
\frac{(Productivity(TS_2) + (P_{max} - Productivity(TS_2))(1 - e^{-d(TF_2 + 1 - TS_2)}))^{C_1}}{C_0(t - TF_2)^{C_2}} & TF_2 < t \leq TS_3 \\
\vdots & \\
(Productivity(TS_{N-1}) + (P_{max} - Productivity(TS_{N-1}))(1 - e^{-d(TF_{N-1} + 1 - TS_{N-1})}))^{C_1} & TF_{N-1} < t \leq TS_N \\
\frac{Productivity(TS_N) + (P_{max} - Productivity(TS_N))(1 - e^{-d(TS_N - t)})}{C_0(t - TF_{N-1})^{C_2}} & TS_N < t \leq TF_N \\
(Productivity(TS_N) + (P_{max} - Productivity(TS_N))(1 - e^{-d(TF_N + 1 - TS_N)}))^C_1 & t > TF_N \\
\end{cases}
\]
Human Resource Productivity Model

- An example: a developer's coding productivity model:
Solution Model Set

- There are three solution models in our framework:
  - Human resources substitution: find a person to substitute the left employee
  - Tasks cancellation: cancel the tasks of the left employee
  - Human resources reassignment: cancel some unimportant tasks, and then find a person to substitute of the left employee

- Solution model set is open and extensible
Agenda

- Overview of the Research Group
- Research Background
- Research Topics and Framework
- Research Progress
- Publications
Publications

2007 (11)


Publications

2007 (11)

2008 (15)
Publications

2008 (15)

- Dapeng Liu, Qing Wang, Junchao Xiao, Juan Li, Huaizhang Li, RVSim: A Simulation Approach to Predict the Impact of Requirements Volatility on Software Project Plans, in Making Globally Distributed Software Development a Success Story. 2008. p. 307-319.


- Jian Zhai, Qiusong Yang, Ye Yang, Junchao Xiao, Qing Wang, Mingshu Li, Automated Process Quality Assurance for Distributed Software Development, in the Second International Conference on Software Engineering Approaches For Offshore and Outsourced Development (SEAFOOD). 2008: Zurich, Switzerland

Publications

2008 (15)


Publications

2009 (6)


- Junchao Xiao, Qing Wang, Mingshu Li, Qiusong Yang, Lizi Xie, Dapeng Liu. Value-based Multiple Software Projects Scheduling with Genetic Algorithm. International Conference on Software Process 2009 (ICSP2009). Vancouver, Canada, pp. 50-62


Thanks!

Q&A