Facilitating Cooperation in Global Software Design via Micro-Estimation

Bin Xu¹, Hua Hu¹, Yun Ling¹, Xiaohu Yang², Zhijun He², Albert Ma³
¹ College of Computer Science, Zhejiang Gongshang University, Hangzhou, China
{xubin, huhua, yling}@mail.zjgsu.edu.cn
² College of Computer Science & Technology, Zhejiang University, Hangzhou, China
{yangxh, hezj}@zju.edu.cn
³ State Street Corporation, Boston MA 02111, USA
amma@statestreet.com

Abstract

Rescheduling design tasks is essential to reduce the impact of communication delay in global cooperative software design. However, it is difficult due to the undetermined duration, delay and task sequence. Micro-estimation on demand refines the estimation of effort and duration for the tasks in the next short period. When using Multiple Component Status Transition Graph (MCSTG) together with Micro-estimation, we may easily obtain the possibility of a task and the available time for this task to be finished. With the extended MCSTG, the designers may reschedule the tasks according to the critical and importance so as to facilitate the cooperation.

Keywords: Task Rescheduling, Global Cooperative Software Design, Multiple Component Status Transition Graph, Communication Delay, Micro-Estimation.

1. Introduction

Software outsourcing is prevalent nowadays; however communication and coordination become the bottleneck in distributed software development. Some survey showed that the management cost sacrificed lots wage saving. Besides, some previous research [1, 2] showed that distributed work items appear to take about two and one-half times longer to complete as similar items where all the work is collocated together. Such delay appears to be owing to communication and coordination issues instead of the size or complexity of cross-site work.

Different tactics have been suggested to reduce the cultural impact in global cooperation model [3, 4] and the methodology of virtual team management in global development has also been introduced in [6, 9]. Some practices to reduce the coupling of distributed work are suggested by [3, 7, and 10], including “Developing different subsystems at different sites” and “Executing different process steps at different sites”.

Typically, the design of a large complex system in global project bridges the requirements or specification to the development. The design activities should be finished by the cooperation of different people with different skill sets. As a result, design is formidable due to frequent and large amount of communication and coordination in a global development environment.

While Communication Queue [8] may be used to minimize the impact of communication delay, task dependence including data and control dependence should be well handled so as to avoid the conflicts incurred. On the basis of analysis of task dependence between components, Multiple Component Status Transition Graph is suggested so as to not only describe intra component dependency, but also describe the inter-dependency between components [11].

Since the design activity is complex, the former design activities will be revisited if there are some defects or requirement changes. As a result, the designers are not sure which will be the following tasks when they get a certain task. Besides, the duration of each design task is not determined, so it is hard to reschedule design tasks in global cooperative environment. Therefore, duration estimation of current task and estimation of possible following task is essential for the better design tasks rescheduling.

In this paper, Micro-Estimation is proposed, which is used to handle the loop in the task network and make refined estimations so as to facilitate the cooperation in global software design.

The rest of this paper is organized as following. The problem in graph based coordination due to the contained loops is identified in section 2; Micro-Estimation technique is proposed in section 3. Section 4 states how to expand Fast Automatic Notification sub-system with Micro-Estimation, and section 5 suggests a way to integrate Communication Queues sub-system with expanded Fast Automatic Notification sub-system. The summary and discussion is given in section 6, and we also introduce the currently status of the research there.

2. Graph based coordination

Coordination based on Multiple Components Status Transition Graph (MCSTG) has been introduced by the
authors, the entry and exit criteria are studied in coordination [Figure 1] [11] as well as the effort pressure controlling in [12].

It will be very helpful for the project managers to master the timelines of every task, including when a task should be finished, the available time for a certain task to be started, and resource need in the near future.

![Figure 1. Component Status Transition Graph [11]](image1)

Regarding the time management, critical path arithmetic has already been introduced to calculate the timelines of a project including the ES (Early Start Time), EF (Early Finish Time), LS (Last Start Time), and LF (Last Finish Time) for each task. Giving the example in Figure 2, we may easily know that all the tasks can be finished in 71 days, and task D can be started between 7th day and 15th day with the involved formulas, ES = \[
\text{max}(\text{EF (Preceding tasks)}) \] and EF = ES + Duration.

![Figure 2. AOA task network without LOOP](image2)

When the duration is not determined, the PERT techniques can be used to get the most possible timeline for the tasks and project. However, when considering the bug fixing, requirement change and involving development, there may be some loops in the network. The formulas above are no longer useful when there are some loops in the task network [Figure 3]. Though the network is very simple, however, the Early Start Time for Task D couldn’t be got at the very beginning because the loops are not certain and the duration for task A, B, and C is not fixed in each loop.

![Figure 3. A simple task network with loop](image3)

3. Micro-Estimation

The timeline calculation is still a problem in a real project when there are some loops in the task network before we know the possibilities of the branches, the times when the loop occurs and the duration of the tasks in the loop.

With the historical project data, the duration of the tasks can be estimated in statistics. However, it is insufficient for the decision making towards better coordination in global cooperative software design. Micro-Estimation is proposed to solve such problem. When performing Micro-Estimation, the scale of task is very small as it estimates the timeline of each piece of tasks. Besides, the interval of time is short as it only considers the current available tasks and ignores the following further steps.

3.1 Translation of MCSTG into AON network

Giving the example in Figure 1, assume that there are some relationship between CSTG of α and β [Figure 4].

![Figure 4. Relationship between different CSTG](image4)

All these dependence may be translated into four cases, FS (task A should be finished before the start of task B), FF (task A should be finished before the finish of task B), SF (task A should be started before the finish of task B) and SS (task A should be started before the start of task B). MCSTG may then be translated into Arrow on Node (AON) task network as follows [Figure 5]. The
dependence of FS can be ignored as it is the most possible case.

Figure 5. Part of AON task network

3.2 Possibility estimation for the branches

If there is a loop in the task network, there must be at least one of the nodes having several branches after it [Figure 6]. Assume that we have generated some of the historical project data and wrapped them into (Component, Task, Duration, Time). At first, all the structured data are sorted according to the attributes of Component and Time. The possibility of each branch can be calculated from the number of the tasks happened in the history. For instance, regarding the possibility of S6 to S5, we only need to count all the tasks after S6, named Post (S6), and among which how many of them are S5, named Move (S6, S5). The possibility is the division of Move (S6, S5) of Post (S6).

In practice, the possibility of each branch varies from different kind of components and the complexity. However, to be simple, the average possibility of the branch may be used in the estimation.

Figure 6. Possibility of branches

3.3 Duration estimation

Duration estimation is based on the assumption that the duration of adjacent tasks are in the linear relationship. Regarding the example shown in Figure 6, \( P(M,K) \) can be calculated from the historical data:

\[
P(M,K) = \frac{\sum(Duration(Task \ M))}{\sum(Duration(Task \ K))} \quad (1)
\]

If the duration of task M is known as \( Duration(M) \), the duration of task K can then be estimated as:

\[
Duration(Task \ K) = Duration(M) \times P(M,K)
\]

Because the duration of each task is very important in coordination, PERT technique is used to improve the precision of estimation. When taking task M and task L in figure 6 as the examples, a table may be generated indicating the possible duration for task L from the historical project data [Table 1].

Table 1. Ratio indicating of conjoint tasks

<table>
<thead>
<tr>
<th>Task M</th>
<th>Task L</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>0.50</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1.33</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>1.29</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>0.67</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Regarding each pair of conjoint tasks, such as Task M and Task L in the example, the minimal, average, and maximal ratio will be calculated from table 1, and assigned to \( T_o \), \( T_m \) and \( T_p \) indicating the optimistic, most possible and pessimistic duration.

\[
T_o = \text{minimal (ratio (T, L))} \ ,
\]

\[
T_p = \text{maximal (ratio (T, L))} \ , \text{ and}
\]

\[
T_m = P(T, L).
\]

Figure 7. Duration estimation using PERT

Typically the expected duration will be,

\[
T_e = \frac{(T_o + 4T_m + T_p)}{6}.
\] (3)

Since the design tasks are of different critical and importance, the expected duration will be large for the large scale and complex design task and the critical or important tasks will also be assigned with more time. As a
result, the software engineers and architects may adjust the duration estimated according to factors. That is to say, if the task is of large scale, complex, critical, or importance, the time will be longer so as to increase the possibility of success. While on the other hand, if the task is of small scale, simple, and of low importance, the time can be less.

When the duration for current tasks are adjusted, duration of the subsequent tasks may then be estimated accordingly. The possible work can then be clarified due to the known possibility of each branch and the duration for each task according to formulas 1, 2 and 3:

\[
\text{Effort} = \sum \text{possibility} \times \text{Estimated Duration}
\]

Each task will be specified with “required expertise”, and “delay time”. Therefore, the human resource usage and knowledge/expertise requirement may be generated according to the effort estimation. Besides, the possible communication delay time will be notified and other design may shift to other tasks during the period.

3.4 Critical path analysis

In order to shorten the entire life cycle, tasks are rescheduled according to their emergency. The tasks on the critical path will be placed in the high priority. The critical path analysis technique has been introduced in section 2 [Figure 2], and many project management tools, such as MS project, can recognize the critical path in the task network when there is no loop inside. Other tasks will be scheduled according to last finish time which will not impact the conjoint tasks. The different of time zone between distributed sites should be take care when calculate the available time.

When someone receives the task, the available time will be noticed so that he/she can take care of when the other will need the result and schedule the tasks efficiently.

4. Micro-Estimation expanded FAN

FAN (Fast Automatic Notification) has been used in the design phase of reengineering one module in Lattice system [11]. In figure 8, it has been expanded with Micro-Estimation. In such way, the estimation of duration and resource message can also be generated and sent to the roles. In order to reduce the message subscription effort, the roles only need to declare their expertise and their requirement of monitoring. Message center is built to handle these requirements, filter the message and send the message to a certain group of roles.

The software engineers or architects act as roles in the system. When they finish a task, they will record it in the system, which will be recognized as an event to the system. When an event is input, scenario will be generated, and MCSTG will be checked if the exit criterion of the node is satisfied and then find the nodes whose entry criteria are satisfied.

In the former research [11], an automatic notification mechanism was established incorporating with MCSTG, which accepted the event and response with suggestions and warning messages. Dynamic status message subscription and notification were then used to notify different roles with different messages. However, it may be boring to subscription every small piece of task. In expanded FAN, an expertise based subscription is adopted to reduce the effort in task subscription.

With the Micro-Estimation expand mechanism, the possible tasks with estimated duration and resource will also been generated and sent to the message center. Message center will transfer the task descriptions to the roles with specified expertise or have the monitor requirement.

5. Integration of CQ and ME-FAN

Micro-Estimation expanded FAN (ME-FAN) can be used together with Communication Queue [Figure 9], the roles input events (record the finished activities) into ME-FAN, declare the expertise and require the information from ME-FAN. ME-FAN will determine if the task can be finished according to the exit condition, and generate all the Ready Tasks whose entry criteria is satisfied. Besides, the duration and resource of the possible conjoint tasks will be estimated. All these messages will be sent to Communication Queues.
the tasks will be assigned to the roles according to their expertise.

![Diagram: Task Arrangement](image)

**Figure 9. Using CQ with ME-FAN**

It’s a future integration plan for Communication Queues sub-system and Micro-Estimation expanded FAN. However, the Communication Queue hasn’t been expanded so as to handle the estimation data and it will just ignore such information currently. In order to exploit the Micro-Estimation technique completely, Communication Queues sub-system will be expanded and handle such estimation data so as to arrange the task efficiently and intelligently.

### 6. Summary and Discussion

Graph based coordination is introduced and the problem due to the loops in task network is identified. Micro-Estimation technique is proposed to estimate the conjoint tasks so as to facilitate the cooperation. Multiple Component Status Transition Graph is expanded with Micro-Estimation, and so as the FAN sub-system.

When there are some loops in the task network or the duration of the tasks is not determined, typical critical path analysis techniques fails in finding the critical path. Micro-Estimation can be used together with critical path analysis techniques so as to find the critical tasks on demand.

Currently, ME-FAN sub-system is being tested and improved in a controlled environment. The estimation data will be rushed to the roles according to the declared expertise. The roles may view the estimation data, which will improve their decision making. Communication Queue hasn’t been expanded due to the large effort in intelligent task arrangement. As a result, the entire system couldn’t be used in practice currently.

The future research of the authors will focus on how to realize the entire computer supported cooperative design platform with Micro-Estimation. Some techniques of statistics will be used when expanding the Communication Queues sub-system.

Besides, the authors are planning to verify the Micro-Estimation method mathematically in the near future, so as to expand the research result to other domains.

### Acknowledgement

This work is part of Lattice® Reengineering project which is an attempt to reengineer a large complex legacy system. Lattice Reengineering is funded by State Street Corporation, USA. The project is collaboration between Zhejiang University, China, Zhejiang Gongshang University and State Street Corporation, USA. Lattice is a registered trademark of State Street Corporation, USA.

The contents of this paper are the opinions and conclusions of the author only and do not necessarily represent the position of State Street Corporation or its subsidiaries, affiliates, officers, directors or employees.

### References


