# The Role of Software Models in Developing New Software Systems; A Case Study in Project Management

**Koichiro Ochimizu**
Vice President of
Japan Advanced Institute of Science and Technology (JAIST)
Research Professor and Director of
Center for Highly Dependable Embedded Systems Technology, JAIST

1-1 Asahidai, Nomi, Ishikawa, Japan
ochimizu@jaist.ac.jp

## Abstract

The collaboration process in a joint project with a Japanese company is introduced. Topics included are the role of software models at each phase of the joint project and the skill set model of the collaboration.

We have developed a software project scheduling tool suited to the Japanese work environment.

Through the joint project, models play a key role as an effective communication media between Academia and Industry. They are useful for:

- Problem-discovery and problem-setting;
- Abstraction of the real world to extract necessary information, organizing this data into a useful information structure;
- Construction of a mathematical model to assure the generalizability of the theory and the tool;
- Visualization of the organizational resource usage state and definitions of several metrics.

*Keywords*: Joint Project with Company; Project Scheduling Tool; Load-Capacity Model; Resource Binding; Integer Linear Programming (ILP).

## 1 Introduction

Many activities for incremental technological innovations are now performed in Japanese companies. For the success of a new technological innovation, collaboration between industry and academia is very important. In this paper, I will address the role of models as communication media between academia and industrial people in the joint project mentioned below.

### 1.1 Technical Results

Embedded software for digital electrical appliances development projects in recent years are planned based on appropriate release time.



However, most projects require long-term development. Therefore, two or more projects are implemented concurrently, as shown in Fig. 1.

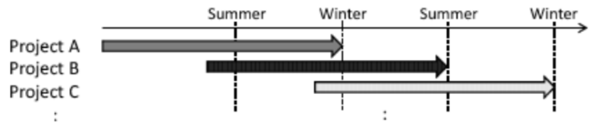



Fig.1 Multiple Projects Problem

To evaluate the feasibility of a new software development project in a situation in which two or more projects are being implemented concurrently, it is necessary to consider not only software development workload but also residual capacity of an organization.

We have developed the Load-Capacity Model for verifying feasibility at an early stage of the software development project. We have developed a method for simultaneous project scheduling, and a resource-binding based on the Load-Capacity Model. This method is formulated as Integer Linear Programming, using Load-graph and Capacity-graph, which represent workload and organizational capacity, respectively, as inputs. Outputs of this method are Gantt Charts and Load-Capacity Diagrams. The load-capacity diagram shows the state of an organization after workload has been assigned. Experimental results on realistic examples show that the method provided useful solutions in a practical time period. (Saito, Kusanagi, Ochimizu 2011), (Kusanagi, Saito, Ochimizu 2011), (Saito, Kusanagi, Ochimizu 2012).

### 1.2 Skill Set Model of Collaboration

One effective method to promote research on software engineering is joint work with industry, which enables us to recognize and solve problems in the real world. Through several case studies we performed, including this case, four types of key persons are needed for the success of collaborations.

(1) Key person 1 (from industry)

A person who is highly motivated for problem-discovery and problem-solving, and knows the real world very well.

(2) Key person 2 (from academia)

A person who has expertise in software engineering, and who knows the various kinds of research outcomes in software engineering. Key person 2 should try to discover the real problem together with key person 1 from superficial problem statements. Key person 2 is also responsible for proposing several candidates for a solution, based on his/her knowledge and experience. Key person 1 should be able to evaluate the effectiveness of the proposed solutions.

(3) Key person 3 (expert on mathematics)

A person who can propose and formulate mathematical models for the solution. Mathematical model is important to assure generalizability of the problem and the solution.

(4) Key person 4

A person who can develop a prototype tool based on the conceptual models and the mathematical models adopted. A tool is very important for field testing and then for familiarizing the new result in industry.

(5) Key person 1 again

Key person 1 should have authority to perform several field tests using a tool, with the permission and help of people in the field.

## 2 Problem-Discovery and Problem-Setting

Both key person 1 and key person 2 should discover a real problem to be solved, from among the superficial problem statements in the field.

### 2.1 Original Proposal from Industry

Fig.2 shows the original proposal from key person 1 which represents the structure of the organization with problem statements.

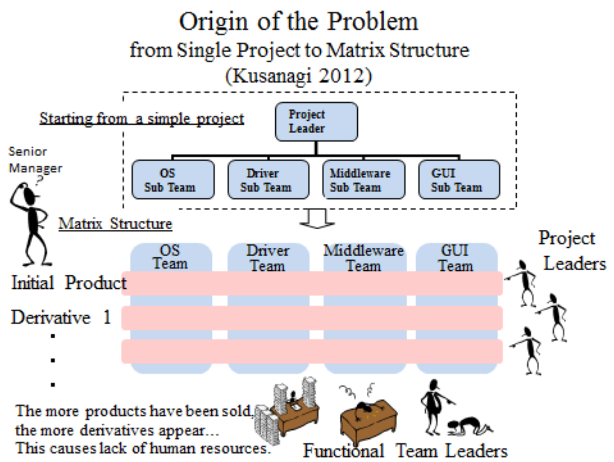



Fig. 2 From Single Project to Matrix Structure

New product development starts from a simple project, as shown at the top of Fig. 2. The more products have been sold, the more derivatives appear. And a new project is initiated for each derivative. So, the structure of the organization changes from the single project structure to the matrix structure, as shown at the bottom of Fig.2.

There are several functional teams which supply experts in operating systems, drivers, middle-ware and GUI to several projects. When we start to develop a new product, a project leader needs to manage a simple project. In a matrix structure, each project needs its own project leader.

A matrix structure is just a direct abstraction of an organizational structure, and it is an informal model. The informal model is useful for problem-discovery if it is a good abstraction and reflection of the real world. We could observe and discuss several problematic situations based on this structure.

Time must be spent at this stage because problem-discovery and problem-setting are the most important tasks for the success. We must consider what the problem is very carefully. There are several candidates as the problem. The following information shown in Table 1 was given from key person 1 to key person 2, to share knowledge of the current status related to concurrent execution of multiple projects . There are three types of stakeholders. Those are a senior manager, a project leader, and a functional team leader.

Table 1 Role and Problem Statement of each Stakeholder (Kusanagi 2012)

| | Role | Problems |
|---|---|---|
| **Senior Manager** | - should manage all of the products<br>- should grasp situations both of projects and functional teams to release all of the derivatives by due date | - hard to understand excess or shortage of resources of his organization<br>- hard to adjust resource allocation among multiple projects |
| **Project Leader** | - should achieve QCD goals of his/her project | - worry about whether each functional team can supply sufficient human resources to his project to meet his QCD goals |
| **Functional Team Leader** | - is responsible to supply proper human resources to multiple projects simultaneously<br>- should allow proper time for team members to improve their skills<br>- should accumulate know-how | - hard to decide whether he can accept workload requested by a new project<br>- education and training of his team members |

There are several superficial problems; process-related ones, product-related ones, and human-resource-related ones. In our collaboration, we agreed the human-resource-related problem is the most influential one.

### 2.2 Our Solution

Through discussion, we reached the following conclusion, as shown in Fig.3.

*A project scheduling method should take into account both work load of a project and the residual capacity of an organization.*

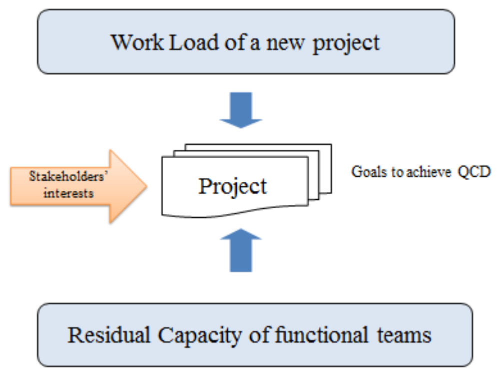



Fig.3 Our approach to the solution

Now we need to develop a theory and models which can deal with resource management of functional teams for a given workload. We proposed the Load-Capacity Model, which contains necessary information to evaluate feasibility of a new project.

## 2.3 Abstraction of the real world to extract necessary information and to organize this data into a useful information structure

Next we tried to extract necessary information for both work load and capacity. Fig.4 shows the extracted information.

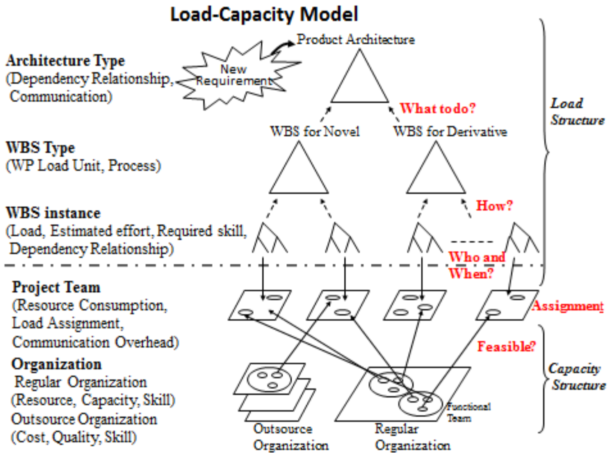



Fig. 4 Load-Capacity Model (Kusanagi 2012)

The Load-Capacity Model shown in Fig.4 consists of five components: Architecture Type, WBS Type, WBS Instance, Project Team and Organization (Regular Organization and Outsource Organization). Architecture Type, WBS Type and WBS Instance are basic components of the load structure. Regular Organization and Outsource Organization are basic components of the capacity structure. Teams of a new project need to satisfy the requirements of both the load structure and the capacity structure (Saito, Kusanagi, and Ochimizu 2011, 2012).

*1) Architecture Type:* “Architecture type” means the structural aspects of a system architecture, typically it consists of GUI, application, middleware, driver, and operating system. The attributes of an architecture type are dependency relations and communications. In general, functional teams are organized according to the architecture of the system. Relationships among components of the architecture require communications between the developers. These communications are one of the load factors.

*2) WBS Type:* WBS (Work Breakdown Structure) type is a set of basic work units, named Work Packages (WPs) usually organized as a tree structure. There are two WBS types, WBS type for new developments and WBS type for derivative developments. WBS types are usually defined by the Software Engineering Process Group (SEPG) of each organization. The attributes of a WBS type are the load unit of each WP (WP load unit) and the process. WP load unit of each WP becomes the standard of the estimation. The process is a set of one or more WPs, and WBS type includes processes as subsets.

*3) WBS Instance:* The WBS instance is a set of WPs instantiated and tailored from the WBS type, and it corresponds to all WPs that are actually executed by the project. The attributes of a WBS instance are the total load, the standard estimated man-hours (SEH), the required skills, and the precedence constraints. In the Load-Capacity Model, the total load is the sum of the SEH of each WP and the communication overhead. However, the communication overhead does not appear easily in the WBS instance. SEH is derived from WP load unit. There are precedence constraints between WPs, which are derived from the dependency relations and the processes.

The skill given to each WP as an attribute is called the required skill, and a skill of a resource is called “own” skill. We use $\theta$ as the symbol of skill. Examples of skills are business knowledge in requirement definition, design methodology in system design, and programming language in software implementation. “skill” in this model is different from ability. In our model, we temporarily deal with skill as a classification of skill. We can assign WP to the resource if the required skill corresponds to “own” skill.

*4) Project Team:* A project team is a set of people involved in product development. The attributes of a project team are the consumption of resources, the distribution of the load, and the communication overhead. Resource consumption is the assignment of the load (WP) to the resources based on the relation between the delivery date and the skill.

*5) Organization:* The organization is a set of developers.

The organization consists of a regular organization and outsource organizations. Each organization has two or more subsets, and we call each subset a functional team. An organization offers members of functional teams to each project as resources to accomplish the project. The attributes of a regular organization are resources, capacity and skill. The attributes of an outsource organization are cost, quality and skill. In the Load-Capacity Model, the resources are classified according to the number of skills. In our model, a resource which has two or more skills is defined as a highly-skilled resource.

## 3 Formulation of a mathematical model to assure the generalizability of the theory and the tool

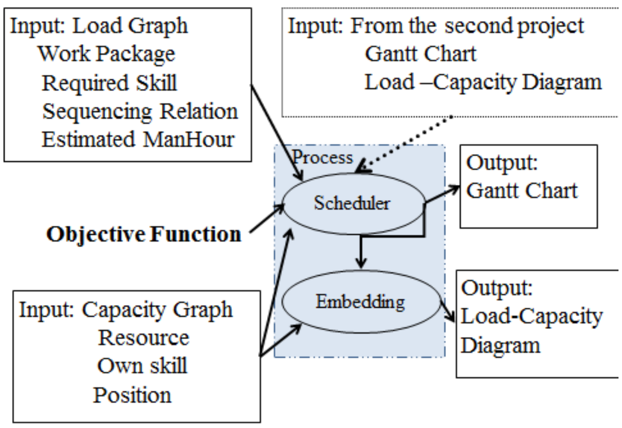



Fig.5 Mathematical Formulation (Saito 2012)

Mathematical model of the solution is necessary to assure the generalizability of the problem and the solution.

We formulated our resource-binding method as Integer Linear Programming. Fig. 5 shows an outline of our model.

## 3.1 Load Graph

There two inputs of our algorithm. One is the Load Graph, which is a graph representation of the upper part of Fig. 4.

A WBS instance showing all precedence constraints is called a load-graph. An example is shown in Fig. 6 (Saito, Kusanagi, and Ochimizu 2011).

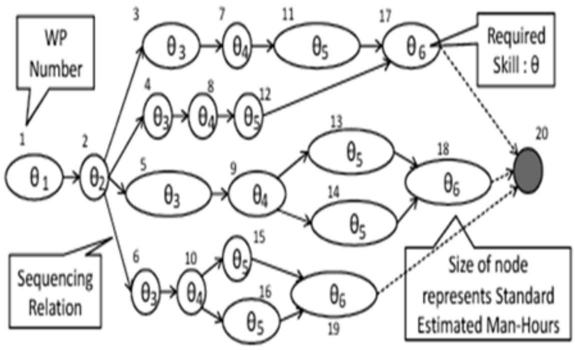
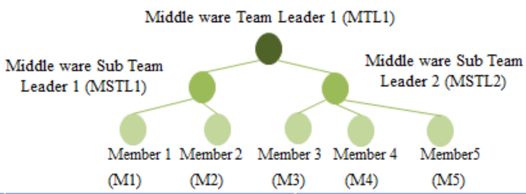



Fig. 6 An example of a Load Graph

A Load Graph is a directed acyclic graph. Each vertex represents a WP, and the directed edge represents the sequencing relation between WPs. It is assumed that SEH (number of time slots) and a required skill are given to each WP as attributes.

## 3.2 Capacity Graph

Another input is a Capacity Graph, which is a graph representation of the lower part of Fig.4. The capacity-graph is derived from the capacity structure of the Load-Capacity Model. Figure 7 shows an example of a capacity-graph (Saito, Kusanagi, and Ochimizu 2011).




| | Network Middleware | | | | | | DB Middleware | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Diff of Reqt. Spec | Plann-ing | Detailed Design | Peer Review | Programming and Unit Test | Inte-gration Test | Diff of Reqt. Spec | Plann-ing | Detailed Design | Peer Review | Programming and Unit Test | Inte-gration Test |
| MTL1 | o | o | o | o | o | o | | | | o | | |
| MSTL1 | o | o | o | o | o | o | | | | | | |
| M1-M2 | | | | o | o | o | | | | | | |
| MSTL2 | | | | | | | o | o | o | o | o | o |
| M3-M5 | | | | | | | | | | o | o | o |

Fig. 7 An example of a Capacity Graph

The capacity-graph shows the hierarchical structure of the organization. In a capacity graph, a vertex represents a resource (human) and a role, and edges represent relations in the organization. A vertex (resource) in the capacity-graph has one or more "own" skills. The residual capacity of resource $r$ at time $l$ is shown by a capacity function $f(r, l) \in \{0, 1\}$ where value 1 means WP is assigned. This team's "own" skill is shown in the table in Fig.7 .

## 3.3 Load-Capacity Diagram

The load-capacity diagram shows the state of an organization after workload has been assigned. The idea is to embed the load-graph into the capacity-graph satisfying the constraints (Saito, Kusanagi, and Ochimizu 2011).

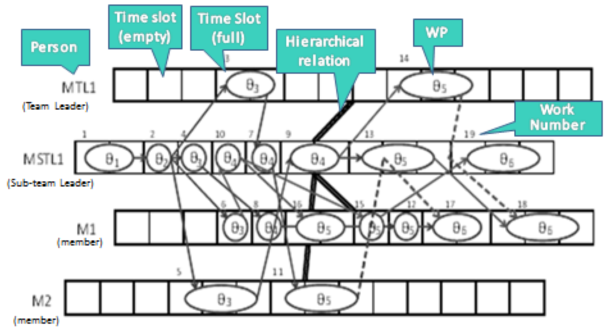



Fig. 8 Load-Capacity Diagram
(ex. One Team of four persons)

Each row of the Load-Capacity Graph corresponds to a member of a project, representing who has time resources, represented by a sequence of time slots. In this example, we assume a project consists of only middleware team members.

## 3.4 Gantt chart

Gantt chart is added as one of the outputs, because key person 1 advised us that it is necessary to prepare user-familiar output. Fig. 9 shows an example of a Gantt chart produced by our system.

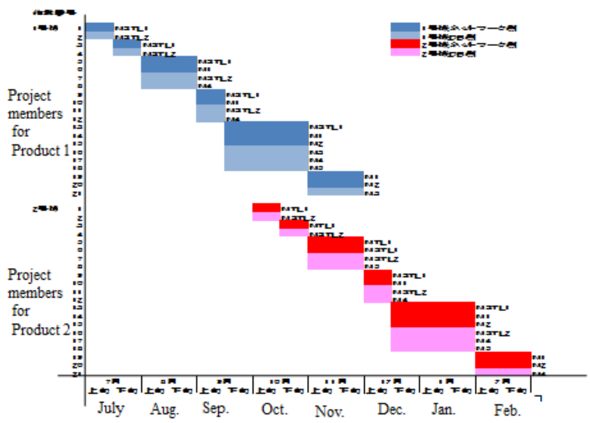



Fig. 9 Gantt Chart (Kusanagi 2012)

## 3.5 Resource-Binding algorithm

We adopted Integer Linear Programming as a resource binding algorithm. The reasons why we adopted the ILP are mentioned below.

(1) Various conditions of the project can be formulated as the constraints of ILP

(2) Two or more schedules can easily be generated by changing the objective function and the constraints

(3) The minimum resources or the minimum development period are obtained by an optimal solution under the given constraints

(4) The solution is obtained faster than in the past, by improvement of the performance of the ILP-solver and calculator

Table 2 shows examples of constraint representation, and Table 3 shows an example of an objective function which represents the resource-allocation policy.

Table 2 Decision Variables and Constraints (Saito 2011)

- Binary decision variable: $z_{irl} \in \{0,1\}$
  - Only when $WP_i$ is bound to resource $r$ and starts in slot $l$
- Constraints:
  1. Each WP must be implemented at least one time
     $$\sum_{r:I(i)}\sum_{l} z_{irl} = 1$$
  2. The sequencing relation must be satisfied
     $$\sum_{r:I(i)}\sum_{l} l\cdot z_{irl} \ge \sum_{r:I(j)}\sum_{l} l\cdot z_{jrl} + h_j$$
  3. WPs bound to the same resource must not be concurrent
     $$\sum_{i:B(r)}\sum_{m=\alpha}^{l} z_{irm} \le 1$$
  4. Remaining capacity of each resource must be considered
     $$\sum_{m=\alpha}^{l} z_{irm} \le \prod_{m=\alpha}^{l} f(r,m)$$

Table 3 Objective Function (Saito 2011)

- We can consider various objective functions, which express the policies adopted in the project
- Example: Minimize the development period and the number of WPs to be assigned to Team Leader and Member2

$$\min \sum_{l=1}^{L+1} l\cdot\left(\sum_{i:B(TeamLeader)} z_{iMTLl} + \sum_{i:B(Member2)} z_{iM2l}\right)$$

## 4 Visualization of resource allocation state by " pieces of ice in a cup" metaphor

We adopted " pieces of ice in a cup" metaphor to visualize the resource-binding result, and to develop metrics.

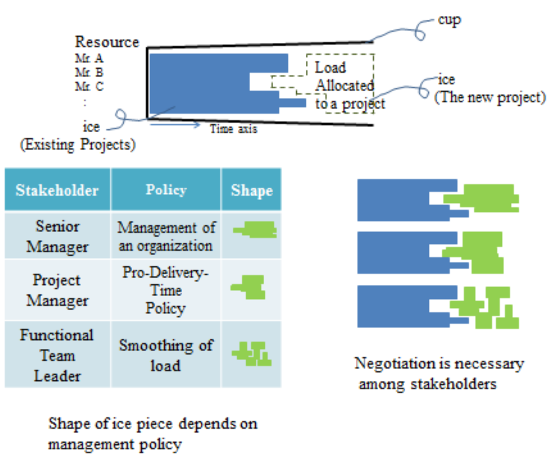



Fig.10 "ice pieces in a cup" metaphor

If we compare the capacity of an organization to a cup , and the load of a project to a piece of ice, scheduling can be compared to putting several pieces of ice into the cup. The shape of a piece of ice depends on the project management policy ( Fig. 10).

Fig.11, Fig.12, and Fig.13 show examples of the scheduling result of a third project, based on different work-load-allocation policies. All of the results used real data collected in field testing.

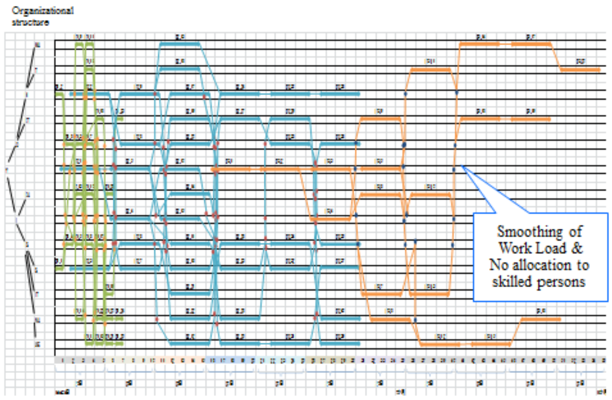



Fig.11 "Smoothing of work load" policy (Kusanagi 2012)

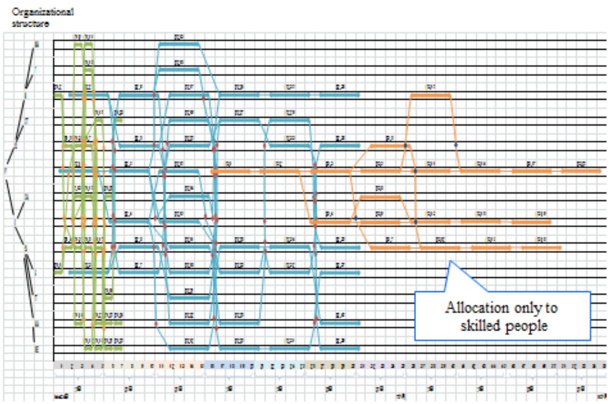



Fig.12 "Use only skilled people" policy (Kusanagi 2012)

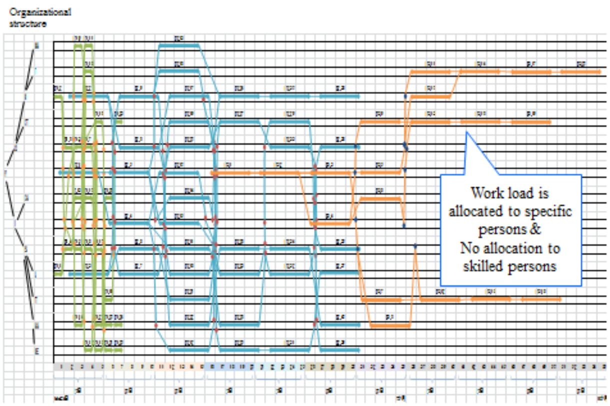



Fig.13 "Uneven distribution of work load" policy (Kusanagi 2012)

## 5 Developing the Metrics

Fig. 14 helps to visualize the resource allocation. We can define several metrics based on the presentation of

Fig.14. Table 4 shows the requirement for the metrics and the metrics defined.

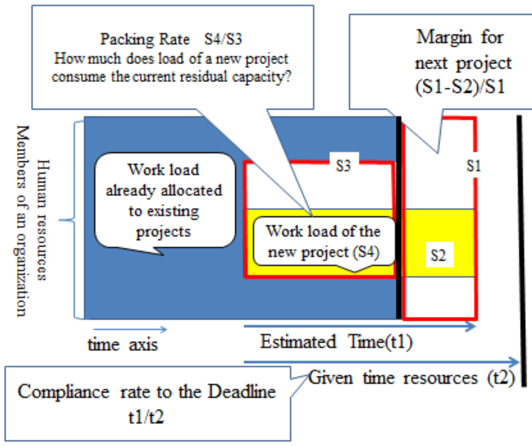



Fig.14 Metrics

Table 4 Usage of Metrics (Kusanagi 2012)

| | Role | Responsibility | Metrics |
|---|---|---|---|
| Senior Manager | Manages both multiple projects and all of the Products | Judge feasibility of all projects. Understand the status of resources of each functional team | All of the following three metrics |
| Project Manager (Leader) | Achieve QCD goals for a specific project | Judge feasibility of his/her project at the early stage of project planning. | Compliance rate to the Deadline = Estimated Time/ Given time resources |
| Functional Team Leader | - is responsible to supply proper human resources for multiple projects simultaneously<br>- Should allow proper time for team members to improve skills | - Balancing of several policies<br>- Judge the feasibility of a new project from the viewpoint of resource supplier<br>- Prepare for the next project | - Packing Rate = Load of the new project at the present period/ Residual capacity at the present period<br>- Margin for next project = residual capacity at the extended period / total organizational capacity at the extended period |

## 6 Prototype Development

We developed a prototype tool, named Load-Capacity Balancer, based on the conceptual models and mathematical models, to publicize the new results.

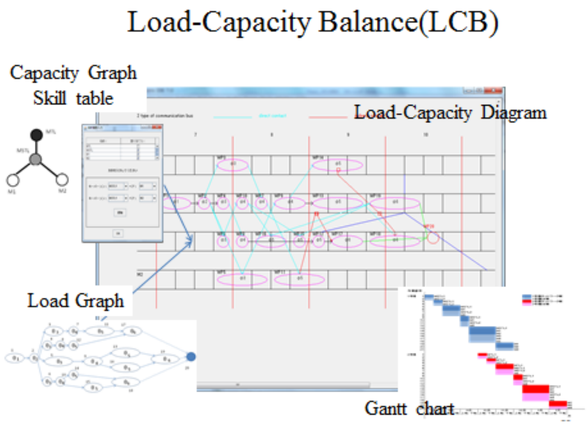



Fig. 13 Load-Capacity Balancer (Saito 2012)

We have performed a computer experiment on some input classes to evaluate execution time of the proposed ILP-based method. The results show that the method provided solutions in a practical time period for combination of both realistic size load-graphs and capacity-graphs.

## 7 Conclusion

We created many informal diagrams through the joint project. All of them played an important role at each phase of the collaboration.

(1) The abstract depiction is a trigger of discussion, and was useful for sharing knowledge about situations in the field.

(2) The abstract depiction is also useful for extracting the necessary information from the field.

(3) Mathematical model is necessary to develop the tool

(4) Visualization of computing results is also necessary for practical usage of the method and the tool in the field.

Without the informal models created through our collaboration, we could not communicate with each other to achieve valuable consensus and agreement.

## 8 Acknowledgement

I will express my best thanks to my partners in this joint project, Dr. Takumi Kusanagi who works for Toshiba played a very important role in this joint project as key person 1. Dr. Akinori Saito, who was a PhD candidate during this joint project, and played a very important role too as key person 3 and key person 4.

## 9 References

A. Saito, T. Kusanagi, and K. Ochimizu (2011): A Simultaneous Scheduling and Resource Binding Method Based on the Load-Capacity Model. *Proc. of APSEC2011.*

T. Kusanagi, A. Saito, and K. Ochimizu (2011): The proposal of the Feasibility Metrics of a new Project based on the Load-Capacity Model. *The 6th International Conference on Software Process and Product Measurement (Mensura2011)* .

A. Saito, T. Kusanagi, and K. Ochimizu (2012): A Software Project Scheduling Method Based on the Load-Capacity Model. T*he IEICE TRANSACTION ON INFORMATION AND SYSTEMS (JAPANESE EDITION)*, **J95-D**(2): 225.-236.

A. Saito (2012): A Project Scheduling Method based on the Load-Capacity Model (in Japanese). Ph.D. thesis. Japan Advanced Institute of Science and Technology, Japan.

T. Kusanagi (2012): Proposal and Evaluation of the Method for Feasibility Study of a new Project based on the Load-Capacity Model (in Japanese). Ph.D. thesis. Japan Advanced Institute of Science and Technology, Japan.