Doctoral Thesis

Studies on Motivation and Learning Strategies in Introductory Programming Courses

March 2017

DINH THI DONG PHUONG
Studies on Motivation and Learning Strategies in Introductory Programming Courses
（初等プログラミング・コースにおける動機と学習方略の研究）

March 2017
2017年3月

DINH THI DONG PHUONG
ディン ティ ドン フーオン

Principal referee: Professor SHIMAKAWA Hiromitsu
主査：島川 博光 教授
Studies on Motivation and Learning Strategies in Introductory Programming Courses

DINH THI DONG PHUONG

Abstract

The thesis proposes a successful learning environment for an introductory course based on motivation and learning strategies of students. The environment composes of a discipline to lead the students to study to achieve the course goal, and supporting services responsive to all types of the students to encourage them to study. During the course, there would be students who get behind in class. The formative assessment is conducted to distinguish them from others for teaching staff to take care.

Motivation and learning strategies are internal processes. They represent the programming learning status of a student. Motivation explains the reasons why the student wants to study programming, while learning strategies explain his approach to obtain knowledge and skill in programming. There are various learning statuses in the students. Learning status of a student varies during a course. Instructional Design and Motivational Design propose to respond to all sorts of the learning statuses of the students with affordable teaching staffs to achieve the best learning performance. However, in real training condition, limited numbers of teaching staffs demands the teachers to design the learning environment covering all kinds of learning statuses. The thesis proposes to use a persona to characterize motivation and learning strategies of a group of students having similar motivation and learning strategies. The personas are divided into 2 types: passive and active ones. Students belonging to passive personas need supervisions to overcome difficulties, while students belonging to active personas can study by themselves. The methods proposed in the thesis address the problems, making best use of personas from the viewpoints of design of course plans and formative assessment of students.

Since the course goal and the learning environments have to be built before the course, the thesis proposes PMD (Persona, Motivation, and Discipline) method to design a course plan, utilizing the data of the student of the past course. The data includes the learning behavior which is taken automatically with a learning support system. The data also include the student personas which come from the contextual inquiry of the past students. Because the personas of the students would keep maintained in some successive courses in the same training institute, it is possible for the method to use past student personas to design the course plan.
In the course plan, the discipline incorporates the good learning strategies of the top good students with the affective motivation of the personas. The good learning strategies ensure that they will reach to the course goal, if the students follow the strategies. The affective motivation makes the students work conforming to the strategies. The good learning strategies are analyzed from the learning behaviors, while the affective components are from student personas.

Applying the PMD method to over 500 students supervised by 10 teachers and 100 TAs, we have improved the learning time of the students and succeeded in keeping their scores maintained horizontally in 2 successive courses.

During the course, there would be students who get demotivated because they cannot overcome difficulties. Teachers should detect them to give supervisions to them. Traditional methods use questionnaire or learning behaviors. However, the questionnaire takes time from students. It is difficult for the teachers to understand the student learning correctly from behaviors. To grasp learning status of the students, the thesis proposes to use the non-negative matrix factorization which decomposes the matrix representing the learning behavior into two result matrices: a weight matrix and a gene matrix. The weight matrix represents the motivation and learning strategies of every student, while the gene matrix represents the associations of the motivation and learning strategies with the behavior factors. We utilize the similarity of the current course and the past course under the view point of student personas and course settings to initialize the two result matrices. We applied the method to figure out learning status of students in an introductory C programming course of Ritsumeikan University in 2013. We compare results with learning status obtained by the contextual inquiry conducted after the course. The activeness of individual students can be predicted with more than 70% accuracy in the latter half of the course.

The success in the application of the course plan implies teachers of introductory programming courses can regard many students as few groups to direct their learning to the goal. The preciseness of the formative assessment proves the proposed method can detect passive students during the course, even if active students get demotivated because of learning hurdles. The teachers can overcome the shortage of teaching staffs, assigning teaching staffs only to detected passive students. Without imposing extra effort on the students to understand their learning status, the teachers can achieve an efficient course with the methods for the course plan and the formative assessment, in spite of the limit of teaching staffs.
Contents

1 Introduction
  1.1 Problems of a C programming course .................................. 1
  1.2 Motivation and learning strategies of learners ....................... 6
  1.3 Personas to characterize student groups .............................. 8
  1.4 Inclusive course plan with discipline ................................ 10
  1.5 Formative assessment of personas ................................... 13
  1.6 Approaches and organization of thesis ............................... 16

2 Motivation and learning strategy of student ............................ 19
  2.1 Motivation ................................................................. 19
    2.1.1 Definition ......................................................... 19
    2.1.2 Motivation components ......................................... 20
  2.2 Learning strategies .................................................... 23
  2.3 Obtaining motivation and learning strategies ....................... 26
    2.3.1 Questionnaire .................................................... 27
2.3.2 Contextual inquiry ........................................... 29

3 Course plan with discipline utilizing persona motivation .................. 31

3.1 Introduction .................................................................. 31

3.2 Introductory programming practice course .............................. 33

3.2.1 Motivation and learning strategy ................................... 33

3.2.2 Summary and progressive learning behavior ....................... 35

3.2.3 Cycle of self-efficacy ................................................... 36

3.3 Personas for inclusive and stable course plans ......................... 37

3.3.1 Contexts ................................................................... 37

3.3.2 Personas ................................................................. 39

3.3.3 Enforcement with discipline ........................................ 40

3.3.4 Facilitation with motivation ........................................... 41

3.3.5 PMD method ........................................................... 42

3.4 Steps to realize PMD method ............................................ 43

3.4.1 Identifying Personas .................................................... 43

3.4.2 Finding Discipline ...................................................... 46

3.4.3 Choosing services containing affective components ............... 47

3.4.4 Choosing facilitating services ........................................ 48

3.4.5 Integrating services into feasible course plan ..................... 49

3.5 An example of PMD method ............................................ 50
3.5.1 Training condition ........................................ 50
3.5.2 Personas from contextual inquiry ......................... 51
3.5.3 Early submissions and long learning time .................. 54
3.5.4 Hard deadline assignments and weekly tests ............. 56
3.5.5 Facilitating services ...................................... 57
3.5.6 New course plan ......................................... 58
3.5.7 Results and discussions ................................... 58
3.6 Conclusions .................................................. 63

4 Association of internal factors with behavior factors ....... 65
4.1 Introduction ................................................... 65
4.2 Formative assessment ........................................ 67
  4.2.1 Formative assessment process ............................ 67
  4.2.2 Formative assessment in programming education ......... 69
4.3 Formative assessment based on internal factors ............. 70
  4.3.1 Persona types .......................................... 70
  4.3.2 Persona judgement ...................................... 71
  4.3.3 Internal factor based NMF transformation ............... 72
4.4 Figuring out actual associations ............................. 75
  4.4.1 Behavior factor values to be analyzed .................. 75
  4.4.2 Associations using vectors .............................. 76
## Contents

4.4.3 Associations of internal factors with personas ........................................... 77
4.4.4 Associations of internal factors with behavior factors ................................. 79
4.5 Discussion: Preciseness in distinction ............................................................... 79
4.6 Conclusions ......................................................................................................... 80

5 Grasping motivation and learning strategies from behavior .................................. 81

5.1 Introduction ......................................................................................................... 81
5.2 Behavior factors and internal factors ................................................................. 83
  5.2.1 Behavior factors ......................................................................................... 83
  5.2.2 Learning status .......................................................................................... 84
  5.2.3 Adaptive supervision ............................................................................... 86
  5.2.4 Obtaining student learning status ............................................................. 87
5.3 Learning status through NMF .......................................................................... 88
  5.3.1 Internal factors reasoning behavior factors .............................................. 88
  5.3.2 Personas from past students .................................................................... 91
  5.3.3 Non-negative matrix factorization ............................................................ 92
  5.3.4 Perspective for decomposition ................................................................ 94
  5.3.5 Making full use of previous course .......................................................... 95
5.4 Application for actual courses ............................................................................ 97
  5.4.1 Application outline .................................................................................... 97
  5.4.2 Learning behavior of students ................................................................. 98
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.3</td>
<td>Cleansing values</td>
<td>98</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Five kinds of personas</td>
<td>99</td>
</tr>
<tr>
<td>5.4.5</td>
<td>Weight matrices of previous course</td>
<td>100</td>
</tr>
<tr>
<td>5.4.6</td>
<td>Gene matrices of previous course</td>
<td>101</td>
</tr>
<tr>
<td>5.4.7</td>
<td>Two-step NMF</td>
<td>102</td>
</tr>
<tr>
<td>5.5</td>
<td>Evaluation and discussion</td>
<td>102</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Evaluation overview</td>
<td>102</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Prediction of learning status</td>
<td>103</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Accountability for active students</td>
<td>105</td>
</tr>
<tr>
<td>5.5.4</td>
<td>Accountability for accuracy</td>
<td>107</td>
</tr>
<tr>
<td>5.6</td>
<td>Conclusions</td>
<td>109</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions and future works</td>
<td>111</td>
</tr>
<tr>
<td>6.1</td>
<td>Difficulties in building course plan</td>
<td>111</td>
</tr>
<tr>
<td>6.2</td>
<td>Solutions</td>
<td>112</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Design of a course plan with PMD method</td>
<td>112</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Formative assessment of personas with NMF</td>
<td>114</td>
</tr>
<tr>
<td>6.3</td>
<td>Future works</td>
<td>115</td>
</tr>
</tbody>
</table>

References 117
List of Figures

1.1 Common problems of a programing course .................................. 4
1.2 A course plan for introductory programming ............................... 12
3.1 A context is a chain of states ...................................................... 37
3.2 PMD method with enforcement of discipline ............................... 43
3.3 Steps to realize PMD method ...................................................... 44
3.4 Personas from contextual inquiry ................................................. 53
3.5 Learning time of top students and others .................................... 55
3.6 Submission rate of CP 2010, CP 2011, and CP 2012 ....................... 59
3.7 Learning time of students in 2010, 2011, and 2012 ......................... 60
3.8 Score of students of CP 2010 ...................................................... 61
3.9 Score of students of CP 2011 ...................................................... 61
3.10 Score of students of CP 2012 ..................................................... 62
4.1 Strength of transformations ...................................................... 73
4.2 Transformation of behavior to internal factors .............................. 74
5.1 Many internal factors affect a behavior 89
5.2 Decomposing behavior matrix by NMF 93
5.3 Contextual inquiry for NMF initialization 95
5.4 Prediction of the active and the passive 104
5.5 Rates of active students 106
5.6 Evaluation of persona prediction 108
List of Tables

3.1 Course schedule ................................................................. 50
3.2 Days until submission .......................................................... 55
3.3 Confirmation of motivation components of services and attributes .... 56
4.1 Personas ................................................................. 71
4.2 Example of behavior factors .................................................. 76
4.3 Association of personas with internal factors .............................. 78
4.4 Association of internal factors with behavior factors ..................... 79
5.1 Behavior matrix ................................................................. 85
5.2 Learning status vectors of personas ........................................ 100
5.3 Weight matrix initialization ................................................... 101
5.4 Course settings in 2012 and 2013 ........................................... 105
Chapter 1

Introduction

1.1 Problems of a C programming course

Although programming abilities are essential for any college students whose major fields are IT, many of them are suffering from attainment of the abilities. Especially, it is serious they feel difficulties to learn programming even in an introductory programming course [31, 20]. To see its reasons, we must reveal problems in the course, subjecting what happens in the course to the light of objective facts.

An introductory programing course is often involved in the chain of IT training program. Its goal is predetermined in consistence with other courses of the training program. The course often consists of 15 weeks. Teachers divide the course goal into weekly sub-goals of knowledge and skills. They organize the learning topics and assignments into a course plan, which describes how the course will be conducted. The teachers expect as many as students will master the weekly sub-goals comprehensively to accomplish the course goal. However, it is always a big matter how to make the students master the weekly sub-goals, because there are so many kinds of students. A course plan motivating high-level students usually disappoints students who are poor at programming, while
an easy course plan preferred by poor students seldom encourages high-level ones. The teachers must build a good course plan before the course, to ensure a successful course. Therefore, they discuss many times to make the course plan.

If the teachers let their students learn freely, it would be difficult for the students to achieve the course goal [3]. The teachers should impose a discipline to make the students study toward the achievement of the goal [53, 4]. If the students learn following the discipline, they will get to the goal. At the same time, the teachers should prepare supporting services for the students [65, 23, 61, 16]. Even if difficulties occur during the study, the services should work so as to encourage students to continue their learning [9, 67, 17, 18, 13, 58]. The course plan is designed as a combination of a discipline and supporting services.

In order to establish the discipline as well as the supporting services of the course plan, the teachers have to investigate their students. They have to grasp prior conditions of their students before the course starts. Although The prior conditions involve students’ knowledge background, their inclination for learning is more important to predict how they will behave during the course. If the teachers do not understand their students, the students do not work as the discipline requires, even though they impose the discipline. For example, suppose a discipline obliges the students to solve all advanced assignments within one week deadline. However, students who lack their confidence in programming do not solve the assignments.

To know the student inclination, their motivation and learning strategies play vital roles. Many researches point out that motivation and learning strategy of a student are key components resulting in his or her leaning success [59, 30, 50, 60, 63]. Instructional Design [15] and Motivational Design [30] center on students to design the course and instructions. The methods try to examine motivation and learning strategies of individual students to make a course matching their preferences most. The teachers should not only inspect motivation and learning strategy of their students before the course, but also predict
those of the students during the course [15, 30].

Motivation and learning strategies are factors treated in a student’s internal process. They are not observable. It is difficult to obtain these factors with small effort. In common education, the teachers often use the following three ways to assess these factors [56]. The first one is observing the student learning directly. The second one is obtaining the ratings by other students toward the student. The third one is conducting self-report. However, these ways take large amount of effort and time from both of students and the teachers. Observing students learning directly would take much time of teaching staffs, which prevents them from giving supervisions to the students. Asking for other students to rate a specific student would burden them, taking their learning time. Making the students to do self-report such as conducting interviews or answering questionnaires also take their learning time. These methods reveal severe drawbacks. In addition, the teachers need the frequent information of their students. If these methods are applied frequently, the students are tired of giving their information. They give their information haphazardly. The information would not be the truthful one.

To build a good course plan, the teachers have to find a method to obtain motivation and learning strategies of the students frequently. The methods should not impose the extra load on the students [30]. The method should bring about the students information truthfully [30].

Only the collection of motivation and learning strategies of the students does not solve problems in programming course. Figure 1.1 summaries the problems which teachers have to deal with for the design of the course plan. It also states matters concerning the student side.

A course plan would be designed so as to achieve successive weekly sub-goals sequentially. Once the course plan is fixed, it is hard to modify the course plan during the course. It is inevitable for teachers to build a monolithic course plan, before the course starts.
1.1 Problems of a C programming course

Students have diversity in motivation and learning strategies. Teachers would cluster students into several groups from the viewpoints of motivation and learning strategies, to build an inclusive course plan which covers as many groups of students as possible. It also contributes to a consistent course plan to each group. Since appropriate supervision is predetermined to each group, it prevents supervision from varying with teaching staffs.

The teachers also have to consider the formative assessment for the coming course. At the beginning of the course, most of the students have strong desire to learn the subject. The more the course proceeds, the more difficult the learning topics are [8]. Their desire varies along the course [64]. Some students can follow the course with strong desire. But others feel hard to follow the course because they cannot understand the topics. The desire to learn of these students may fall down. They do not want to complete assignments to improve knowledge and skills. The teachers have to grasp the learning status of the students correctly to be able to response adaptively to the change.
Any course plan leaves some students stuck, because of mismatch of their learning strategies [36, 27, 40]. Since the monolithic course plan is rigid, its discipline and supporting services are unchangeable during the course. In such cases, teaching staffs should take care of the stuck students with customized supervision [30]. It is difficult with limited power of teaching staffs, if a course plan leaves many students stuck. The two famous design methods stated above assume a training condition where teaching staff including teachers and teaching assistants (TAs) are affordable in any circumstance. If the methods find a student cannot solve problems, the teaching staffs can be assigned to help him overcome the problems in time. However, in common programming training, there are very limited teaching staff. In usual case, one teacher and some TAs have to take care of many students (e.g. over 40 students) of a class.

The above states the problems under the view point of the teaching side. For the learning side, there is often lack of immediate feedbacks reflecting the status of their learning to them. They do not know how much they have achieved the goal. They do not know if their learning ways are good one. The obscureness of the learning makes the students demotivated.

Students taking the course are always wondering how they should learn to master the goal. The course plan should state it clearly to students [30]. However, any course plan cannot convince all students [15]. The mismatch of learning strategies degrades motivation of the students. Motivation and learning strategy of a student would be grasped frequently so that students needing customized supervision would be chosen for teaching staffs to be assigned. Here again, motivation and learning strategy of a student would be grasped frequently with formative assessment.
1.2 Motivation and learning strategies of learners

Motivation and learning strategies are internal process existing inside of a student. A student with little motivation to learn do not engage in the learning much [45]. A student with inappropriate learning strategies may not achieve the goal [45]. In other words, motivation and learning strategies explain the learning performance of the student.

Motivation is the degree the student desires to learn and the volition he holds to overcome difficulties to achieve the goal. The motivation urges him to take learning activities to achieve the goal. It represents how much the student is interested in the activities. It sustains him with the learning until achieving the goal.

The degree a student wants to engage in the learning results from many components. According to Pintrich et al. [44], they include 6 components: intrinsic, extrinsic, task value, control belief, self-efficacy, and affective. The intrinsic components involve a student’s perception of himself to be participating in a task for the reasons such as challenge, curiosity, and mastery. Opposite to intrinsic components, extrinsic components involve external factors such as score, reward, and praise of others. Task value components involve the evaluation of the student about the task under the view point of interest, importance, and usefulness. Control belief components concern the belief of the students about their effort. Self-efficacy components correspond to the belief about the student’s ability to achieve the task. Affective components refer the anxiety of the student caused from tests, deadline, and examinations.

All of the motivation components contribute to motivate the student to the learning to achieve the goal. The student with motivation components other than affective ones engages in the task by his own desire. In the meanwhile, the student with affective components works because he is afraid of the factors such as tests.

Motivation of a student strongly affects his learning to obtain knowledge and skills. It influences what, when, and how he learns [55]. A student who has motivation to learn
can engage in the activities he believes to help him. He intentionally attends to lectures, takes notes, organizes the knowledge points, does rehearsals what he has learned, asks for helps when he cannot solve problems, and check his understanding level. He takes positive beliefs of the learning. Collectively, these activities improve the learning.

Motivation is an internal process that urges the student to learn and maintains his learning until the goal is achieved. On the other hand, learning strategies are approaches the student takes to achieve the goal. Pintrich et al. [44] points out 9 strategies common in students. They are rehearsal, elaboration, organization, critical thinking, meta-cognitive self-regulation, time and study environment, effort regulation, peer learning, and help seeking. Rehearsal refers to the review of the learning matters to put them into long term memory. Elaboration is the strategy to elaborate key points, knowledge structure, categories, etc., to master them comprehensively. Organization is the ability to connect prior knowledge in a logical form. Critical thinking refers to the capability to refer to and to use prior knowledge to solve the current problems. Meta-cognitive self-regulation involves the consciousness of a student himself about his cognition. Time and study environment expresses the ability of managing the time and the learning environment. Effort regulation concerns the effort against distraction and difficulties to accomplish the goal. Peer learning refers to the learning cooperation ability. Help seeking concerns the ability to seek for helps to overcome problems.

In order to success in a specific learning subject, it is important for the student to take appropriate learning strategies. Programming is a hard subject which requires students to master much abstract knowledge. It also demands them much practice to improve programming skills. If the student takes enough critical thinking, proper meta-cognitive self-regulation, good understanding of learning environment, appropriate time to study, enough effort regulation, he will achieve the programming goal.

The teachers should assess motivation and learning strategies of every student to understand the learning status of a student [29, 32]. However, motivation and learning
strategies are not observable factors. The teachers have to infer them from applying the methods on the students such as observation, ratings from other students, and self-report.

1.3 Personas to characterize student groups

The students of a programming course vary in motivation as well as in learning strategies. They have different degree for each motivation component. They take different approaches in every learning strategy. A student with low intrinsic motivation has little inspiration and desire to learn the subject. The teachers should make him understand the usefulness of the subject, as well as make him to experience certain success. Appropriate supervisions should be prepared for the student [30].

In an optimal condition, the course should match all preferences of individual students during the course. However, it is impossible in real situations. There is limitation of teaching staffs who are assigned to give supervisions to the students. In common situation, one teacher and some teaching assistants are assigned to care for about 40 students of a class.

Beyer and Holtzblatt proposed the contextual design [5] to enable software systems to provide appropriate services for users of diversities. The contextual design plans software systems according to the following guideline:

- The method examines user requirements with contextual inquiries.

- It puts users similar in their requirements together into a group.

- To each group, it assigns a persona, a virtual user who represents the group.

- A scenario is defined to state how the persona behaves in a given context.

- The method prioritizes personas the system should target.
1.3 Personas to characterize student groups

- According to the priorities, the method incorporates services into the system.

The method proposed in the thesis applies the contextual design to achieve a course plan for a class consisting of diverse learners. According to the guideline, we should examine groups of learners using a contextual inquiry when we are planning a course. We should decide which groups are primary ones or secondary ones to target in the course. Using the idea of user groups, we consider grouping the students based on their motivation and learning strategies. A group consists of similar students in term of degree of components of their motivation and learning strategies. Persona is a virtual student representing the characteristic of motivation and learning strategies of a group. The number of groups is much less than that of students. When designing the course, educators can consider specific groups to give careful supervisions, e.g. groups of students who are low in intrinsic components and task value components.

The students belonging to a persona have similar motivation and learning strategies in component and in degree of each component. They have similar degree of engagement in the learning and similar approach to obtain the knowledge and skill the course requires. When making the course plan, the educators examine the presence of all of the personas among the whole students. They consider taking care of all of the personas during the course as much as possible. Along the course, even though the persona of a student changes, the course can take care the persona-changing students. Since the course plan can take care all of the personas, the designation of TAs on proper students, especially personas who need supervisions to move forward their learning, can be realized.

The course plan is designed before the course. It is hard for educators to predict personas of a course which has not been conducted. However, in reality, student groups among the students in the same university are quite similar in characteristics in some successive courses. It means once educators have obtained the personas in the whole students of a course, they can assume the following courses have those personas with same characteristics. To obtain the personas for the course design, the educators can
use a contextual inquiry on the students of the previous course, where the students have experienced full learning items in the course.

We assume a large amount of students can be represented with few personas. It enables the teachers to take care of all the students. Because the teachers can master the characteristics of all personas before the course, they can consider the discipline appropriately to lead their learning to achieve the goal. All of the personas can be given factors to motivate the learning. It also enables the teachers to take care of specific personas, e.g. passive personas, if they can assess the students every week.

1.4 Inclusive course plan with discipline

The teachers organize the course goal using successive weekly sub-goals, each of which is achieved in the week with lectured topics and assignments. The teachers think that, if the students attend many lectures and solve many assignments, they will be able to master the week goal. The teachers request all of the students to attend the lectures and to solve assignments as much as possible. However, the learning ways the teachers thought do not always bring about good results. Excellent students try to solve many assignments early. They achieve good grades. However, many students who do not want to spend much time on the study. They do not know the importance of many assignments they have to solve. They solve only some easy assignments. As the result, they get behind the course.

Every course should have a discipline to achieve its goal. If the course plan does not embody the discipline firmly, the students would study as they want to. Since the students study as they prefer, it would be very difficult for the teachers to follow them during the course. There is high possibility that many students will not reach to the course goal.

Assume teachers examine records of past course, when they make a course plan, there was a programming course in the university. In the past course, the good students who
have achieved the course goal would take a particular learning ways. Their learning ways would be different from others who have not achieved the course goal. With the facilitation of support system, learning behavior of the students is taken automatically into learning logs. If the teachers compare learning logs for the weekly learning behavior of the top good students with that of others, they would understand their difference in their behavior. From the difference, the teachers take into account particular behavior which results in good achievement.

From a contextual inquiry of the past students, the teachers understand their learning scenarios which include motivation, learning strategies, and learning behavior. The teachers grasp the good learning strategies based on the associations of learning strategies and learning behavior. The course plan imposes a discipline to achieve the goal on the students of the new course. They study as if they know the good strategies by a discipline. Since the affective component of motivation makes the students study with their anxiety, we incorporate the affective components of every persona with the requirements of the course to make the discipline executable.

The course plan should be designed to direct all students to the course goal with the incorporation of disciplines. A good discipline is made of the good learning strategies and the affective components of the personas. The strategies are investigated from students performing well in the past course. The affective components are also taken from the motivational characteristics of the personas of the past course with contextual inquiry on the past students. The affective component makes them study as if they have the good learning strategies to achieve the goal.

At the same time, the course plan facilitates many motivating services to match the motivation preferences of each persona to keep them maintain their motivation to learn. Under the discipline of the course plan, all of the students would learn as if they have good learning strategies to achieve the goal. During the learning, there would be students feel hard to keep up with the class. The teachers have to predict learning contexts occurring
1.4 Inclusive course plan with discipline

during the course. For example, some students cannot solve hard assignments. They want to ask teaching staffs for help. The teachers include motivating services such as assigning teaching staffs for the course into the course plan. As stated above, there are few personas for the students. Each persona characterizes motivation and learning strategies as well as their own motivating factors which incline them toward learning. The contextual inquiry enables teachers to provide the motivating service. The course plan should incorporate these motivating factors for all of the personas to motivate them along the course. In the actual situation, there would be many motivating factors enumerated. The teachers may eliminate some of them, considering the balance of the services toward all of the personas.

Figure 1.2: A course plan for introductory programming

A student may change his persona during the course. In a week, he may belong to an active persona because he has high intrinsic, high task value and high meta-cognitive component. In another week, he might encounter many assignments he cannot solve. He might find the subject uninteresting. His task value falls down to low value. He belongs to a passive persona. Since the course plan takes into account all of the personas in the students along the course, it is inclusive to all personas in any week. Figure 1.2 describes the course plan.
We applied the course plan, which is built up according to the contextual design, for C programming courses in years 2011 and 2012 in Ritsumeikan University. The course prevented students from losing their motivation to learn. Their score and their learning time kept horizontally stable along the course.

1.5 Formative assessment of personas

When designing the course plan, the teacher must consider the methods to do formative assessment. When the learning topics become difficult, there will be students who feel hard to follow the course, e.g. the assignments are too difficult for them to solve. If they cannot overcome the problems, they may get de-motivated. The teachers should detect the students in time to give timely supervision to them.

The assessment is an indispensable work in every week of the course. With the assessment, the teachers understand what degree of the weekly goal and the entire goal of the course has been reached to by each student. Thereafter, a teaching staff can be designated for the students. The teaching staffs will give adaptive supervisions to help them move forward their learning so that they can catch up with the course.

Formative assessment should be conducted every week [1]. Common formative assessment often uses questionnaire. However, it takes the students much time. In addition, if the questionnaire is conducted many times, the students feel tired of giving answers. They would answer the questionnaire in a slapdash manner. The results from questionnaires do not reflect the learning truthfully.

There are assessment methods from student behaviors which are taken automatically using a web site. The web site supports the students in displaying the curriculum, lecture contents, assignments, and so on. The learning behavior of every student can be understood with a certain degree from the learning logs. Nevertheless, it is still hard to guess
the whole image of the learning of a student using the behavior. For example, the learning logs show a student submits many assignments and takes a good score. Nevertheless, in real situation, the student could not solve many assignments. He copied the codes from his friends and submitted them to the teachers through the support system. With a good score, the teachers may assess him as good student. In programming education where it is impossible for teachers to observe the whole process where the students learn, it is necessary to consider an assessment method which reflects the status of the learning correctly.

When students learn programming, many of the learning tasks are supported with computers. It is possible to take logs of most of students’ learning behaviors. The programming learning behavior expresses the learning of a student quantitatively. From the learning logs, we can extract many behavior factors and calculate the amount of each. Since it is hard to interpret the meaning of each behavior factor and its amount, the method proposed in the thesis tries to transform them into the amount of components of motivation and learning strategies. With the amount of every component, the teaching staffs can grasp a full image of the learning of the students. The amount of every component helps teaching staffs give adaptive supervision to a student.

Suppose a student takes an amount of time to solve assignments. He found the subject interesting. He should have certain intrinsic component [30]. He should also have certain task-value component [30]. The amount of time he spends on the learning would result from many components. Each component affects a specific behavior factor in proportion to the amount of the behavior factor. We assume that, the amount of a behavior factor is linear sum of the components with certain weight for each. The weights represent the strength of every component a student holds. The method figures out the weights for every student.

A student holds many behavior factors, forming a behavior vector. We transform the behavior vector into the weight vector representing the amount of each component. A
programming course has many students, forming a behavior matrix. In the same way, weights for all components of the students also form a weight matrix. All elements in these matrices are non-negative.

Under specific course settings, which consist of same assignments, same requirements, and same teaching staff for all students, it is possible to use non-negative matrix factorization (NMF) to decompose the behaviors matrix into two matrices, one of which is the weight matrix.

The amount of every behavior factor of a student in a week can be calculated from the learning log at the end of the week. With the decomposition, we can obtain the status of the learning of every student at the end of every week without imposing burdens on the teachers and on the students.

In order for NMF to produce a good weight matrix, it is essential to provide initial values for the matrices. We utilize the status of the learning of the students of a past course. We applied the assessment method based on NMF for a C programing course in year 2013 in Ritsumeikan University. With the status of the learning calculated by the NMF, we figured out the persona for every student. We found that, if a student takes active personas in a period of three continuous weeks in the second half of the course, it is more than 70% that he will take active persona at the end of the course. The finding implies that, the teaching staffs can ignore the students who take active personas in the second half of the course. They should focus on only students who take passive ones to make them become active. Therefore, instead of taking care of all students, teaching staffs can focus the care for only passive persona students to achieve an efficient course.
1.6 Approaches and organization of thesis

Programming is a subject which contains a lot of abstract concepts and practical skills. It requires the students to spend much time to master them. Studying only in class time, e.g. 90 minutes, is not enough to obtain them. The students should be encouraged to self-study anytime anywhere. The programming learning topics, assignments and their requirements must be accessed easily. Scores of assignments as well as feedbacks reflecting their learning status should be delivered to them in time.

A learning support system is often facilitated to assist the works. A learning support system becomes indispensable for programming courses, because it contributes to the management of the education in many aspects. The discipline and supporting services of the course plan are incorporated into the learning support system. The system delivers the content of the course to the students. It enables them to view the scores, the feedbacks, and the learning progress, anytime anywhere. If we monitor how individual students make access to it, we can grasp the learning behavior of them, which is useful for the formative and summative assessment.

Approaches in the thesis work on improvement of the introductory programming courses, based on motivation and learning strategies of individual students. The approaches analyze learning behavior logs of individual students collected through a programming support system. Among the problems, the thesis addresses the following ones:

- a course plan to make the students to study to master the weekly goal, and
- formative assessment during the course.

To explain the approaches, the thesis is organized into 6 chapters. Chapter 1 introduces the main problems of the design of a C programming course including the formative assessment. Because the solution is based on motivation and learning strategies of students, we present them in detail in chapter 2. Chapter 3 explains the method to design a course
1.6 Approaches and organization of thesis

plan for the course. It also presents a learning support system to collect learning behavior
logs of individual students. Chapter 4 and Chapter 5 focus on the formative assessment.
Chapter 4 examines the relationships of learning behavior of students with their motiva-
tion and learning strategies. Chapter 5 presents a method to quantitatively figure out
motivation and learning strategies of individual students from their learning behavior.
Chapter 6 summaries the work and states the future work.
1.6 Approaches and organization of thesis
Chapter 2

Motivation and learning strategy of student

2.1 Motivation

2.1.1 Definition

There are many definitions of motivations. Nevid [41] states "The term motivation refers to factors that activate, direct, and sustain goal-directed behavior". According to Slavin [59], "Motivation is an internal factor that activates, guides, and maintains behavior over time". Pintrick et al. [44] define "Motivation is process whereby goal-directed activities are instigated and sustained." These typical definitions of motivation concern two main issues.

First, motivation of a student is an internal process. It is not observable. A student is considered motivated, if he has inspiration or desire to obtain programming knowledge and skill. When a student is motivated, he is energized and activated to achieve the goal. In the meanwhile, a student who feels no inspiration to learn is considered unmotivated.
Motivation is the feeling that activates, urges the student to learn. It keeps him engaged in the learning until achieving a certain goal.

Second, the internal process targets a goal. The motivation is toward a specific goal. It directs behavior to achieve the goal. When a student is motivated to master programming knowledge and skills, he initiatively engages in the activities such as joining lectures, taking notes, and solving programming assignments intentionally for the goal. The goal helps the student be conscious of what they are trying to obtain. Motivation encourages activities to attain the goal. Motivated activities are sustained directing the goal.

Because motivation is not observable, we cannot observe it directly. We infer its presence from behaviors and verbalizations. For example, when we see a student spends time to solve programming difficulties he encounters, we infer that he has motivation to learn programming. If a student says "I really want to challenge this programming assignment", he is considered to have high motivation for this task.

Toward a specific goal, individual students would show different desire and interest to obtain it. A student with strong desire and interest to learn a topic would engage in the learning stronger than a student with little ones. A student wants to learn programming because he feels enjoyable, while another tries it to get high score for good credits. There are many factors and reasons that cause a student motivated.

2.1.2 Motivation components

Motivation of a student is the inspiration to urge him to take activities to achieve a specific goal. The inspiration may come from, or result from the insides of students, such as enjoyment and satisfaction. The inspiration may come from factors which are not directly related to the activities themselves. These factors are such as score, rewards, and evaluations of others.

Motivation components involve factors which cause the inspiration toward the goal.
Let us examine some specific examples of motivation components in programming learning.

- **Peter**: He is smart but he does not like to work with a computer. He prefers to work with a human rather than a machine. Since the programming subject is a compulsory one, he is obliged to work on it. He often engages in programming exercises with his friends to avoid being bored.

- **Anna**: She is an intelligent student. She is attracted to programming learning. She is motivated to learn new things and enjoys the exploration. She makes goals for herself and tries to reach to the goal with her best effort.

- **Jane**: She is a very intelligent student. She often gets the highest score of programming exercise. Jane wants to be the student who has the highest score all of the time. If she is praised by the teacher or by her friends, she feels highly motivated.

- **Rose**: She wants to be an information engineer in the future. She learns programming because she has found it is a useful subject for her future job. She has also found the subject is interesting.

- **John**: John does not work hard to get good score for programming. If he gets a good score, he feels that he is lucky or that the task is easy. He never believes that he is good at the subject.

- **Lili**: She is an industrious student. She has a strong belief of her capacity to learn programming. When she encounters a problem, she tries to overcome it in various ways. She can solve them most of the times. Her confidence is getting stronger and stronger.

- **Deana**: Even though she tries to do assignments spending many hours, she does not get high score. It makes her lose the belief that effort does not betray any person.

- **Mary**: She is motivated to avoid being a poor student in the class. She often looks at the web page of the learning support system where the score range of the class
is displayed. She just wants to get enough score to be an average student because she is certain that she can get credit with an average score.

Each of the above students learns programming with different reasons. The Motivated Strategies for Learning Questionnaire (MSLQ) [44] arranges these reasons into 6 categories as follows.

- **Intrinsic component**: The component involves the perception of himself to be participating in a task for the reasons such as challenge, curiosity, and mastery. Anna is considered as a student with high intrinsic toward the programming learning. In the meanwhile, Peter has low intrinsic toward the learning. The intrinsic motivation of a student sustains his feeling to achieve the oriented goal.

- **Extrinsic component**: Opposite to intrinsic component, an extrinsic component involves external factors such as scores, reward, and praise of others. It is the means to make the students to participate in the task. Jane has extrinsic motivation in the above examples.

- **Task value component**: It involves the perception and evaluation of the students about the task in term of interest, importance, and usefulness. Rose is a student who has this motivation component.

- **Control belief component**: It concerns the belief of the student about his or her effort. If he believes that the more he takes effort the more achievement he will get, he would be more active to study. Deana is the student who loses this component.

- **Self-efficacy component**: It is the belief about the student’s ability to achieve the task. Lili is a student with strong self-efficacy. If a student does not believe she has an ability to achieve the task, she would work lackadaisically on difficulties. She would give up easily when she does not understand.
2.2 Learning strategies

- **Affective component:** If the student works because he minds of tests or external factors, he is considered to have affective components. Mary is afraid of her score. The worry of low scores makes her work. Peter is another student who works because of an affective component. He tries to study programming because it is a compulsory subject.

A student with motivation to learn programming will engage in the activities he believes helpful for him to achieve the goal. He attends lectures, takes notes, organizes the knowledge points, does rehearsals what he has learned, completes homework, and asks for helps when he cannot solve assignments. He takes positive beliefs of the learning. All of the activities contribute to improve his learning.

Among motivation components, the intrinsic one is what the teacher expects their students hold and improve time by time. With the intrinsic component, students find activities to learn programming interesting, and challenging. They can engage in the activities without any obvious external reward such as scores or praise of others. They enjoy activities or find it as an opportunity to explore new programming knowledge and master new skills [12, 7].

Among the components of motivation, the affective component causes students to learn with a different mechanism. The students with the motivation components other than affective one work and engage in the activities by his own will. In the meanwhile, the student with affective components works because he is afraid of factors such as tests or deadlines.

### 2.2 Learning strategies

Motivation is the internal process that urges and sustains the student inspiration and will to achieve the goal. Learning strategy, in the other hand, is the approaches and
methods the student takes to achieve the goal. Pintrich et al. summarize components of learning strategies in the Motivated Strategies for Learning Questionnaire [44]. The components compose of two categories. One is cognitive and meta-cognitive strategies. These strategies involve the cognition of learning activities and of how to utilize them to achieve the goal. They consist of rehearsal, elaboration, organization, critical thinking, and meta-cognitive self-regulation. The other is resource management strategies. They involve the understanding of the learning environment as well as the capacity of the student to take its full advantage to achieve the goal. The category includes time and study environment, effort regulation, peer learning, and help seeking.

- **Rehearsal**: Rehearsal is practicing what has been learned. In programming learning, this strategy helps the students recall programming concepts, codes, etc.

- **Elaboration**: It refines important points of the knowledge. It contributes to putting them into long-term memory. It involves categorizing pieces of knowledge, and summarizing them. The strategy helps the students identify key points of each lecture. It also helps them integrate the new knowledge with the prior one.

- **Organization**: The strategy concerns the ability to pick up knowledge point and connect them in a logical and connecting fashion. It often outlines, and groups knowledge points.

- **Critical thinking**: It is the ability to use prior knowledge to understand the new knowledge, or to solve real problems. The strategy helps the students develop their own ideas to overcome difficulties.

- **Meta-cognitive self-regulation**: It is the degree of consciousness of a student about his cognition. With the consciousness, the student would have ideas to control, or regulate himself to make the situation better. For example, a student knows he is often asleep in the lectures. To avoid the asleep, he takes a cup of coffee before the class.
• **Time and study environment:** It involves the ability to manage time and learning environment. Scheduling and controlling the time to achieve a task is very important in programming learning because the learning often takes much time. Well understanding of the learning environment helps the student utilize provided services best.

• **Effort regulation:** Problems often occur during the process to achieve it. For example, the task becomes boring. The work gets difficult. Deadline is coming. Something attractive irrelevant to the learning appears. Achieving a task requires certain effort. Effort-regulation refers to the self-control ability to achieve the task. "Effort-regulation strategies work in the process whereby learners systematically direct their thoughts, feelings, and actions toward the attainment of their goals", states Schunk et al. [54].

• **Peer learning:** It is the ability to collaborate with friends to accomplish a task. It is found that collaborating learning is an effective and interesting way in programming learning.

• **Help seeking:** Suppose a student faces with a problem. If he cannot find how to solve it by himself, he should look for helps. The learning environment often provides supporting services such as teaching assistants, and tutors. The Internet is another means to find solutions from others. Good students know where and when they should ask for helps.

Since programming is a subject which contains a lot of abstract concepts, it requires the students to take a lot of effort to master them. It also requires them to practice to make many program codes to obtain programming skills. It requires the students to overcome many difficulties. To achieve the programming knowledge and skills, the students have to take various learning strategies appropriately. Among the strategies, effort regulation is a strong predictor for success in on-line programming courses [68, 57].
2.3 Obtaining motivation and learning strategies

There are many ways to obtain motivation and learning strategies of the students such as direct observation, rating from others, and self-report.

It is easy for the teachers to intentionally observe a student’s learning behavior. From the observation, they can understand to what extend the student involves himself in the learning activities, how much effort he takes to solve problems. However, the direct observation may be superficial. Motivation and learning strategies vary with individual students. They result in learning behavior of every student, affecting with each other in a specific context. The teachers may fail to capture essential components of motivation and learning strategies of the student, because they are complicated. The teachers have to infer the presence of the components from the student behavior they observe.

Although direct observation cannot bring a comprehensive understanding of motivation and learning strategies, it is often used in common education [21]. The teachers may infer some motivation components with the observation focusing on the persistence at certain activities, the effort to perform well the activities, and how willingly the student engages in those activities. However, it is an ad hoc approach to determine the components from only certain activities. It is dangerous not to judge the components in a comprehensive way. There is also a drawback the direct observation is apt to pay attention to negative points in motivation and learning strategies of the student.

The teachers can have others (e.g. students, parents, and teachers) rate the students on various components of motivation and learning strategies. For example, the teachers can ask them whether Anna makes effort to solve the assignments. Rating from many people reveals more components than the direct observation. It is more objective than the direct observation. Owing to rating from many people, it brings more information about motivational process which underlies behavior. However, it still tend to indicate negative points like the direct observation, because it is difficult for people to judge the
2.3 Obtaining motivation and learning strategies

level of cognitive engagement as it is.

Self-report includes questionnaires and interviews. It founds on evaluation by students themselves, instead of judgement of others. Unlike the direct observation and the rating from others, it indicates positive points as well as negative ones. The following section explains these two methods in detail.

2.3.1 Questionnaire

Motivation brings about the student’s learning ambition. Learning strategy explains the student’s approach and methods to achieve the learning tasks. The educator wants to know motivation and learning strategy of all students of their course, to adjust their course correspondence to many of the student’s perspectives. Moreover, for supervision to individual students, it helps them adapt to each student preference.

Questionnaire is a common method which is widely used to obtain motivation and learning strategies of the students. With questionnaire, the students will choose an answer for a question in case of a close question. In case of an open question, they can give their opinions for a predetermined matter.

Since the questionnaire is often organized in a well-format, it takes educators small time to collect data from the students. It also takes little time to pre-process the data for analysis. It brings about a practical advantage, because the analysis which seldom takes much time enables the educators to provide analysis results back to the students in a short time period. To obtain student’s motivation and learning strategies, educators can utilize sample questions for every components, such as ones given in MSLQ [44] presented by Pintrich et al.

Though questionnaire has many advantages, it reveals many drawbacks when we apply for obtaining motivation and learning strategies. Most of the time, the students think that, they do not have any benefit to give answers to the questionnaire. They often
give cursory answers, especially to questions which are taking time to answer. Therefore, the questionnaire results do not express the student motivation and learning strategies correctly.

Students, especially slow in learning ones, are afraid that if they give truthful answers about their own feeling and approaches toward the learning, they may be treated in a particular way. They intentionally avoid giving truthful answers. As a result, the questionnaire brings about only good evaluations from students.

In addition to that, MSLQ consists of 6 components of motivation, and 9 components of learning strategies. There are 15 components in total. If educator uses one question to obtain each component, they have to prepare at least 15 questions in a questionnaire. To improve preciseness of the analysis, more than one questions are necessary for each component. Many questions in a questionnaire would take them much time to answer.

Motivation as well as learning strategies would vary during a programming course. The educators should conduct the questionnaire frequently during a course, which make the students tired of answering. It is almost impossible for obtaining truthful information from students. Actually, we tried to obtain student motivation and learning strategies using a questionnaire based on MSLQ in a course consisting of 15 weeks. To obtain motivation and learning strategies, sample questions in MSLQ are to be answered in 7-scale (-3, -2, -1, 0, 1, 2, 3). Choosing one option, students express the degree of agreement and disagreement to the question. We conducted questionnaires in the 8th week, the 11th week, and the 14th week. The class had 48 students. The number of students answering all of the questions in each questionnaire was 46, 40, and 16 respectively. In the 11th week and the 14th week, the questionnaire results contain only positive opinions of the students for every component.

Questionnaires founds on subjective answers from students. Even though they prepare answer format in a qualitative fashion, educators can only understand rough evaluation
on a specific component of a student, from students’ answers in questionnaires. It is impossible to understand it with quantitative preciseness. In formative assessment, the educators should know every component of each student quantitatively to give appropriate feedbacks.

### 2.3.2 Contextual inquiry

Educator can use the contextual inquiry [5] to obtain student motivation and learning strategies. In this method, the educators conduct a detailed and deep interview for students about their motivation and learning strategies. The interviewers write down the interview contents as scenarios.

The scenarios contain matters such as why they study the subject, what motivates them, how they study it, and how they overcome difficulties. The educators read the scenarios to elaborate information involving motivation and learning strategies.

In order for the student be able to explain their programming learning comprehensively in the interview, the educator should let them conduct the interview after they have experienced the programming course.

As it is obvious from the process, the contextual inquiry imposes heavy loads on both of students and the educators. Student time is consumed for interviews which are irrelevant to their learning. Educators take much time for reading and analyzing the scenarios. From the view point of time and power, it is hard to apply this method frequently to understand students.
2.3 Obtaining motivation and learning strategies
Chapter 3

Course plan with discipline utilizing persona motivation

3.1 Introduction

Programming practice courses are often compulsory subjects of engineering training in many universities. Since the number of learners taking the course often reaches to hundreds, the course involves many teachers and teaching assistants (TAs).

Motivation and learning strategies are essential elements for learners to be successful in the learning [43]. Both of them have many components [44]. Large number of learners have wide diversity in the components. Some learners study programming for their curiosity. They control their time for programming. Other learners uninterested in programming engage themselves in it under support of friends, because it is a compulsory subject. The course should be inclusive, which means it must provide benefits not for specific learner groups, but for as many learner groups as possible.

Moreover, the learners change their motivation and learning strategies as they improve their abilities. The Instructional Design methods [50, 15] consider it important to follow
the changes to give services suitable for individual learners. However, any adjustment during the course is critical in case many parties are involved. Since the adjustment may affect some parties, it needs assent from them, which prevents timely accommodation. In addition to that, many educators such as teachers and TAs take charge of hundreds of learners. The success of the adjustment depends on the ability of each educators, which lacks the fairness in education. Therefore, the course plan should initially involve services helpful for learners in various situations they may encounter throughout the course. In other words, the course plan should be *stable*. However, it would be infeasible to include tactics for huge diversities coming from hundreds of learners into a course plan.

To address the diversities, studies based on the ARCS model [30] esteem learner motivation. Motivating factors in e-learning tools for programming are discussed in [33]. McGill [38] evaluates programming education with robots from formal and qualified feedbacks from learners. Motivation transition of learners in programming are analyzed in [64]. An example of programming course design methods based on the ARCS model is illustrated in [28]. Many kinds of tactics are proposed to maximize the performance of all learners [39, 6, 14]. Nevertheless, none of them addresses a design method to build an inclusive and stable course plan in an educational environment where hundreds of learners are supervised by many educators.

We propose the PMD (Persona, Motivation, and Discipline) method to design inclusive and stable course plans for environments many parties in large quantity take part in. The method assumes characteristics of whole learners usually remains unchanged for a few years. It takes actual contexts of past learners and their operations on the learning supporting site. A persona combined with its scenario is a model of learners similar in motivation and learning strategies. The model enables teachers to predict their learning behaviors. Before the course starts, the teachers determine educational services based on the prediction. Course designers extracts a discipline from self-regulation strategies of high achievement learners to direct various contexts of all learners toward the course plan.
3.2 Introductory programming practice course

In the PMD method, the teachers make a balance of facilitating services with the discipline, referring motivation components stated in a scenario of each persona. A persona provides every educators with a common image of a group of learners similar in motivation and learning strategies. Founding on the common image, a course plan is designed with educational services for motivation of every persona and the discipline. The PMD method minimizes variances in supervision among many educators.

Applying the PMD method to over 500 students supervised by 10 teachers and 100 TAs, we have improved the learning time of the students and succeeded in keeping their scores even for harder assignments in 2 successive years.

3.2 Introductory programming practice course

3.2.1 Motivation and learning strategy

Motivation and learning strategies of a learner play vital roles to his or her success [59]. Pintrich et al. [44] have proposed "A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)" which is a guideline to understand motivation and learning strategies of a learner. It explains motivation and learning strategies with multiple scales. For each scale, it provides question samples for easy reference. The scales can be used to identify motivation and learning strategies of a learner studying programming.

For motivation, MSLQ lists up intrinsic, extrinsic, task value, expectancy, and affective components. Learners who have intrinsic components participate in target tasks because their goals are achievement of the tasks. They work because of reasons such as their challenge, curiosity, and mastery. For intrinsic components, MSLQ lists example questions like "In a class like this, I prefer course material that really challenges me so I can learn new
things.”. MSLQ also indicates the correlation of the question with the final grade is 0.22. Learners depending on extrinsic components participate in tasks because a learning task is the means to their goals. Components such as rewards, scores, and competition make them engage in the tasks. Task value components are determined by learner perceptions of the tasks in terms of interest, importance, and utility. Expectancy components refers to learner beliefs that their efforts will results in good outcomes. Affective components are related to anxiety for tests, grading for credits, and so on. For example, since they worry about tests, they make efforts.

A learning strategy is a personal approach to understand information and solve problems. Learning strategies consist of resource management strategies and self-regulation strategies. The resource management relates to the learner ability to understand the usefulness and effectiveness of given resources for the learning. As for resource management, MSLQ enumerates study environment strategies regarding to setting of study places, peer learning strategies corresponding to collaboration with peers, and help seeking strategies to get supports from others including peers and instructors. Self-regulation strategies work in the process whereby learners systematically direct their thoughts, feelings, and actions toward the attainment of their goals [54]. Yukselturk [68] states self-regulation learning strategies are strong predictor for success in on-line courses. Self-regulation strategies in MSLQ involve cognitive and meta-cognitive strategies, strategies on time, and strategies on effort regulation. Cognitive strategies refer to methods to learn, such as how to remember new concepts, to organize them, and to understand them, while meta-cognitive strategies correspond to methods to tune and adjust cognitive strategies. Strategies on time refers to planing study time. Effort regulation strategies control their efforts in the face of distractions and uninteresting tasks.
3.2 Introductory programming practice course

3.2.2 Summary and progressive learning behavior

To improve the usability of a working environment, the contextual inquiry method [5] insists to delve into working processes of users with interviews to them. To establish a good course plan, we need to know every detail of behaviors of learners in specific conditions. Suppose the learner mentions help from her friend to solve the assignment, when an interviewer asks how she solves an assignment. In the contextual inquiry, the interviewer should delve into topics the learner mentions. In this case, for example, the interviewer asks why the help occurs, how the help takes place, and how the learner feels after the help. The further the interviewer delves into the context, the clearer image of her learning the interviewer obtains. In the contextual inquiry, the interviewer describes a scenario summarizing what the interviewee has experienced. The scenario expresses contexts [11], which consist of sequences of actions and feelings of the learner in specific conditions. The method refers to behaviors in the scenario as summary learning behaviors. Summary learning behaviors contain useful information to identify motivation and learning strategies of the learner.

To obtain a clear image of learning behaviors in specific conditions, a questionnaire would not be a good method, because questions in it are pre-determined with assumptions of investigators. Learners who give answers to the questionnaire may not have real experiences for conditions assumed in questions. It prevents collection of truthful information from learners. The contextual inquiry lets us know what behaviors learners take in actual conditions.

Web servers facilitate learning anywhere and any time. They also enable to take operation logs from learners to understand their learning progress. The logs express interactions of learners with the servers through Web pages such as viewing assignments, downloading sample codes, and submitting source codes. Since the logs are taken throughout the course, learning activities are recorded in a progressive way. The logs express progressive
Together with summary learning behaviors from the contextual inquiry, progressive learning behaviors bring information expressing motivation and learning strategies of learners. Summary learning behaviors are obtained from oral words of learners. Meanwhile, progressive learning behaviors are series of genuine actions learners actually take. Suppose the teachers try to support their motivation and learning strategies using only summary learning behaviors. Information in interviews is not always expressed in a quantitative manner. The teachers should confirm summary learning behaviors against progressive learning behaviors. Compared with summary learning behaviors, progressive learning behaviors are far less expressive for what learners want, what they expect, what they mind, how they behave in specific learning contexts. It is necessary to combine the both kinds of learning behaviors to design a successful course.

### 3.2.3 Cycle of self-efficacy

Every learner has her own motivation components. Some learners work because of extrinsic components, and others for expectancy ones. Because of the diversity, requirements for a course plan vary with learners. Among them, intrinsic components are indispensable for successful learning. The educators should lead low intrinsic learners to a status where they have enough intrinsic motivation. Because of low intrinsic motivation, they are not willing to work on programming. However, if they work on programming assignments, they may be able to solve some of them. The experience gives the learners pleasures and confidence, which increases expectancy in them. Because of confidence, they get willing to work, which means the intrinsic motivation is improved in them. It is a good cycle of self-efficacy. Even if those learners start to work, they will stop learning, when they face difficulties in programming. Educators should support them, providing services facilitating their learning. For example, face-to-face supervisions of TAs should be prepared for learners who cannot solve easy basic assignments.
For learners whose initial intrinsic motivation is high, they would have high will to master the knowledge. The educators should give services to promote their motivation further. For example, hard graphics assignments with visual execution results would appeal them to explore programming more. Announcement of the score rank would promote them to try to solve many hard assignments. If there is no service to maintain and promote their intrinsic motivation, the intrinsic motivation gets decreased [30].

3.3 Personas for inclusive and stable course plans

3.3.1 Contexts

Some learners are eager for programming, while others have no interest in it. Every learner has her own state in learning. States strongly depend on her motivation and learning strategies. In a learning process, a learner receives a stimulus. Her state changes into a new one as a reaction to a stimulus in an environment. She takes an action depending on her state, her environment, and the stimulus.

A scenario expresses state transitions of a learner along with her actions. We can understand meanings, functions, purposes of actions with states, stimuli, and environments.
[19] described in a scenario. A context is a chain of states along with stimuli and environments, which is visualized with figure 3.1. Similarly, a learning behavior corresponds to a series of actions along with a state transition.

A context depends on motivation and learning strategies of the learner. MSLQ works as a good guideline to interpret motivation and learning strategies. We can traverse causal relationships of actions in a context. When the teachers see the scenario, they assume motivation and learning strategies which explains state changes and actions in a context. They verify the assumed ones in others. Since the context expresses how her state changes, the teachers can identify motivation and learning strategies which results in the state changes, referring to the guideline. MSLQ states scales of motivation and learning strategies. It also enumerates sample questions. For each of them, MSLQ shows the strength measured by the scale for its answer. Based on the similarity with the sample questions, motivation and learning strategies implied in the context can be assessed with rough ranges such as strong, weak, and none. Suppose the following scenario is obtained through the contextual inquiry. A learner faces with a hard assignments in a practice class. She tries to ask help for a TA as her action. She also feels pleasant with encouragements from TAs. However, at that time, there is no TA available in the environment. She runs into a disappointed state. Referring to MSLQ, we can interpret the learner as an owner of a weak extrinsic component and a strong help seeking strategy, because she expects encouragements and easily gets disappointed when she fails to be helped.

Since a learner differs from others in motivation and learning strategies, she shows different actions even if she experiences same events as others do. Precise contexts and correct interpretation of them are necessary to provide suitable educational services for learners. The contextual inquiry enables us to get truthful contexts of learners, because it delves into what actually happens in learning processes.
3.3 Personas for inclusive and stable course plans

3.3.2 Personas

A large number of students in programming training are divided into several classes. Since each of the classes is supervised by a separate teacher and TAs, variance of teaching often appears among classes. It should be avoided. Meanwhile, each class has great diversity because there are various learners. Course designers should minimize the variance as well as cover the diversity.

As Cooper states, scenarios of learners similar in their characteristics can be compiled into a scenario for a persona [10]. A persona is a virtual learner with a scenario stating its behaviors in specific contexts. It represents learning behaviors of a group of learners similar in motivation and learning strategies.

Using a persona with its scenario, the teachers help teachers in individual classes grasp a single image of members in a learner group, which contributes to elimination of the variance among classes.

Instead of taking care for many individual learners, course designers can concentrate on managing a small set of personas. An inclusive course can be designed, taking care of personas dominant in the course.

A persona enables to predict reactions they take on specific stimuli in a specific environment. Course designers can choose appropriate educational services for each learner group. Owing to the prediction with personas, they can determine all educational services covering learner diversity, before the course starts. Teachers can conduct a stable course plan throughout the course. Instead of following development of individual learner, the teachers cover all personas. Suppose a class consists of several personas including $P_w$ and $P_f$. $P_w$ learns programming because it is a compulsory subject. If programming problems seem easy, he engages in the learning, otherwise he tends to set the problems aside. Namely, $P_w$ has a strong affective component and a weak self-regulation strategy. On the other hand, since $P_f$ has extrinsic components, he works well when he gets a praise from
3.3 Personas for inclusive and stable course plans

TA. At one time of the course, with supervisions of TAs, a learner who used to belong to $P_w$ may have proceeded to $P_f$. A learner of $P_f$ may fall into $P_w$ because of no care from TAs. As each learner is constantly changing his characteristics, he may belong to different personas when the time proceeds. However, if the whole set of personas are involved in the design of the course plan, the course would be stable against the change of learners.

From empirical analysis in teaching, our research assumes that the set of personas covering almost whole learners in an educational institute such as a university remain unchanged for a few years. The contextual inquiry is conducted on learners who have experienced the programming course.

3.3.3 Enforcement with discipline

To give services suitable for the learning, the teachers should grasp all of the learning contexts a persona stays in. However, it is difficult because there are too many learning contexts.

The number of learning contexts can be reduced, directing the learning of all personas toward the course goal. As explained in 3.2.1, self-regulated learners direct themselves to the course goal [68].

The elimination of learning contexts out of the course goal makes it possible to design an inclusive course plan. For example, low motivated learners want to extend their codes submission as late as possible. If deadlines are 2 week while new assignments come out every week, they would engage in assignments 1 week behind. They would lose chances to acquire new skills provided for them every week. Self-regulated learners would submit their codes within 1 week, to catch up the course progress.

A discipline is necessary to enforce all learners behave as self-regulated ones do. Controlling environments and stimuli, all learners should be directed to the course goal. The PMD method regards a discipline as self-regulation strategies of high achievement learn-
ers. Referring to a scenario of every persona, teachers find affective components of every persona, such as tests and deadlines. They make the course settings which integrate the affective components, to form enforcing services.

Suppose the discipline is solving all assignments early. To enforce it on a persona who minds scores, the educators should announce that solving assignments after deadlines would receive no score. The enforcement would make the persona try to solve all assignments within the deadlines. To impose the discipline to another persona anxious of tests, the educators should often test their understanding during the course. Since they mind tests, they would try to solve all assignments before the tests.

Learning strategies to achieve high results vary with training conditions. The teachers should determine a discipline from actual learning behavior in a target training condition.

Assuming personas in an educational institute do not change for a few years, the PMD method extracts a discipline from self-regulation strategies of high achievement learners who have achieved good results in previous courses.

### 3.3.4 Facilitation with motivation

Enforcement alone does not work well to make learners engage in studying programming. As pointed out in the ARCS model [30], motivation of learners tends to decrease along the course. The teachers have to get their attention to improve task value components, and show the relevance to enhance intrinsic motivation components. They create a good cycle of the self-efficacy in each persona for the confidence, which satisfies the persona. Since a scenario of each persona indicates its motivation components, they can choose appropriate services for every persona.

Suppose a persona is weak in intrinsic motivation, but strong in anxiety of bad scores. With the enforcement of solving all assignments within one-week hard deadline, the persona would start to work because of the anxiety. However, since he has little will to learn,
he would not be willing to solve any problem. The course plan should prepare services where supervisions to overcome problems are timely given to them. Since the persona has little confidence, face-to-face supervisions of TAs would be effective. Once he can overcome the problems, the confidence in solving problems would be improved in him. After his ability gets improved, the course should not only give face-to-face supervisions services to support him to solve problems, but also give him promoting services to encourage him in learning more, such as feedbacks showing his learning improvements.

### 3.3.5 PMD method

To address training of programming for large number of learners supervised by many teachers and TAs, the PMD method aims at achieving an inclusive and stable course plan. The course plan must be set up before the course starts. Moreover, major services should not be changed throughout the course.

A persona along with its scenario models a group of target learners. It enables teachers to predict behaviors of learners corresponding to it. A discipline is an external factor to make learner engage in the training. Course designers enforce a discipline on all learners to direct their learning toward the course goal. Motivation represents an internal factor for learners to engage in training of their own accord. Course designers choose services for each persona from motivation stated in its scenario.

Expecting high achievement learners regulate themselves toward the course goal, the PMD method identifies a discipline from their progressive leaning behaviors. Since teachers can predict learning behaviors of every persona from its scenario, they can choose course settings providing an environment and stimuli to force all learners to behave as if they have the discipline. The course settings direct all learners toward the course goal. The direction eliminates their contexts out of the course goal, as presented in figure 3.2. For example, a discipline making learners work on many assignments release teachers from
3.4 Steps to realize PMD method

The PMD method consists of five steps as shown in figure 3.3: identifying personas from the contextual inquiry, finding a discipline from learning behaviors of high achievement learners, choosing services combined with affective components to enforce the discipline, choosing services to facilitate the learning of every persona, and integrating the services to achieve a feasible course plan.

3.4.1 Identifying Personas

The contextual inquiry method is used to prepare qualified data expressing learning contexts. To get qualified contexts, the interviews are conducted on more than 50 learners.
3.4 Steps to realize PMD method

The learners having finished a programming course conduct interviews with each other to describe their learning behaviors. Since learners have to make a lot of efforts in the programming course, many complains would fill up the interviews just after the course. Interviews after a too long break will lose truthful contexts. In the PMD method, interviews are conducted after 1 semester break from the course.

Before the interviews, lectures are conducted to teach the learners what are contexts. The lectures also explain the purpose of interviews is to know contexts where learners gain or lose their vigor to study programming. The lecture emphasizes the procedure of the contextual inquiry obliges interviewers to delve into the details of the learning behaviors mentioned in the answers.

For example, suppose an interviewee mentions he solves all assignments within the week. The interviewer asks why he takes the learning strategy. He explains assignments in every week are oriented for a specific programming skill. By solving all of them, he can
acquire the programming skill, which is necessary to solve assignments in the following weeks. He also wants to avoid solving assignments of 2 weeks at the same time. The interviewer delves in a reason of his enthusiasm for acquiring all kinds of programming skill. His answer reveals he wants to be a game programmer.

Using interviews from many learners, personas are determined according to the following procedure.

1. Actual contexts obtained in each interview is summarized as a scenario. A scenario shows learning behaviors of a learner in specific contexts.

2. Each scenario is analyzed manually to identify motivation and learning strategies based on the scales explained in MSLQ [44]. Scenarios are classified into groups based on the similarity of motivation and learning strategies.

3. A representative learner is determined in each group as who best represents characteristics of motivation and learning strategies in the group. The scenario of the learner is referred to as a representative scenario.

4. In each group, significant learning contexts in scenarios other than the representative one are appended to the representative scenario, to be compiled into as a hybrid scenario.

5. Based on the hybrid scenario, a persona is determined as a virtual learner who characterizes motivation and learning strategies of the learner group. Course designers manually determine the strength of the association of the persona with each kind of motivation, referring question examples and their scale values in MSLQ. The strength is assessed in rough ranges such as strong, weak, and none.

The procedure reveals how many personas exist, what motivation and learning strategies each persona has.


3.4.2 Finding Discipline

A persona characterizes motivations and learning strategies in a group of actual learners. We do not always find a persona indicating efficient learning strategies to attain the course goal, because the classification of personas has no relationships with the efficiency. Meanwhile, learners of good achievements in past courses are expected to own efficient learning strategies to achieve the course goal.

The progressive learning behaviors express how the best learners proceed their learning. To find the discipline, efficient learning strategies are figured out from their progressive learning behaviors. Assuming a normal distribution, the second step of the PMD method focuses on the best 17% learners, who are better than the average by more than one standard deviation based on the score of assignments and tests.

The analysis of the progressive learning behaviors of high achievement learners enables teachers to make assumptions on what are their learning strategies. Among many learning strategies, self-regulation strategies are esteemed in the PMD method, because they are important in on-line courses [68]. The teachers pick up several self-regulation strategies assumed to be used by high achievement learners. They examines whether each of the strategies appears in the progressive learning behaviors, to confirm high achievement learners have used it. They adopt a set of confirmed self-regulation strategies as a discipline.

For example, suppose a programming course consists of weekly classes. The objective of earlier weeks is prerequisites for later weeks. Let the setting of the course as follows. Three difficulty levels of assignments are given to the learners every week: easy, intermediate, and hard ones. The course requirement for credit is to solve intermediate assignments within a two-week deadline. Easy assignments help some learners fill in understanding gaps. Hard ones are prepared to satisfy other learners with challenging minds. High achievement learners would regulate themselves so that they solve all assignments within
one week, in spite of the two-week deadline. The regulation avoids duplicated burdens of assignments in the week and new ones coming in the succeeding week. They also seem to know that solving only intermediate assignments does not enhance their programming ability.

Judging from their learning behaviors, teachers make a discipline so that learners should finish all assignments within one week.

3.4.3 Choosing services containing affective components

In the third step, teachers seek for affective components to enforce the discipline, using the progressive learning behaviors. They utilize anxiety of learners to make learners behave according to their intention. The anxiety may come from affective components such as a deadline, a test, a check, and so on. Once the affective components for the discipline are identified, the teachers find out services containing the affective components.

Suppose a discipline is to make learners work for longer time. Many affective components are available, each of which makes a specific persona study. For example, to personas afraid of the checks, the teachers can adopt interactive checks, where supervisors question learners on their codes in a face-to-face mode to confirm their understanding. To those afraid of tests, the teachers should conduct frequent tests during the course. Let us consider another discipline aiming that the learners should solve all assignments early to acquire enough programming skills for successive weeks, as mentioned in 3.4.2. The course designers need to identify what factors work well to make learners of a specific persona finish all assignments before the succeeding week comes. If hard deadlines are found to be an affective component to the persona, assignments with hard deadlines should be included in the course plan. It is effective to prevent the persona from postponing their submissions.
3.4.4 Choosing facilitating services

This step aims at enumerating service candidates to facilitate the learning of personas. Course designers choose services to facilitate each persona to keep the discipline, from the correspondence of motivation components of every persona to motivation effects of each service.

The first task in this step examines motivation effects of each service. For a service used in previous courses, the teachers confirm how it contributes to motivating learners, using the progressive learning behaviors.

Let us consider an interactive check, where a TA checks a source code in a face-to-face manner with a learner. The TA together with the learner confirms what points the code has achieved, and what points the code has not. The TA supervises the learner to achieve a good code. Consequently, the learner would improve knowledge and skill after careful supervisions. In other words, interactive checks effective for learners owning extrinsic components improve their intrinsic and expectancy components. To confirm the effects of the interactive check, we can examine how the learner behaves after he or she receives the interactive check: whether she takes another interactive check, whether she tries to solve more assignments, whether she improves her score, and so on.

Sometimes, learners request a new service in interviews. Since course designers have no progressive learning behaviors on it, they have no other way to estimate its motivation effects. Suppose graphics assignments as new motivation components. They are expected to have association with task value effect, because they appeal the learners with the visual execution results.

In the second task, the teachers choose facilitating services for each persona. The association strength of each persona with motivation components is interpreted from scenarios as explained in 3.4.1.
3.4 Steps to realize PMD method

The teachers choose services which are incident with the persona in terms of motivation aspects, to determine facilitating service candidates for the course plan. At this time, they give priority to ones whose motivation effects are confirmed in the first task. An example is a help-seeking persona who has extrinsic and affective components. Suppose a scenario shows that, a persona prefers interactive checks and pair programming. Suppose interactive checks are confirmed to be associated with extrinsic components, while pair programming is only estimated to be, in the first task. Since interactive checks are associated with the help-seeking persona in terms of extrinsic components, they are included in the service candidates for the persona. Pair programming also joins the service candidates, but it has a lower priority than interactive checks.

3.4.5 Integrating services into feasible course plan

An actual course may not include all service candidates because of constraints such as the cost or the quantity of supervisors. Course designers should build a feasible course plan, considering the constraints. Some services may cost too much to provide for all learners, even though they are effective. Course designers should provide the services, giving high priorities on personas they want to improve most.

At the same time, the teachers provide other services to promote learners who fail to receive the costly services, considering contexts where they need the services. Contexts acquired from the contextual inquiry play vital rolls in the consideration. For example, interactive checks for all learners are infeasible because of the limitation of human resource such as teachers and TAs. Even if interactive checks are confirmed to be good for many personas, course designers have to reduce the number of learners using the service. It is one way to eliminate advanced learners from a target group of the service. Suppose the contextual inquiry shows that the learners want to get not knowledge but praises from TAs in interactive checks. Reflecting the context, the teachers substitute interactive checks with a ranking service.
3.5 An example of PMD method

We present experiences of application of the PMD method upon the condition of Ritsumeikan university, Japan. The course plan in 2010 is improved into ones in 2011 and 2012.

3.5.1 Training condition

Every academic year, about 500 freshmen enrol the mandatory introductory programming practice course. The total number of students in 2010, 2011, and 2012 are 474, 546, and 557, respectively. The lecture course on C language specification is synchronized with the programming practice course. In the lecture course, students are divided into 2 classes whose teachers have common materials and tight communication for identical teaching. Meanwhile, they are divided into 10 classes in the practice course. Each class is supervised by 1 teacher and 10 TAs, who spent almost whole class time for individual supervision and source code checks, because they are indispensable in programming practice.

Table 3.1 shows the course schedule, which does not include the 1st, the 2nd, the 10th and 15th week. They are used for Linux practices, emacs practices, a mid-term test, and an end-term test, respectively. The schedule is identical in the three years.

The course plan in 2010 is the followings. Every week, one regular practice class and

<table>
<thead>
<tr>
<th>Week</th>
<th>Content</th>
<th>Week</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>variable &amp; expression</td>
<td>9</td>
<td>function with array</td>
</tr>
<tr>
<td>4</td>
<td>conditional statement</td>
<td>11</td>
<td>string</td>
</tr>
<tr>
<td>5</td>
<td>loop statement</td>
<td>12</td>
<td>string with pointer</td>
</tr>
<tr>
<td>6</td>
<td>nested loop</td>
<td>13</td>
<td>structure</td>
</tr>
<tr>
<td>7</td>
<td>function call</td>
<td>14</td>
<td>recursive call</td>
</tr>
<tr>
<td>8</td>
<td>array</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Course schedule
two complementary ones, 90 minutes for each, are arranged for all of the students to practice under supervisions of TAs and teachers. Assignments of every week are arranged into three difficulty levels: easy, medium, and difficult. The total score given for all assignments of a week is 130, in which the easy ones, the medium ones, and the difficult ones take 30, 60, and 40, respectively. The students can submit source codes for the easy and difficult assignments any time in the semester, while two-week soft deadlines was obliged to the medium ones. If a source code for a medium assignment is submitted later than 2 weeks, the submission is accepted but the score of the code is deducted 10% every one week delay. The assignments and some sample codes are given to the students via a Web site though which the students can submit the source codes at any time from any place. The regular classes take attendance into account while the complementary ones do not. To help students overcome initial programming difficulties, the teachers and TAs give interactive checks on all source codes submitted before the mid-term test.

To the codes submitted after the mid-term test, they do not give any interactive check, but off-line checks.

### 3.5.2 Personas from contextual inquiry

We extracted personas in the first step. We conducted the contextual inquiry to get summary learning behaviors. The following shows a part of an interview example of student R (interviewer) and student E (interviewee).

*R:* What makes you inclined to programming?

*E:* I regard programming as a challenge. When I have solved a tough assignment, I obtain a strong sense of achievement.

*R:* When you face a tough one, you might sometimes find no way to solve it. Have you ever run into such a situation?

*E:* Yes, many times. In such a case, I will repeat to check sample codes in the textbook. Sometime I try easier assignments in the same section.
R: What will you do, if you cannot get anything from them?
E: I will search Web pages explaining similar matters using the Internet. I do not prefer to be supervised by TAs, because I feel lost in the challenge.
R: But, they give you hints, even an encouragement sometimes.
E: I am not pleased, even if they encourage me. But, when they give high grades to my codes, I get satisfactions.

The scenario below is described from the interview.
“Mary regards the programming as a challenge. She likes programming because she gets a strong sense of achievement when she has solved hard assignments. When she cannot solve them, she repeats to check sample codes or tries easier assignment in the same section of the textbook. If she cannot get anything from them, she tries to find explanation of similar matters with the Internet. Because of her pride, she seldom asks TAs for supervision. Their encouragements do not please her, either. In the mean time, high grading for her codes from TAs satisfies her.”

The scenario reveals the context of facing tough assignments. In the context, student E has shown her attitude to address assignments. She also explains ways to overcome them, which is utilizing information from the textbook and the Internet. The attitude implies her intrinsic component, while searching for information from textbooks and the Internet indicates her learning strategies. The context also reveals two resources, sample codes and explanations, which are hints to prepare services for the course plan.

To build a course plan of 2011, we have got scenarios from 73 students. We have found 3 groups of students similar in motivation and learning strategies. The groups remain unchanged in the analysis for 2012. One group consists of 27 students willing to learn. They understand the more they engage in the study, the more they can get knowledge and skills. Some of them want their high ability to be praised by supervisors or friends. This group, represented by persona P1, is specified to have strong intrinsic, expectancy, and task value components, along with weak extrinsic ones. Moreover, P1
knows how they should study to improve their abilities. They know to utilize various resources such as explanations by TAs and sample codes in the Internet. It reveals that P1 has learning strategies to achieve a good result.

The second group consists of 19 students who understand the compulsory programming subject brings many benefits to them. Their learning purpose is to get a credit. Their attitudes to the learning are not stable. As far as they can go on with assignments easily, they are eager to study. But, for assignments which seem hard to solve, they tend to put them aside, because they do not believe their ability. They often stick to specific resources such as TA helps. They do not know utilizing many kinds of resources. We refer to this group as persona P2. We regard P2 has weak intrinsic, and strong extrinsic, weak task value, and strong affective components. P2 has no learning strategy.

The third group, represented with persona P3, consists of 27 students who think that they are obliged to learn programming. They do not like to learn but have to, because of the compulsory subject. They learn only at school with help of supervisors. Different from P1 and P2, P3 does not show any will to learn. P3 has no intrinsic component. Since P3 minds obligations so much, P3 has strong affective component and weak extrinsic one. P3 has no learning strategy.

![Figure 3.4: Personas from contextual inquiry](image)

Figure 3.4: Personas from contextual inquiry
Figure 3.4 shows associations of the three personas with motivation components. The solid lines show strong associations, while dashed lines correspond to weak ones. The strength of the association is manually determined when the teachers determine personas from scenarios of many learners, referring the combination of sample questions and their correlation with the final grade in MSLQ.

The figure also indicates service references of the personas in their summary learning behaviors with dotted lines. Assignments are requirements for students to make source codes. Interactive check is briefly described in section 3.4.4. Complementary classes are prepared for teachers and TAs to supervise students at a loss to solve assignments. Sample codes are model codes with detail explanations. Graphics assignments [62] enable students to view execution results visually. Ranking shows scores of top ranked students.

3.5.3 Early submissions and long learning time

The top 17% highest score students are selected according to the average score of weekly assignments. To find efficient self-regulation strategies specified in the second step of the PMD method, we examine the difference of learning behaviors of the top students from those of the remaining. Every week has its own matters to learn. The matters in the week are used in the succeeding weeks. High achievement students would know the importance of catching up the course progress. They would try to understand matters in the week for succeeding weeks. Among several indexes, we have examined the submission timing and the learning time, to know the self-regulation ability. As we have expected, they solve many assignments within the week, to master the programming knowledge and skills of the week.

Table 3.2 shows the average days from the announcement of assignments until the code submissions. As the table presents, the top ones submit their source codes much earlier than the remaining. They submit all source codes for basic assignments within 14
3.5 An example of PMD method

Table 3.2: Days until submission

<table>
<thead>
<tr>
<th>Submissions</th>
<th>Top 17%</th>
<th>Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic assignments</td>
<td>14</td>
<td>28</td>
</tr>
<tr>
<td>Intermediate assignments</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>Hard assignments</td>
<td>31</td>
<td>55</td>
</tr>
</tbody>
</table>

days, those for intermediate assignments within 6 days, and those for hard assignments within 31 days. Specially, they solve almost all intermediate assignments within 1 week, which means the submission before the assignment issue in the succeeding week.

Figure 3.5 shows the average learning time of top students (dashed line) and that of the remaining (solid line). The dashed line keeps higher than the solid one throughout the course. Every week, the top students spend more time to learn than the others do.

Both of early submissions and long learning would be important features among learning strategies of the top students. Early submissions without spending much time for coding would result in codes with bugs. On the other hand, if students spend long time on the learning without submitting codes, their learning would not be fruitful.
3.5 An example of PMD method

Table 3.3: Confirmation of motivation components of services and attributes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard assign.</td>
<td>No deadline</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>Afford</td>
</tr>
<tr>
<td>Interme. assign.</td>
<td>2-week soft d.line</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td>Afford</td>
</tr>
<tr>
<td>Graphics assign.</td>
<td>2-week soft d.line</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td>Afford</td>
</tr>
<tr>
<td>Easy assign.</td>
<td>No d.line</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td>Afford</td>
</tr>
<tr>
<td>Inter. checks</td>
<td>all assign.</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
<td>Limit</td>
</tr>
<tr>
<td>Compl. classes</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td>Limit</td>
</tr>
<tr>
<td>Mid-term test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td>Afford</td>
</tr>
<tr>
<td>End-term test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Afford</td>
</tr>
</tbody>
</table>

3.5.4 Hard deadline assignments and weekly tests

The analysis in 3.5.2 has showed P1 has self-regulation strategies, while P2 and P3 do not. We should enforce the strategies on P2 and P3 as the third step in figure 3.3. To enforce early submission and long learning time as a discipline, we have to find affective components of P2 and P3. As figure 3.5 shows, the learning time of all students increases strongly between the 13th and the 14th week in the 2010 course. The source code submissions are accepted until the end of the 14th week in 2010. It is a hard deadline. The 15th week is reserved for the end-term test. It proves that the students mind hard deadlines and the end-term test a lot. Table 3.2 also reveals the affective effects of deadlines. For no-deadline assignments, it takes long time for students to submit them. In contrast, for soft-deadline ones, the submissions happen within the deadline. We conclude hard deadlines and tests cause the students to learn.

We can expect services with these affective components work as a discipline to direct students toward the course goal. Assignments with one-week deadlines would make students solve all assignments before new assignments are issued in the succeeding week. A weekly test would make students study long to master the knowledge and skills every week. With the services, students with no learning strategy would behave as if they have the learning strategies of the top students.
3.5 An example of PMD method

3.5.5 Facilitating services

With the discipline, students of P3 would start to learn because they are afraid of the tests and the deadlines. However, they would stop learning soon because they have no intrinsic motivation. They do not want to overcome the difficulties. The course plan should have services so that TAs give supervisions in a timely fashion, not only to help them overcome the difficulties, but also to make them understand the matters. Students belonging to P2 have both intrinsic motivations and affective components. The affective components make them start to learn, but they are easily discouraged by difficult problems because of weak intrinsic motivation. The course plan should provide services with which they can acquire supervisions to solve the problems. For those who belong to P1, since they have strong intrinsic motivation and learning strategies, they can learn by themselves. The course designers should provide services to encourage and promote them.

As the fourth step of the PMD method, we examine services for each persona in figure 3.4. P3 prefers easy and intermediate assignments, interactive checks, complementary classes, and sample codes. As table 3.3 shows, all of the former three services have extrinsic components, which match motivation components of P3. Easy assignments initiate them into programming. Since P3 minds scores, they engage in intermediate assignments. Interactive checks and complementary classes provide them with on-time supervisions from TAs. All of the services are good candidates for facilitating P3. Providing sample codes is a new service. Sample codes would be helpful for P3 because they contain detail comments. For P2, interactive checks, complementary classes are also good because P2 can acquire supervisions from the two services. Sample codes are also useful for P2 to refer to. In addition to interactive checks and complementary classes, P2 indicates the preference of intermediate assignments and graphics assignments. They are chosen as candidates because the motivational effects of these services match those of P2. Similarly, hard assignments, and ranking are chosen for P1.
3.5.6  New course plan

We have listed service candidates as following: three difficulty levels of one-week hard
deadline assignments, on-line weekly tests, interactive checks, complementary classes,
graphics assignments, sample codes, and ranking.

According to the fifth step, a feasible course plan must be integrated, picking up
services from the candidates. With constraints of 1 TA over 5 students in regular and
complementary classes, we cannot give interactive checks for all source codes of many stu-
dents. We must limit the checks for certain students to make the new course plan feasible.
Students belonging to P1 and P2 would join to complementary classes on their will to get
supervisions from TAs, because they have intrinsic motivation. Students corresponding
to P3 may not ask questions to TAs when they encounter problems, because they have
no intrinsic motivation. We should oblige the checks on these students. Through the
checks, oral explanations from TAs would help them not only solve problems, but also
understand the basic points. The demonstrations to make a successful program during
the checks would increase their confidence to continue to learn. Therefore, interactive
checks are obliged only for students who fail the test of the week.

3.5.7  Results and discussions

We have conducted the programming exercise courses with course plans of 2011 (CP 2011)
and 2012 (CP 2012) designed with the PMD method. The course plan differs from the
one of 2010 in terms of

- 1 week hard deadline to submit weekly assignments,
- weekly tests,
- interactive checks for failures of every weekly test, and
3.5 An example of PMD method

- announcement of top 10 score ranking.

![Figure 3.6: Submission rate of CP 2010, CP 2011, and CP 2012](image)

Different from CP 2011, TAs are encouraged to check source codes as soon as they are submitted in CP 2012.

The progressive learning behaviors in 2011 and 2012 are compared with those in 2010 to see the effectiveness of the PMD method. First, we examine how the enforcement and the facilitation work on all of the students. Assignments are issued every week. We should check how deadlines and weekly tests improve source code submissions. Figure 3.6 shows how many source codes are submitted on each day after the announcement of the assignments. In CP 2010, the submission, presented by black bars, happens very little within 14 days. The source codes submitted within 14 days is only 37.45% of the total assignments. In contrast, in CP 2011 and 2012, which imposes the students to submit source codes within one week, the submission happens almost every day, as presented with striped bars and gray bars. The total submission rate in 2011 and 2012 are 84.05% and 87.03%, respectively. Earlier check of source codes in 2012 encourages students in earlier submission.
Figure 3.7 presents the average learning time of students of the three course plans. The students in 2011 and 2012 spend much more time than the ones in 2010.

The average learning time of students in 2010, 2011, and 2012 are 205.5 minutes, 257.0 minutes and 288.1 minutes, respectively. The average learning time of the top 15% students in 2010 is 254.5 minutes. It is amazing the average learning time in 2011 is more than that of the of the top students in 2010. It is also the case in 2012.

The dashed line corresponding to CP 2010 increases rapidly at the 13th week, which is almost the end of the semester. In the meantime, the solid line of CP 2011 and the dotted line 2012 do not increase.
3.5 An example of PMD method

Figure 3.8: Score of students of CP 2010

Figure 3.9: Score of students of CP 2011
In 2011 and 2012, the farther the course proceeds, the shorter the average learning time is, though the submission rate is higher. That means, in 2011 and 2012, it takes less time for students to solve assignments, as the course proceeds.

Finally, we compare student scores in 2010, 2011, and 2012. We classify students into 4 groups by their total scores of all weeks to examine how the score of each group changes. Figure 3.8, figure 3.9, and figure 3.10 present the average score of each group in 2010, 2011, and 2012, respectively.

In 2010, all of the lines fall down rapidly from after the mid-term test where students learn difficult matters such as array and structure. In contrast, the lines of the four groups in 2011 and 2012 keep horizontal. All of the students in CP 2011 keep working throughout the course. Because they improve programming skills, they solve assignments using less time as figure 3.7 shows. It is also noteworthy that the scores of poor students of 2011 and 2012 rise in the middle of the semester, while that in 2010 goes down rapidly right after the beginning of the semester. The result proves that the course plans revised with
the PMD method in 2011 and 2012 can prevent the learners from reducing their scores. The discipline has directed all students to the course goal.

The revised course plan gives P3 affective components with weekly tests and hard deadlines, along with supervisions of interactive checks. Since they can get explanations and demonstrations through interactive checks, they attain to solve assignments. The revised course plan gives them beliefs to programming, or expectancy components, which motivate them to learn more. For P2, the revised course plan provides learning strategy using one-week hard deadlines. Since they finish assignment within 1 week, they can get knowledge necessary to solve new assignments in the next week before the new assignments are issued. Since they can solve the new ones, they have confidence and interests in programming. To P2, the revised course plan brings expectancy components and task value components. For P1, the revised course plan does not only promote their intrinsic motivation with hard assignments, but also motivates them with extrinsic effects of ranking service. Overall, the revised course plan can enhance motivation for all of the personas, which makes them keep working throughout the course.

3.6 Conclusions

We propose the PMD method for designing an inclusive and stable introductory programming practice course that motivates all learners throughout the course. It utilizes personas representing learners with similar motivation and learning strategies. The personas are determined from summary learning behaviors taken with the contextual inquiry. The method enforces a discipline on all personas to direct their learning to the course goal. The teachers choose services to facilitate each persona under the enforcement. The method has enlarged the learning time and improved their scores in the successive 2 courses of introductory programming practice for over 500 freshmen in each course in Ritsumeikan university.
3.6 Conclusions
Chapter 4

Association of internal factors with behavior factors

4.1 Introduction

In the introductory programming course, students are required to understand various abstract concepts. They also have to master skills to write codes. In the programming education in high educational institutes like universities, a few teachers supervise large number of students. The environment makes it difficult to provide sufficient supervision for the students suffering from difficulties in programming. If the students fail to get supervisions at the difficulties, they would get desperate, because they are at loss what to do. The lack of timely supervisions easily deprives their ambitions to learn programming.

Once students lose their ambitions to learn programming, its recovery is quite difficult, because the programming topics to be learned in a course are arranged in a consistent fashion, where basic ones precede complex ones to be understood based on them. During programming courses, teachers should monitor whether students maintain their inspiration and will to learn programming. However, the monitoring is infeasible, because
teachers are too few to pay attention to whole students. We need a tool to automate the monitoring of learning status of students during the course.

Many institutes use learning portfolios expressing learning behavior. The behavior involves what kinds of assignment a student solves, when they submit assignments, and the score the student earns. The behavior is extracted from the student learning logs taken through support learning web sites. However, it is difficult for teachers to judge whether the student is stuck in the learning, only from portfolios. Most of stuck students are similar in many behavior factors, but they have their own reasons resulting in the behavior factors. The reasons are very different, depending on individual students. The teachers should understand the reasons to provide suitable supervisions [22].

To understand the reasons, the teachers should examine the student learning status, which is expressed by factors of their motivation and learning strategies. To know the learning status, conventional methods conduct interviews or questionnaires based on motivation and learning strategies [32]. However, we cannot expect these methods to bring a practical solution. It will cost tremendously if we conducted interviews for all students. Repeated questionnaires during a course annoy students. Students tired of many questionnaires submit no serious answers, which prevents teachers from getting truthful data. It is necessary to know learning status of students from their usual learning activities without annoying them and at low costs.

Experienced teachers could know learning status of students from fine records of their learning behavior. Nevertheless, they should focus on a few students who are likely to lose learning ambitions for programming. A way to find current students similar to past students who have lost learning ambitions would bring a promising solution to predict the learning status of current students. The similarity calculated with records of learning behavior of both students enables teachers to focus on students with signs of losing ambitions.
Let us consider a method to transform behavior factors into the internal factors constituting motivation and learning strategies to distinguish students to be cared from others. There are many students taking a course. Each student has many behavior factors. We transform behavior factors of the students into internal factors. If students of similar motivation and learning strategies study under the same course settings, the students would tend to behave in a similar way.

Non-negative matrix factorization (NMF) [13] would be a good mathematical way to achieve the transformation. NMF decomposes a matrix representing behavior factors of students into 2 matrices. One represents associations of internal factors with each student. The other represents associations of every internal factor with behavior factors. It is important to examine correspondence of these two kinds of associations, for success of the decomposition.

This chapter explains a method to figure out the 2 kinds of association weekly. It figures out internal factor values of freshmen in actual programming courses in a university. It compares the values in the matrices of successive years. An educational institute like a university usually admits students at a specific range in the degree of intelligence. When students of an institute are compared in different years, the proposed method finds many similar students in terms of internal factors. It suggests that internal factors make it possible to immediately distinguish learning status of current students with characteristic factors of past students.

4.2 Formative assessment

4.2.1 Formative assessment process

The learning of students must be assessed frequently for two main purposes. One is to understand the student learning status. The assessment brings teaching staffs about what
extent their students master the learning topics. The assessment also brings them about feeling and attitude of the students toward the learning. It is called the assessment of learning [1]. The other is the assessment for supervision. After the assessment of learning, the teaching staff would adapt their instructions to the student status for achieving the course goal.

Since formative assessment aims at frequent evaluation of the learning, we should prepare well the process of the assessment, including obtaining the information of student learning, in advance. The process would take as little time as possible when being conducted because the course is ongoing. The process should also not be labor intensive so that it can be executed easily.

Common methods prepare interviews or questionnaires to obtain learning information from students [1], that is, for the former purpose. Even though these methods are widely used, we cannot expect them to bring a practical solution. It would costs tremendously if we conducted interviews for all students. Repeated questionnaires during a course annoy students. Students tired of many questionnaires submit no serious answers, which prevents teachers from getting truthful data.

Schmoker [52] proposed a data-driven decision-making model for the latter purpose of formative assessment. They construct a data team consisting of educators who teach the same content standards to their students. The data team has meetings regularly to express purposes of analyzing the data. The data team applies the five-step assessments: charts the data, analyzes the results, sets the goal, selects effective teaching strategies, and determines the result indicators. In actual education, it would be difficult to apply the method because of its time-consuming and labor intensive process.
4.2.2 Formative assessment in programming education

Students take various kinds of learning behavior under a given learning environment. It includes how long they engage in studying of a specific subject, when they submit home works, what kind of supplemental material they refer to, and so on. A programming course often facilitates learning supporting web sites where records of learning behaviors can be taken as learning logs using scripts on web browsers.

The learning logs express learning behavior of students truthfully since students almost completely have no sense of the logging. The logging causes no burden to students. The logging brings about behavioral data along the course without any effort of teaching staffs.

Conventional methods have adopted behavior factors [25, 38, 31, 37] as measures to know the learning of students. With the behavior logs, instructors try to direct as many as students with course settings, aiming that students acquire programming skills. The instructional design methods [15, 50] address the issue. The ARCS model proposed by Keller [30] is also considered to lead students to participation in learning with course settings which brings attention, relevance, confidence, and satisfaction.

Even though behavioral data are easily recorded, and it expresses student learning truthfully, it is still a big issue how teaching staffs interpret it so that they can understand the learning of their students. For example, suppose an excellent student who spends short time to learn and achieves good score for a week. If teaching staffs consider him as a student without need of care, they may ignore the other students with the similar behavior, who also spend short time for learning but high score because of copied codes. If teaching staffs examine only some behavior factors, it is hard to assess student learning correctly. They should examine very many kinds of behavior factors simultaneously.

Learning behavior of a student varies according to learning topics and course settings. Assume a student is a good student. He spends long learning time and earns good score in a week, because he understands the learning topics well. But, he takes short learning time
in another week because of some reasons, which prevents him from high score in the week. He might give up early, because he loses his motivation for the learning topic which is too hard to him. It might happen because he cannot find any TA who presents convincing advice. Teaching staff should consider various behavior factors under relationships with learning topics and course settings every week for the judgement.

4.3 Formative assessment based on internal factors

4.3.1 Persona types

As described in section 3.3.2 and section 3.4.1, a group of students similar in motivation and learning strategies are represented with a persona [10]. To examine the personas of the students taking the programming course, the proposed method analyzes results of the contextual inquiry of past students of a similar course, from the viewpoints of factors constituting motivation and learning strategies. From student requirements revealed in the contextual inquiry, personas are divided into 2 categories: active type and passive type. Active persona students can study by themselves. On the other hand, the passive ones cannot proceed with their learning by themselves. It is necessary for teaching staffs to give supervisions to passive persona students timely to prevent them from losing willingness for learning.

In our experience with the programing course of Ritsumeikan University in 2012, we found 5 kinds of personas shown in table 4.1 from student contextual inquiries. Easy going persona and industrious persona correspond to P1 and P2, described in section 3.5.2, respectively. The demanding one, the obliged one and the unwilling one correspond to P3. Industrious and easy-going personas are included in the active personas, while the passive personas consist of the remaining ones. The status of the learning of each persona is represented with 5 motivation factors and 3 learning strategies factors. The motivation
4.3 Formative assessment based on internal factors

factors include intrinsic, extrinsic, task value, self-efficacy, and affective ones. The learning strategy factors consist of effort regulation, help seeking, and meta-recognition.

<table>
<thead>
<tr>
<th>Persona</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industrious persona</td>
<td>has a strong will to improve programming ability.</td>
</tr>
<tr>
<td>Easy-going persona</td>
<td>studies programming as far as it is a fun.</td>
</tr>
<tr>
<td>Demanding persona</td>
<td>wants a preferable learning environment prepared.</td>
</tr>
<tr>
<td>Obliged persona</td>
<td>engages in programming under a pressure all the time.</td>
</tr>
<tr>
<td>Unwilling persona</td>
<td>hates programming learning</td>
</tr>
</tbody>
</table>

### 4.3.2 Persona judgement

Programing courses in universities often have many students but few teaching staffs. One teacher and some TAs have to care many students at the same time. It is difficult for them to supervise all of the students. The teaching staffs should focus their power on ones who cannot proceed with their learning. Passive students should be taken care timely. On the other hand, teaching staffs should leave ones who can learn by themselves to encourage their spontaneous learning. Since every student changes the learning status during the course, teaching staffs should determine the supervision to each student through the formative assessment of the student.

We define personas from student scenarios, which specify behavior of the students in specific learning contexts. A persona represents students whose behavior in their scenarios is similar in terms of their motivation and learning strategies. Each persona represents a group of students similar in motivation and learning strategies. It implies their behavior in a specific learning context. Once we can find students matching a persona, we can predict learning behavior of the students in specific learning contexts. Since programming courses usually change learning topics weekly, learning status of each students also changes weekly. Therefore, as the formative assessment, we target at judging the persona of every student every week.
Motivation and learning strategies include many internal factors, which explain the status of the student learning comprehensively. For the judgement of the persona of every student, it is essential to examine motivation and learning strategies of the student. For example, the intrinsic factor explains the extent of desire, interest, and discovery of the student toward the subject. If a student has a strong intrinsic factor, he would be very interested in the subject. Otherwise, he would not be interested in the learning much. In the case the student has strong intrinsic motivation, the teaching staff can leave him to study by himself. In the meanwhile, in the case the student is weak in intrinsic motivation, the teaching staff would adapt the supervisions to the student to increase his desire to learn.

A persona is determined by the motivation and the learning strategies. To identify the student persona, teaching staffs need to know various factors of motivation and learning strategies for the student, not limited to the intrinsic factor. Once they identify the persona of a student, they can accommodate their supervision for the student according to the persona.

### 4.3.3 Internal factor based NMF transformation

The method aims at figuring out the persona for every student of an on-going course to determine the supervision of a teaching staff for him. Learning behavior of a student can be obtained with a web site which supports their learning providing various kinds of services. We should develop a way to transform his learning behavior into a persona. The study considers two ways of transformation as described in figure 4.1. The first way is to transform behavior factors into persona directly. The other is to transform them into internal factors, succeeded by the identification of personas of students with their internal factors.

There are many students taking a course which provides specific course settings. A
student involves many behavior factors. We can present students with their behavior factors using a matrix, called B, where the rows are students and the columns are behavior factors. Non-negative matrix factorization (NMF) decomposes a given matrix into the product of two matrices, using the hill climbing algorithm [24]. It would be a good mathematical model to decompose matrix B in both of the cases, as described in part a and part b, respectively, of figure 4.2.

For the first case, we should find weights for every student to belong to individual personas. Therefore, we can use another matrix, called P, to represent a target matrix. Its rows are students, while its columns are personas. Each of its elements indicates a persona weight, the degree of each student belongs to specific personas. Values of behavior factors as well as persona weights of every student would be non-negative.

For the second case, we should find weights of individual factors of motivation and learning strategies for every student. Suppose a matrix, called W, to represent the degree each student has specific internal factors. All elements in matrix W would also be non-negative.

In NMF, rows and columns of matrices represent specific items. In order to achieve good decomposition, it is important to consider association between items. Let us take figure 4.1 into account, again. We should pay attention to the causal relationships in it. We determine personas from internal factors in the analysis of scenarios. In addition to
4.3 Formative assessment based on internal factors

that, it is considered internal factors cause students to behave in specific ways in given learning contexts.

Assume persona vectors which represent all internal factors of individual personas. In case of the transformation to personas, we must estimate associations of students with personas in matrix \( P \), and associations of personas with behavior factors in matrix \( G_i \). For the former estimation, we need to perform two steps. The first step is finding internal factors of each student. We construct the internal vector of the student from the result. The next one is calculating the similarity of the internal vector to the each of persona vectors to figure out the persona weights. The second step shows the estimation of persona weights for a student consisting of more than one calculation procedures to achieve the
4.4 Figuring out actual associations

transformation. After it calculates the similarity of the internal vector to the persona vector, it again calculates the weight to each persona using the similarity. As the results, the estimation would not be truthful, or much error prone. As the above discussion on figure 4.1 states, internal factors not only determine personas, but also cause students to behave. The estimation of effect of personas on behavior factors would be weaker than the estimation of effects of internal factors on behavior factors.

The above analysis reveals it is essential to estimate internal factors for every student. As shown in figure 4.1, they are internal factors that affect on each behavior factor directly. We consider transforming learning behavior into internal factors instead of personas. If we know internal factors of a student, we can understand his persona. More than that, we will understand his learning status comprehensively. Therefore, we take into account the transformation into internal factors, described in part b of figure 4.2.

4.4 Figuring out actual associations

The subsection above explains our prediction that the estimation by internal factors is more promising than that by personas. For the formative assessment using personas of students, the former estimation identifies personas of students using their internal factors, after it calculates internal factors from behavior factors. To justify the prediction, we should examine both of the association of internal factors with personas and the association of internal factors with behavior factors.

4.4.1 Behavior factor values to be analyzed

Web 2.0 technologies facilitate us to collect operation logs of students on learning support servers. The logs express interactions of learners with the servers through Web pages. They compose a student portfolio, which expresses the student learning behavior. From
the logs, we can figure out real values of behavior factors. Table 4.2 shows an example of behavior factors and their values in a week:

Table 4.2: Example of behavior factors

<table>
<thead>
<tr>
<th>Sign</th>
<th>Explanation</th>
<th>Sample value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>total length of periods student logs in to server in a week</td>
<td>325 (min)</td>
</tr>
<tr>
<td>$i$</td>
<td>times that student logs in to server</td>
<td>5 (times)</td>
</tr>
<tr>
<td>$a$</td>
<td>the number of submitted assignments of advanced level</td>
<td>3 (assignments)</td>
</tr>
<tr>
<td>$z$</td>
<td>the number of all submitted assignments</td>
<td>8 (assignments)</td>
</tr>
<tr>
<td>$e$</td>
<td>the number of submitted assignments before last day</td>
<td>6 (assignments)</td>
</tr>
<tr>
<td>$l$</td>
<td>the number of submitted assignments on last day</td>
<td>2 (assignments)</td>
</tr>
<tr>
<td>$s$</td>
<td>total score</td>
<td>76 (point)</td>
</tr>
<tr>
<td>$c$</td>
<td>total number of any button clicked on exercise site</td>
<td>533 (times)</td>
</tr>
<tr>
<td>$v$</td>
<td>score of assignments of advanced level</td>
<td>28 (point)</td>
</tr>
<tr>
<td>$p$</td>
<td>the number of times student visit progress page</td>
<td>3 (times)</td>
</tr>
</tbody>
</table>

The learning time of a student in a week, denoted by symbol $t$, is calculated as the sum of his or her login session time into the Web site in the week. Symbol $i$ stands for the number of times the student logs in. Symbol $a$ represents the number of advanced assignments the student submits. There is a deadline for submission of assignments every week. Symbol $e$ shows the number of assignments the student submits on the last day. Symbol $c$ denotes how many times the student clicks items inside the pages of the Web site in the week. Symbol $s$ indicates the score the teachers or TAs give to the students after evaluation. Values of all factors are normalized into a uniform range [0,1].

### 4.4.2 Associations using vectors

Students having finished the programming course have real number vectors, which are composed of behavior factors. Each behavior factor in the real number vector is retrieved from portfolios of the students during the course. The real number vector is referred to as a behavior vector. Once the contextual inquiry is applied to the students, they have another kind of real number vectors, which consists of internal factors. They are referred to as internal vectors. In addition to that, a persona each student belongs to is determined
4.4 Figuring out actual associations

through the analysis of his scenario. The study presented in the thesis figures out 2 kinds of association founding on the real number vectors and personas.

Let us first focus on the association of personas with internal factors. The method proposed in the study calculates the mean, $\mu$, and the standard deviation, $\sigma$, for each internal factor. Some elements have large values, while other has small values. To treat them in a fair way, the element value, $v$, is converted into the $Z$-score, $z$, which is calculated with $z = \frac{v - \mu}{\sigma}$. The method calculates the $Z$-score for all internal factors for every student. For all students belonging to each persona, the method figures out a vector consisting of the mean of $Z$-scores for every internal factor. Along with the mean, the standard deviation is also calculated for every internal factor. The vector composed of the mean of $Z$-scores represents the internal factors of the persona. Especially, small standard deviation indicates most students in the persona have similar value for the internal factor. The internal factor with small standard deviation is characteristic to the persona.

Next, we consider the association of internal factors with behavior factors. Using the internal vectors of students, we can pick up students who have high values for a specific internal factor. For every internal factor, the method calculates the mean of behavior factor values of students picked up. The vector composed of those mean values corresponds to a group of students who are strong in a specific internal factor. It represents the behavior factor values of those students. In other words, the vector implies learning behavior of students with strong values for the internal factor.

4.4.3 Associations of internal factors with personas

We collected learning behavior of students in the introductory C programming course of Ritsumeikan University in 2012. We conducted the contextual inquiry for 42 students under their consent. In the experiment, we focus on 8 kinds of internal factors; they are intrinsic, extrinsic, task value, self-efficacy, and affective factors for motivation, while
effort regulation, help seeking, and meta-cognitive factors for learning strategies [44].

Two teachers read scenarios obtained in the contextual inquiry. To remove personal bias of the teachers, they individually classified students into 5 kinds of personas explained in Table 4.1. When their decision is not coincident, they made an agreement for the decision, exchanging their opinions. The numbers of students who are demanding, easy going, industrious, obliged, and unwilling personas is 5, 10, 15, 6, and 6, respectively.

<table>
<thead>
<tr>
<th>persona</th>
<th>intri.</th>
<th>extri.</th>
<th>task v.</th>
<th>self-e.</th>
<th>affec.</th>
<th>effo re.</th>
<th>help se.</th>
<th>meta-re.</th>
</tr>
</thead>
<tbody>
<tr>
<td>demanding</td>
<td>4.89</td>
<td>7.33</td>
<td>8.28</td>
<td>4.57</td>
<td>4.57</td>
<td>2.99</td>
<td>12.57</td>
<td>2.44</td>
</tr>
<tr>
<td>easy going</td>
<td>52.71</td>
<td>52.1</td>
<td>55.15</td>
<td>53.93</td>
<td>47.82</td>
<td>48.43</td>
<td>49.04</td>
<td>59.42</td>
</tr>
<tr>
<td>industrious</td>
<td>4.77</td>
<td>6.58</td>
<td>9.08</td>
<td>4.92</td>
<td>5.63</td>
<td>4.89</td>
<td>4.77</td>
<td>5.6</td>
</tr>
<tr>
<td>obliged</td>
<td>59.83</td>
<td>55.76</td>
<td>54.54</td>
<td>56.98</td>
<td>47.21</td>
<td>54.54</td>
<td>47.62</td>
<td>54.95</td>
</tr>
<tr>
<td>unwilling</td>
<td>5.84</td>
<td>5.98</td>
<td>7.66</td>
<td>5.46</td>
<td>4.35</td>
<td>5.81</td>
<td>7.35</td>
<td>11.07</td>
</tr>
<tr>
<td></td>
<td>44.77</td>
<td>52.91</td>
<td>46.8</td>
<td>36.63</td>
<td>55.96</td>
<td>60.03</td>
<td>48.84</td>
<td>36.63</td>
</tr>
<tr>
<td></td>
<td>6.11</td>
<td>12.55</td>
<td>11.51</td>
<td>2.88</td>
<td>4.2</td>
<td>3.05</td>
<td>13.95</td>
<td>6.75</td>
</tr>
<tr>
<td></td>
<td>39.68</td>
<td>46.8</td>
<td>44.77</td>
<td>32.55</td>
<td>54.95</td>
<td>36.63</td>
<td>51.89</td>
<td>33.57</td>
</tr>
</tbody>
</table>

For each student, Z-score was derived for each of the internal factors. For each persona, a mean vector was figured out with the average of the Z-scores, as it is illustrated in Table 4.3. In the table, the upper figure in each cell indicates the mean value of Z-scores for all students belonging to the persona, while the lower one shows their standard deviation. We ordered whole cells in the increasing order of the standard deviation. A cell has a bold figure, if its standard deviation belongs to smallest one third in the whole cells. Each row of the mean values in the table is referred to as the weight vector of the persona for internal factors, because it indicates what weight the persona has for every internal factor. It shows the characteristics of the persona. Especially, the bold value is peculiar to the persona.
4.4.4 Associations of internal factors with behavior factors

For all students, the method has calculated the mean $\mu$ and the standard deviation $\sigma$ of the values for every internal factor. For every internal factor, students whose values are larger than $\mu + \sigma$ are picked up. In other words, we collected top 16% students in term of the strength of the internal factor. Table 4.4 presents the mean of behavior factor value of those top 16% students. Let $e$ be the mean of all student values for a specific behavior factor. The cells have bold figures if the values in the cells are either larger than 1.1 or smaller than 0.9. Those behavior factors are specific to the top students in terms of the strength of the internal factor. Let us change the way to see the table. The column vector of table4.4 represents how much a specific behavior factor is affected by each internal factor. The method refers to each column of the table as the gene vector of the behavior factor for the internal factors.

| Table 4.4: Association of internal factors with behavior factors |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | intri. | exri. | task. | self-e. | affec. | effo. | help. | meta. |
| t time          | 390.88 | 389.23 | 439.24 | 332.52 | 355.82 | 402.24 | 357.17 | 357.38 |
| i login         | 14.33  | 11.79 | 13.46 | 12.09 | 10.58  | 13.6  | 10.87 | 10.76 |
| a ad su         | 1.78   | 1.84  | 1.94  | 1.83  | 1.59   | 1.87  | 1.9   | 1.94  |
| z all su        | 6.85   | 6.97  | 7.11  | 6.92  | 6.42   | 7.03  | 7     | 7.18  |
| e ear su        | 4.49   | 3.91  | 3.46  | 3.22  | 2.09   | 2.08  | 1.67  | 3.35  |
| l las su        | 2.36   | 3.06  | 3.65  | 3.7   | 4.52   | 4.96  | 5.33  | 3.83  |
| s score         | 78.6   | 83.94 | 87.44 | 79.69 | 71.51  | 83.46 | 67.9  | 84.67 |
| c click         | 231.04 | 219.01 | 245.6 | 179.04 | 204.09 | 242.87 | 264.43 | 179.37 |
| v ad sco        | 11     | 11.84 | 12.81 | 11.3  | 9.86   | 10.57 | 12.37 | 12.21 |

4.5 Discussion: Preciseness in distinction

In addition to 2012, we also conducted the contextual inquiry for students in 2013. To examine their internal factors, we randomly picked up 15 students. Through the analysis of their scenarios, we obtain a real number vector of internal factor values for each of the students. Their personas are determined from the scenario contents by judgement of
4.6 Conclusions

Some students can learn by themselves, while others need to be attended by teaching staff. The latter are passive students. It is essential to focus teaching staff power on passive students to teach programming to far more students than teachers. To achieve it, we should distinguish passive students from others. Using the contextual inquiry, the proposed method identifies the personas of individual students. At the same time, it examines internal factors and behavior factors of them. The method figures out the association of personas with internal factors as well as the association of internal factors with behavior factors. Combining these associations, the method distinguishes passive students. The method examines the ability of the method. It also discusses the next step for the distinction of passive students immediate after every week exercise.
Chapter 5

Grasping motivation and learning strategies from behavior

5.1 Introduction

Psychologists assert motivation is an internal factor which arouses, directs, and maintains learning behavior [59]. Keller, who proposed the ARCS motivational model, states learners have individual differences with regard to motivation [30]. In addition to motivation, learning strategies are pointed out as an important predictor for leaning success [68, 2]. The method refers to motivation and learning strategies of a student as his or her internal factors. Vectors composed of their values represent his or her learning status. From the view point of learning status, students are classified into two groups: passive and active. Passive students need supervision from teaching staff to overcome difficulties, while active ones can find out a way to solve problems by themselves. It is preferable to predict active students or passive ones by repeating formative assessments. If educators can predict students during the course who will become active at the end, they can focus their supervision to prevent the remaining from running into passive states.
Learning status of a student changes during a course. Teachers perform formative assessment to know it on the spot [1]. They often use questionnaires, which are easy and simple [1, 26, 42, 35]. However, questionnaires have serious drawbacks. They imposes burdens on students. If questionnaires are conducted many times, students get tired of giving truthful answers. Vickers found questionnaires should not be conducted more than three times in a course [66]. Learning status obtained from questionnaires also has problems on its reliability, because students give subjective answers. We need a method to figure out objective learning status of individual students constantly without any burdens to them.

We propose a method to figure out learning status of students in an introductory programming course. The method uses internal factors of motivation and learning strategy [44]. It constantly takes student learning behavior on an e-learning site of programming exercises. It assumes each behavior factor appears as total accumulation of effects of multiple internal factors. Based on the assumption, the method uses the non-negative matrix factorization (NMF) [34], which numerically approximates decomposition of a matrix showing learning behavior of every student into a product of two matrices. One matrix is a gene matrix indicating how each internal factor affects into individual behavior. The other is a weight matrix representing weights individual students have for each of the internal factors.

The weight matrix shows the correspondence of students to the internal factors, which represent motivation and learning strategies. The method induces the correspondence using the initialization of the two matrices. We focus on institutions ordinarily repeating an identical course every year for programming training. Characteristics of learning status of students are expected to be maintained for successive courses. Since topics to be learned are same in corresponding weeks of identical courses, the method derives initial values for the matrices of every week for current students from the past student data of the corresponding week.
To acquire motivation and learning strategy comprehensively, contextual inquiry [5] is conducted on students of the previous course. The contextual inquiry brings about students strong for every internal factor. It enables us to figure out the objective associations of internal factors with behavior factors. It also brings the internal factors of each student. Since the internal factors are representative ones over the weeks in the course, we decompose the behavior matrix in every week in the previous course into a product of a weight matrix and a gene matrix, with initialization utilizing representative internal factors. The matrices derived in the decomposition represent the correspondence from matrix elements to the internal factors in the previous course. The matrices of every week of the previous course are used to initialize those of the correspondence week of the current course. NMF is used again to achieve the precise weight matrix representing learning status of individual students in the current course.

The method is applied to figure out learning status of students in actual C programming course of Ritsumeikan University in 2013. We compare results with learning status obtained by the contextual inquiry conducted after the course. The activeness of individual students can be predicted with more than 70% accuracy in the latter half of the course.

5.2 Behavior factors and internal factors

5.2.1 Behavior factors

Nowadays, every student taking a programming course can learn anywhere anytime because the course is often facilitated with an e-learning web site. He can refer to materials such as video, slides, and sample codes in the site. He reads an assignment, builds an idea to make a code, edits it, and compiles it. He repeats the process until he produces a good code to submit to the web site.
Every student studies in his own way. He takes his own learning behavior. One student may spend a long time to making codes, while another may take short time. Some students may submit rash codes just before their deadlines, while others may submit codes earlier after careful check. Learning behavior of each student can be represented with quantitative factors in a period of time. For example, a student refers to checking items 10 times, submits 5 codes, and so on, for a week. A value for each behavior factor can be based on calculation from learning logs recording how each student uses the web site.

A sample of 5 students, from $p_1$ to $p_5$, in table 5.1 shows five behavior factors. They are the learning time measured in hour ($t$), the number of early submitted codes ($e$), the number of codes submitted on the last day ($l$), the total number of clicks ($c$), and the score ($s$). Each entry of the table represents a value for a week. Code submission before the final day is regarded as early. Factor $c$ shows the number of mouse clicks a student issue to interact with widgets such as buttons and links on the web pages. Factor $s$ is the total score a student obtains for all assignments of the week.

As the table shows, the learning time of students $p_1$ and $p_3$ is more than 10 hours. Their clicking number is far more than those of the others. Since they submit many codes early, they seem to be positive for the programming exercise, though their scores are not high. On the other hand, students $p_2$, $p_4$ and $p_5$ spend less learning time, and have less clicking number than $p_1$, $p_3$ do. Students $p_2$, $p_4$ submit their codes on the last day rather than on early days, while $p_5$ submits more codes on early days than the last day. Their scores are fairly good, especially those of student $p_5$.

5.2.2 Learning status

Through behavior observable from outside, educators should consider what happens inside a student [30]. They should grasp internal factors of the students, to understand why the
student takes peculiar behavior under specific training conditions as the first step to seek for adaptive supervisions.

As the internal factors, MSLQ [44] enumerates motivation factors and learning strategy factors. For motivation, there are intrinsic, extrinsic, task value, expectancy, and affective. Students high in intrinsic factors participate in target tasks because their goals are achievement of the tasks. They study because of their challenge, curiosity, mastery, and so on. Students depending on extrinsic factors are conscious of evaluation from outside. External stimuli such as rewards, scores, and competition cause them to engage in the tasks. The task value factor is determined by student perception of the tasks in terms of interest, importance, and utility. The expectancy factor refers to self-efficacy, e.g. belief that their effort will result in learning success. The affective factor is related to anxiety of tests, grading for credits, and scores. For example, some students make effort because they worry about tests.

A learning strategy is a personal approach to master lectured items and solve assignments [44]. Learning strategies consist of two categories; one is a category of cognitive and meta-cognitive strategies, while the other is a category of resource management strategies. The former refers to methods to learn, that is, how to remember new concepts, to organize them, and to understand them. The latter corresponds to methods to tune and adjust the former. The latter consists of strategies for management of time and study environment, effort-regulation, peer learning, and help seeking. Study environment strategies regard to

<table>
<thead>
<tr>
<th>$B$</th>
<th>$t$ (lang time)</th>
<th>$e$ (erly sbmt)</th>
<th>$l$ (lst sbmt)</th>
<th>$c$ (ckg)</th>
<th>$s$ (scr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1$</td>
<td>11.52</td>
<td>3</td>
<td>2</td>
<td>654</td>
<td>55</td>
</tr>
<tr>
<td>$p_2$</td>
<td>7.75</td>
<td>1</td>
<td>5</td>
<td>191</td>
<td>75</td>
</tr>
<tr>
<td>$p_3$</td>
<td>12.73</td>
<td>4</td>
<td>0</td>
<td>506</td>
<td>60</td>
</tr>
<tr>
<td>$p_4$</td>
<td>6.81</td>
<td>2</td>
<td>4</td>
<td>132</td>
<td>83</td>
</tr>
<tr>
<td>$p_5$</td>
<td>8.47</td>
<td>4</td>
<td>2</td>
<td>310</td>
<td>100</td>
</tr>
</tbody>
</table>
setting up study places. Effort-regulation strategies work in the process whereby learners systematically direct their thoughts, feelings, and actions toward the attainment of their goals. Peer learning strategies correspond to collaboration with peers, while help seeking strategies aim at getting support from others including peers and instructors. Among these learning strategies, Yukselturk [68] found effort-regulation learning strategies are strong predictor for success in on-line courses.

5.2.3 Adaptive supervision

Each student has his own learning status. Suppose students $p_1$ and $p_3$ in table 5.1 struggle with pointers of C programming. Let us consider they consume much time for debugging before they submit codes because they do not have full image of the addresses of variables and values stored there. Since the students have not lost their curiosity for programming, teaching staff consisting of teachers and teaching assistants should clarify the image of a primitive variable, a pointer variable, and the address stored in a pointer variable. It is one way to draw a figure visually with a pen to explain the memory image. It may not preferable to show them an example code because it does not provide them with any memory image [30].

On the contrary, suppose students $p_2$ and $p_4$ do not understand how to define pointers and variables in codes. Since they are confused with difference of pointers and primitive variables, they might get so disparate that they copy codes from their friends, which make their scores high. The research [30] recommends that, teaching staff may explain them the difference of pointers from primitive variables using example codes. The staff should provide the students easy examples to make them understand how to define each kind of variables.

Finally, let student $p_5$ have clear understanding on pointers, which enables him to submit good codes in a short time. His success in attaining programming skill motivates
him to try advanced assignments in the succeeding classes. Teaching staff should let him study by himself, avoiding supervision with no effect [30].

Although students $p_1$, $p_3$, and $p_5$ have some similar behavior factors, the appropriate supervisions for each of them are different because their learning status is different [30].

It is essential to assess the learning status of every student during a course. Moreover, the assessment needs to be frequent, because the learning status of a student would change upon the learning topic, upon stimuli from teaching staff, etc. For instance, in the previous example, students $p_1$ and $p_3$ might get confident with pointers after receiving explanation of memory image of variables from teaching staff. They would engage in programming with different learning status in the later classes.

### 5.2.4 Obtaining student learning status

MSLQ [44] explains detailed descriptions of every internal factor. It provides sample questions to examine internal factors of students. Teaching staff tries to get the learning status of students using the sample questions. However, questionnaires cannot bring about reliable results if they are used many times during a course [66]. Questionnaire results have another drawback. They lack of the objectivity. In a questionnaire, a student specifies how each question goes for him with a specific scale. Since every student chooses the answer according to his own criteria, it is hard to obtain fair assessment of his learning status compared with a standard common to all students.

The contextual inquiry [5] is another good way to infer the learning status. In this method, teaching staff lets their students conduct a detailed and deep interview with each other. The conversations reveal matters such as why they study the subject, what motivates them, how they study it, and how they overcome difficulties. The results of the interviews are submitted to teaching staff as documents. Some external standards [59, 44] state relationships of student learning status with their behavior. Referring the external
standards, the teaching staff manually analyzes the documents to grasp learning status of each student.

As it is obvious from the assessment process, the contextual inquiry imposes heavy loads on the teaching staff. In addition, student time is consumed for interviews which are irrelevant to their learning. From the view point of education, it is hard to apply this method frequently to acquire their learning status. To make the matter worse, the students may explain the learning behavior in an objective manner only after the course finished.

Other conventional methods examine learning behavior for assessment, but they fail to address the problem that different internal factors may cause similar effects on learning behavior [68, 25, 31, 38]. The learning time of students $p_2$ and $p_4$ is short because of blind copies coming from their poor understanding, while $p_5$ takes short learning time because of complete understanding. The learning time and the score look similar among the students, but the submission days of $p_5$ differ from those of $p_2$ and $p_4$. Students $p_2$ and $p_4$ submit many codes on the last day which result in high score. If we focus on a part of the behavior factors, we cannot distinguish $p_2$ and $p_4$ from $p_5$. All of the behavior factors would be examined simultaneously, conscious of effects from each internal factors to multiple behavior factors. Since effects from multiple internal factors accumulate into specific behavior factors, they would be examined quantitatively.

5.3 Learning status through NMF

5.3.1 Internal factors reasoning behavior factors

Many internal factors affect a behavior factor. For the simplicity, let us consider 4 internal factors as shown in figure 5.1. The intrinsic motivation of student $x_i$ urges him to explore the subject, which makes him spend long time on the learning. The extrinsic motivation
Learning status through NMF

Figure 5.1: Many internal factors affect a behavior of scores makes him solve many assignments, which also contributes to lengthening his learning time. The affective motivation coming from the care about deadlines and scores pushes him to study. It affects his learning time, too. The self-regulation represents the regulation capability to adapt effort to the requirement of the course. If the course requires solving many exercises, he would take much time to solve them. In a quantitative manner, value of one behavior factor is considered to be results from accumulation of effects of multiple internal factors.

We assume each internal factor affects a specific behavior factor in proportion to its amount. After we know all internal factors of a student, we calculate the strength of the behavior factor as the weighted sum of the internal factors. Let \( w_{ik} \) \((k = 1, 2, ..., r)\) denote the strength of the \(k\)-th internal factor of student \(x_i\), where \(r\) is the number of the internal factors. \( w_i = [w_{i1}, w_{i2}, ..., w_{ir}] \) is referred to as the internal vector of student \(x_i\). It represents the strength of every internal factor of the student.

The total learning time \((t_i)\) of student \(x_i\) in a week results from the internal factors: intrinsic \((in_i)\), extrinsic \((ex_i)\), affective \((af_i)\), and self-regulation \((sr_i)\). Let \( w_i = \)
[\text{in}_i, \text{ex}_i, \text{af}_i, \text{sr}_i] \text{ be the internal vector of student } x_i. \text{ We have the following equation:}

\begin{equation}
\text{t}_i = \text{in}_i \cdot \alpha + \text{ex}_i \cdot \beta + \text{af}_i \cdot \gamma + \text{sr}_i \cdot \delta = (\text{w}_i, \text{g}_{it}) \tag{5.1}
\end{equation}

using the inner product of \text{w}_i and \text{g}_{it}, \text{ where } \text{g}_{it} = [\alpha, \beta, \gamma, \delta]^T. \text{ The vector } \text{g}_{it} \text{ represents how much each internal factor affects the total learning time of } x_i.

The last day submission (\text{l}_i) of \text{x}_i \text{ indicates the number of source codes submitted within the day just before the deadline. In the same fashion of presenting the learning time, } \text{l}_i \text{ can be calculated quantitatively as following:}

\begin{equation}
\text{l}_i = \text{in}_i \cdot \rho + \text{ex}_i \cdot \sigma + \text{af}_i \cdot \tau + \text{sr}_i \cdot v = (\text{w}_i, \text{g}_{il}) \tag{5.2}
\end{equation}

where \text{g}_{il} = [\rho, \sigma, \tau, v]^T \text{ is the vector represents how much each of the internal factors affects the last day submission of } x_i.

The value of each of the behavior factors is derived from learning logs. Since they are quantities representing actual learning activities, they are all non-negative. A student has many behavior factors. Suppose the behavior factor vector of student \text{x}_i \text{ is } \text{b}_i = [b_{i1}, b_{i2}, ..., b_{in}] \text{ where } b_{i1}, b_{i2}, ..., b_{in} \text{ are values of behavior factors of } x_i \text{ and } n \text{ is the number of behavior factors. We have}

\begin{equation}
\text{b}_i = \text{w}_i \cdot \text{G}_i \tag{5.3}
\end{equation}

where

\begin{equation}
\text{G}_i = [\text{g}_{i1}, \text{g}_{i2}, ..., \text{g}_{ij}, ..., \text{g}_{in}] \tag{5.4}
\end{equation}

using vector \text{g}_{ij} = [g_{j1}, g_{j2}, ..., g_{jk}, ..., g_{jr}]^T \text{ which represents how much each of } r \text{ internal factors of student } x_i \text{ affects the } j\text{-th behavior factor of him. As it is shown in equation (5.3), behavior factors of the student results from } \text{G}_i \text{ weighted with his internal}
factors. We can regard each element of $g_{ij}$ as a vector each internal factor of student $x_i$ affects the $j$-th behavior factor of him. The method refers to $g_{ij}$ ($j = 1, 2, ..., n$) and $G_i$ as a gene vector and a gene matrix of student $x_i$, respectively.

If we can compute internal vector $w_i$ of the student from values of his behavior factors, we know his learning status without any load irrelevant to his learning. Extending the idea, our target is to figure out the values of the internal factors for all students taking a course.

### 5.3.2 Personas from past students

From experiences [46], we found all students taking a course can be classified into some groups, each of which consists of students similar in their learning status. Students with common learning status take similar behavior under specific course settings. We let a persona [11] represent a group. A persona is a virtual student representing typical learning status and behavior of a group. Teaching staff should accommodate their supervision according to each persona [30].

It is possible for teaching staff to know precise learning status of past students, using an interview based on the contextual inquiry. Teaching staff analyzes the scenarios of the contextual inquiry manually to find out learning status of individual students. In this work, they assess the strength of every internal factor for each student, referring to MSLQ [44]. If the scenarios include description matching a sample question of MSLQ, they regard the interviewee, the student in the scenario, has stronger internal factor corresponding to the sample question.

Teachers classify scenarios into specific group if they are similar in terms of internal factors. As the result, each group corresponds to a persona. The strength of every internal factor of a specific student composes a vector representing his learning status. The vector of learning status for every persona, the persona vector, is figured out by averaging the
values of each internal factor of the students in the same group.

Personas are often stable for some successive courses in the same training institute, as far as the settings of the courses are unchanged. Once teaching staff has taken contextual inquiry of students, the vectors of personas can be utilized for succeeding courses.

## 5.3.3 Non-negative matrix factorization

Our proposed method obtains the learning status from their behavior which is taken automatically. It makes the best use of the association of the behavior with internal factors under specific course settings.

Let us assume one programming course consists of 15 weekly classes. Many students, denoted as $x_i (i = 1, 2, ..., m)$, take the course. Suppose $n$ behavior factors are sampled from each student. For the student $x_i$, they are represented with $b_i = [b_{i1}, ..., b_{ij}, ..., b_{in}] (j = 1, 2, ..., n)$. Let us assume each student has $r$ internal factors $f_k (k = 1, 2, ..., r)$. Student $x_i$ has an internal vector denoted as $w_i = [w_{i1}, ..., w_{ik}, ..., w_{ir}] (k = 1, 2, ..., r)$. For student $x_i$, we have $b_i = w_i G_i$. Gene matrix $G_i$ represents how much each of the internal factors affects individual behavior factors. Under a specific course settings, which consists of same assignments, same requirements, and same teaching staff for all students, gene matrices $G_i (i = 1, 2, ..., m)$ for all of the students taking the course will converge to a single matrix corresponding to the course settings.

Suppose a matrix $B^q$ represents learning behavior of $m$ students of the $q$-th week of a programming course. Since each student has $n$ sampled behavior factors, each entry of $B^q$, $b_{ij}$, represents the $j$-th behavior factor ($j = 1, 2, ..., n$) of the $i$-th student ($i = 1, 2, ..., m$). Since student $x_i$ has the internal vector denoted with $w_i$, we can represent the matrix specifying the strength of $r$ internal factors of $m$ students in the $q$-th week as $W^q = [w_1, w_2, ..., w_m]^T$, where $w_i = [w_{i1}, w_{i2}, ..., w_{ir}]$. Each row of $W^q$ is exactly the learning status of a student, which we want to find out. $G^q = [g_{kj}]$ shows how much the
The relationship among matrices $B^q$, $W^q$, and $G^q$ is illustrated in figure 5.2. The rows and the columns of $W^q$ correspond to students and internal factors, respectively. The rows and the columns of $G^q$ correspond to internal factors and behavior factors, respectively.

$$B^q_{mn} \approx W^q_{mr} \cdot G^q_{rn}$$ (5.5)

NMF assumes all of the elements of matrices $B^q$, $W^q$, and $G^q$ are non-negative. The difference of the left hand side from the right hand side of the formula in every element is tried to be simultaneously minimized in the decomposition of NMF, in order to find consistent combination of element values in the three matrices. The consistent value combination explains learning behavior of whole students in terms of the learning status of every student as well as effects of every internal factor on individual behavior factors. Suppose some lazy students do not spend long time on programming, copy their friends codes to get high scores. Even if they are similar to active students in the learning time and scores, the remaining behavior such as the last day submission would be different from those of the actives. The weight vectors of the lazy students figured out with NMF.
explain all of their internal factors which would differ from those of the actives.

5.3.4 Perspective for decomposition

The NMF algorithm can decomposes $B^q$ into matrices $W^q$ and $G^q$ with the hill climbing method. However, it is impossible for us to know which of the factors of the matrix $W^q$ corresponds to a specific internal factor explained in MSLQ, because we cannot constrain NMF to arrange the columns in $W^q$ and the rows in $G^q$ as we expect. We need a perspective to lead $B^q$ to be decomposed so that we can grasp the correspondence of each internal factor to a column in $W^q$ and a row in $G^q$. One way for the perspective is to provide appropriate initial values in $W^q$ and $G^q$, so that the $k$-th column of $W^q$ and the $k$-th row of $G^q$ approximate the correspondence to the internal factor $f_k$.

To obtain such initial values, we focus on usual practices that many courses are repeated annually under similar settings. Let us consider a course in the past whose settings are similar to the current one. It is assumed that students identical in learning status would take similar learning behavior there. Since we know learning status of every past student through the scenario analysis, we can adopt the following perspectives.

- For every current student, we find a past student similar to him in terms of learning behavior. We can use the learning status of the past students for initial values of those of current students.

- For each internal factor, we collect past students who have strong values. We calculate the mean of their behavior vectors. The mean vector shows representative behavior of students strong in the internal factor. It indicates the effect of the internal factor on individual behavior factors.

We achieve the good approximation for the current course with two phases, as it is depicted in figure 5.3. First, for the previous course, we decompose behavior matrix in

94
each week into a \textit{weight matrix} and a \textit{gene matrix}. Second, expecting students similar in internal factors are likely to take similar behavior, we use the matrices of the previous course as initial matrices to approximate the decomposition for the current course.

![Diagram](image)

Figure 5.3: Contextual inquiry for NMF initialization

### 5.3.5 Making full use of previous course

Suppose the settings of the current course are quite similar to those of a previous course. From students in the previous course, we can acquire their scenarios with the contextual inquiry on programming learning, after they finished the course. The scenario analysis explained in section 5.3.2 brings about persona vectors indicating the strength of every internal factor. The persona vectors for every student constitute the \textit{weight matrix} in the
previous course.

Scenarios from contextual inquiry represent internal factors of students for the whole course. They do not state student internal factors in specific weeks. The matrix is the \textit{weight matrix} for the whole course. Let us denote it as $\Omega$.

The scenario analysis also reveals students strong in a specific internal factor. Picking up only those students for the internal factor, we can make the mean of their behavior vectors. As explained in section 5.3.1, it is regarded as effects the internal factor gives to every behavior factor [48]. The effect an internal factor gives to every behavior factor would be different by weeks. The collection of the effects forms a \textit{gene matrix} for a week. Let us denote it as $\Gamma_q$ for $q$-th week of the course.

We know past student learning behavior in each week. Let $\Phi^q$ denote the behavior matrix in the $q$-th week of the previous course in figure 5.3. Taking $\Omega$ and $\Gamma$ as initial matrices, NMF is applied to decompose $\Phi^q$ into $\Omega^q$ and $\Gamma^q$, which are the \textit{weight matrix} and the \textit{gene matrix} for the $q$-th week of the previous course, respectively.

Under specific course settings, students with similar learning status would take similar learning behavior. In other words, \textit{gene matrix} $G^q$ for the $q$-th week of the current course is expected to be similar to \textit{gene matrix} $\Gamma^q$ for the corresponding week of the previous course. Let us regard $\Gamma^q$ as initial \textit{gene matrix} $G^q_0$ in the NMF decomposition for the $q$-th week of the current course.

The learning status in the $q$-th week for all of the past students has been calculated as $\Omega^q$ through the NMF decomposition. For each student of the current course, it is possible to determine the most similar student in the previous course, in terms of learning behavior in the $q$-th week. Let us regard the internal factors of the most similar student in the previous course constitute a vector approximating the internal factors of the student in the current course. A list of such vectors for all current students is close to the \textit{weight matrix} in the $q$-th week of the current course. Let us denote the matrix as $W^q_0$. 96
5.4 Application for actual courses

Using initial matrices $W_0^q$ and $G_0^q$, $B^q$ is decomposed into $W^q$ and $G^q$. Once $W^q$ is obtained, we know the learning status of every student of the current course. To find out personas for the current students, we use persona vectors in the previous course. Suppose the previous course has personas with vectors $p_d (d = 1, 2, ..., z)$, where $z$ is the number of the personas. Teaching staff compare $w_i$ of $x_i$ with $p_d$ to know the persona of $x_i$.

5.4 Application for actual courses

5.4.1 Application outline

We figured out learning status of freshmen in the introductory C programming exercise course at College of Information Science and Engineering, Ritsumeikan University, Japan in the course year 2013. Suppose the course proceeds to the $q$-th week in 2013. We aim to obtain the learning status of individual students of the course through the decomposition of matrix $B^q$ into matrices $W^q$ and $G^q$.

To initialize matrices $W^q$ and $G^q$ with proper values, we utilize the data collected from the course in 2012, as explained in section 5.3.5. The combination of the contextual inquiry and the two-step NMF enables us to get the learning status of individual students of the course in 2013, while it is in progress. We achieve $\Omega$ and $\Gamma$ through the contextual inquiry of students in 2012. The first NMF is applied to $\Phi^q$ to figure out $\Omega^q$ and $\Gamma^q$ of the course in 2012, taking $\Omega$ and $\Gamma$ as their initial matrices, respectively. The second NMF decomposes $B^q$ into $W^q$ and $G^q$, taking $\Omega^q$ and $\Gamma^q$ as their initial matrices to figure out the learning status of students in 2013.

After vectors of internal factors of individual students are obtained in 2013, the cosine similarity is calculated between individual students and personas. It predicts active students. The remaining needs cares, not to run into passive ones.
5.4.2 Learning behavior of students

To collect learning behavior from students without imposing irrelevant loads, we implemented an e-learning site for programming exercise, with Web 2.0 technology. Every operation with widgets such as buttons on the site is recorded with the ID of the student and the time stamp. Learning behavior logs which are time series of operations indicate what kind of information the student sees, when the student submits assignments, and so on.

Students mention various behavior relevant to learning in their contextual inquiries. We figured out many of them as behavior factors from records in their learning logs. In addition to the factors described in section 5.2.1, there are the number of log-in times into the site, the number of codes submitted for advanced assignments, that for all assignments, scores of advanced assignments, the number of times viewing the graph representing learning progress, advanced assignments, and the sample codes of advanced and basic assignments.

5.4.3 Cleansing values

We should fairly treat values of behavior factors measured with different units [51]. The learning time is measured in minutes, which has very big values, e.g. 250 minutes, while the number of early submissions is measured by the number of submitted source codes, e.g. 5 source codes. All of values in the matrices should be normalized into the same scale.

Suppose a specific behavior factor $b$ consists of values whose maximum and minimum are $O_M$ and $O_m$, respectively. The proposed method scales the measured value $v_m$ for factor $b$ into the normalized value $v_n$ ranging from 0.0 to 1.0 using $v_n = \frac{v_m - O_m}{O_M - O_m}$.

Since NMF allows only positive linear relation from $W^q$ and $G^q$ to $B^q$, we need value
conversion to deal with a reverse relationship of one factor to another. Let us take a concrete example. The intrinsic factors affect both of the learning time, and the last day submission. The longer the learning time is, the stronger the intrinsic factor is. On the contrary, the less the number of last day submissions is, the stronger the intrinsic factor is. Even though the number of last day submissions is a good behavior factor representing the intrinsic factor, it has a reverse relationship with the intrinsic factor. Values of the number of last day submissions have to be reversed so that the NMF works well. One way to reverse the values is to subtract an original value from the maximum value.

5.4.4 Five kinds of personas

We analyzed scenarios obtained in the inquiry of 40 freshmen enrolling the course of 2012. The analysis brings about 5 personas: easy-going, industrious, demanding, obliged, and unwilling [46].

The easy-going persona knows how to study to acquire programming. They are willing to study the useful and interesting programming subject because they are strong in intrinsic motivation. They have strong meta-cognitive strategies enough to achieve programming with regular effort. The industrious persona also has high intrinsic motivation as the easy-going persona does, but less meta-cognitive strategies. Both of the personas are active, because they can learn by themselves. When they encounter difficulties, they can find a way to overcome them, so as to put forward their learning.

In the meanwhile, the rest three personas, the demanding, the obliged, and the unwilling cannot overcome difficulties by themselves. When they encounter problems, they have little will to solve them. They are passive because they need supervision when they face difficulties. The demanding persona wants an environment where knowledge can be acquired easily. The obliged persona feels pains in learning, because they lack of initiatives to learn. Among the three personas, the unwilling is the most serious one, which is
weak in intrinsic, extrinsic, task value, and efficacy, toward the learning.

Table 5.2: Learning status vectors of personas

<table>
<thead>
<tr>
<th>Persona</th>
<th>intr</th>
<th>extr</th>
<th>task</th>
<th>self</th>
<th>affe</th>
<th>effo</th>
<th>help</th>
<th>meta</th>
</tr>
</thead>
<tbody>
<tr>
<td>easy-going</td>
<td>0.54</td>
<td>0.53</td>
<td>0.59</td>
<td>0.56</td>
<td>0.44</td>
<td>0.45</td>
<td>0.46</td>
<td>0.68</td>
</tr>
<tr>
<td>industrious</td>
<td>0.68</td>
<td>0.60</td>
<td>0.58</td>
<td>0.63</td>
<td>0.43</td>
<td>0.58</td>
<td>0.43</td>
<td>0.58</td>
</tr>
<tr>
<td>demanding</td>
<td>0.33</td>
<td>0.55</td>
<td>0.30</td>
<td>0.60</td>
<td>0.48</td>
<td>0.08</td>
<td>0.58</td>
<td>0.15</td>
</tr>
<tr>
<td>obliged</td>
<td>0.38</td>
<td>0.54</td>
<td>0.42</td>
<td>0.21</td>
<td>0.60</td>
<td>0.69</td>
<td>0.46</td>
<td>0.21</td>
</tr>
<tr>
<td>unwilling</td>
<td>0.27</td>
<td>0.42</td>
<td>0.38</td>
<td>0.13</td>
<td>0.58</td>
<td>0.21</td>
<td>0.52</td>
<td>0.15</td>
</tr>
</tbody>
</table>

In the way explained in section 5.3.2, we obtained the persona vectors described in table 5.2. The table shows the association of the personas with the internal factors, after the normalization. In the table, column names intr, extr, task, self, affe, effo, help, and meta stand for intrinsic, extrinsic, task value, self-efficacy, affective, self-regulation, help seeking, and meta-recognition, respectively. The active personas show high values of most of the internal factors, while the passive personas show low values of intrinsic, task value, and meta-recognition.

5.4.5 Weight matrices of previous course

For the previous course, personas of students are determined though the analysis of their learning scenarios. In the contextual inquiry, students answer to interviews, looking back over their learning after the course finishes. The motivation and learning strategies in their scenarios would be their typical ones along the course. For example, if a student turns out the easy-going in the analysis, he would belong to the easy-going persona for many weeks of the course. Let a predominant persona be a persona a student belongs to in most of the weeks of the course.

Table 5.3 shows an example of initial internal vectors for 6 students. In the example, students $y_1$ and $y_6$ have the same predominant persona, which is demanding. When $\Phi^q$ is decomposed, the persona vector of the predominant persona is used as the internal vector for each student, as explained in section 3.5. The vector constitutes $\Omega$, which is the same
5.4 Application for actual courses

Table 5.3: Weight matrix initialization

<table>
<thead>
<tr>
<th>Persona</th>
<th>( \Omega )</th>
<th>intr</th>
<th>extr</th>
<th>task</th>
<th>self</th>
<th>affe</th>
<th>effo</th>
<th>help</th>
<th>meta</th>
</tr>
</thead>
<tbody>
<tr>
<td>demanding</td>
<td>( y_1 )</td>
<td>0.33</td>
<td>0.55</td>
<td>0.30</td>
<td>0.60</td>
<td>0.48</td>
<td>0.08</td>
<td>0.58</td>
<td>0.15</td>
</tr>
<tr>
<td>easy-going</td>
<td>( y_2 )</td>
<td>0.54</td>
<td>0.53</td>
<td>0.59</td>
<td>0.56</td>
<td>0.44</td>
<td>0.45</td>
<td>0.46</td>
<td>0.68</td>
</tr>
<tr>
<td>industrious</td>
<td>( y_3 )</td>
<td>0.68</td>
<td>0.60</td>
<td>0.58</td>
<td>0.63</td>
<td>0.43</td>
<td>0.58</td>
<td>0.43</td>
<td>0.58</td>
</tr>
<tr>
<td>obliged</td>
<td>( y_4 )</td>
<td>0.38</td>
<td>0.54</td>
<td>0.42</td>
<td>0.21</td>
<td>0.60</td>
<td>0.69</td>
<td>0.46</td>
<td>0.21</td>
</tr>
<tr>
<td>unwilling</td>
<td>( y_5 )</td>
<td>0.27</td>
<td>0.42</td>
<td>0.38</td>
<td>0.13</td>
<td>0.58</td>
<td>0.21</td>
<td>0.52</td>
<td>0.15</td>
</tr>
<tr>
<td>demanding</td>
<td>( y_6 )</td>
<td>0.33</td>
<td>0.55</td>
<td>0.30</td>
<td>0.60</td>
<td>0.48</td>
<td>0.08</td>
<td>0.58</td>
<td>0.15</td>
</tr>
</tbody>
</table>

over the weeks. \( \Omega \) will be refined so as to present internal factors of individual students in each week of the previous course, through the decomposition of \( \Phi^q \) with NMF.

5.4.6 Gene matrices of previous course

A gene matrix represents the association of behavior factors with internal factors. Each row vector shows effects of an internal factor to each of the behavior factors. The effects would be revealed from behavior of students strong at the internal factor. Internal factors of every student in the previous course are known from the scenario analysis. The method picks up top 16% students strong for each internal factor, because they are stronger than average students by more than 1.0 times standard distribution. The method calculates the mean for each behavior factor of the students superior to others. For example, to figure out the association of the intrinsic factor with every behavior factor, the top 16% students at the intrinsic factor are filtered out. The mean value of each behavior factor is calculated for these top students. It brings about the row vector corresponding to the intrinsic factor. We can get \( \Gamma^q \), using the mean of every behavior factor of students superior to others at every remaining internal factor.
5.4.7 Two-step NMF

The elements in $\Omega$ do not change over 15 weeks of the previous course. However, behavior of students would change weekly because the learning topic varies every week. For example, student would solve assignments for conditional statements easily, while they would struggle with pointers. They consume more time in the week for the latter than for the former. The method refines $\Omega$ and $\Gamma_q$ into $\Omega_q$ and $\Gamma_q$ through the decomposition of $\Phi_q$ with NMF, respectively.

Weight matrix $W_q$ consists of row vectors of internal factors of individual current students. For each student in the current course, we find the most similar one in the same week of the previous course in terms of behavior. To achieve it, we calculate the cosine similarity of his behavior vector for every student in the previous course in the same week. The internal vector of the most similar past student is taken from $\Omega_q$. Our method uses it to set up each row of an initial weight matrix for the $q$-th week of the current course. In the meanwhile, $G_q$ is initialized with $\Gamma_q$ as it is. NMF refines initial matrices of $W_q$ and $G_q$ for the decomposition of $B_q$.

5.5 Evaluation and discussion

5.5.1 Evaluation overview

With the proposed method, we calculated learning status of individual students of the course in 2013, which consists of 48 students. In the 1st week, students practiced UNIX operations. They had no programming exercise. The 2nd week and 3rd week, there were many basic assignments but there were not any advanced ones. In the 14th week, scores were not taken into account of the course. We did not include these weeks in the calculation.
5.5 Evaluation and discussion

For every week, we figured out the internal vectors of the students in 2013 with the proposed method. Based on the cosine similarity, we identified the most similar persona vector among the five ones mentioned in section 5.4.4. Separately, we applied the contextual inquiry on the students in 2013, after the course finished. We analyzed the scenario of each student to identify the persona of the student.

5.5.2 Prediction of learning status

After learning week by week, students finally get either of active or passive. Every week, teaching staff tries to maximize students who are finally active. We identified the predominant persona of each student with the scenario analysis. Since the scenario of $x_i$ represents his status after the course finishes, his predominant persona represents his final learning status. It would be preferable the method could predict which learning status each student finally has. Let us examine whether the method can predict the final learning status of each student.

Let us consider the major learning status of $x_i$ during 3 recent successive weeks. In each successive 3 weeks, we calculated how many percentage of the students have major learning status according with their final one. The higher the rate, the more precisely the method predicts final learning status of students. Figure 5.4 depicts the result.

In the graph, the horizontal axis shows the successive weeks. Apparently, the prediction is much more precise in the active learning status than in the passive one. After the 9th week, the method can predict active students in the final learning status in the accuracy over 70%. Since we can distinguish active students from others after the 9th week, teaching staff can focus their attention on the other students who cannot solve learning difficulties by themselves. Efficient supervision would lead the course to a successful one.

Figure 5.4 shows the prediction for passive students gets worse as the course proceeds. Let us consider why the method is poor at predicting passive ones. The method founds on
the scenario analysis to determine personas. Before we analyze scenarios, we got consent from students. Many active students consent. The number of consenting easy-going and industrious students was 9 and 15, respectively. But, few students consent if they are demanding, obliged, and unwilling students. The number was 5, 6, and 5, respectively. The method performance to discriminate a specific persona gets higher, if it founds on many example students of the persona.

Teaching staff generally designs course settings to make students follow a specific learning discipline, with which they expect to improve student programming. Most of students working harder would get to know the learning discipline in the given course settings.

In the both years of 2012 and 2013, the method was applied to the compulsory programming exercise course. Active students worked harder as the course proceeds. They would take similar learning behavior in the both years. It allows the method to find active students more accurately as the course proceeds.

![Figure 5.4: Prediction of the active and the passive](image_url)
The passive learning status founds on demanding, obliged, and unwilling personas. Few examples for them prevents us from attaining all kinds of behavior of passive students. Passive students in 2013 might take various kinds of behavior passive students in 2012 did not take. It degraded the prediction for passive students. However, data collection in many similar courses would cover various kinds of passive behavior, which improves the prediction.

### Table 5.4: Course settings in 2012 and 2013

<table>
<thead>
<tr>
<th>id</th>
<th>course in AY2012</th>
<th>wk</th>
<th>learning item</th>
<th>course in AY2013</th>
<th>wk</th>
<th>learning item</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>Linux operation</td>
<td>1</td>
<td></td>
<td>Linux operation</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>c2</td>
<td>printf, scanf</td>
<td>2</td>
<td></td>
<td>printf, scanf</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>c3</td>
<td>variable, expression</td>
<td>3</td>
<td></td>
<td>variable, expression</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>c4</td>
<td>conditional statement</td>
<td>4</td>
<td></td>
<td>conditional statement</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>c5</td>
<td>loop statement</td>
<td>5</td>
<td></td>
<td>loop statement</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mid-course test</td>
<td></td>
<td></td>
<td>nested loop</td>
</tr>
<tr>
<td>c6</td>
<td>nested loop</td>
<td>6</td>
<td></td>
<td>nested loop</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>c7</td>
<td>function</td>
<td>7</td>
<td></td>
<td>function</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>c8</td>
<td>array</td>
<td>8</td>
<td></td>
<td>array</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>c9</td>
<td>function with array</td>
<td>9</td>
<td></td>
<td>function with array</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>mid-course test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>end-course test</td>
</tr>
<tr>
<td>c10</td>
<td>pointer</td>
<td>11</td>
<td></td>
<td>pointer</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>c11</td>
<td>string</td>
<td>12</td>
<td></td>
<td>string</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>c12</td>
<td>structure</td>
<td>13</td>
<td></td>
<td>structure</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>c13</td>
<td>recursive call</td>
<td>14</td>
<td></td>
<td>recursive call</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>end-course test</td>
<td>15</td>
<td></td>
<td>end-course test</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

### 5.5.3 Accountability for active students

The method figures out internal vectors of students, assuming they take identical learning behavior in similar courses. Table 5.4 shows the course settings in 2012 and 2013. The course settings of the two courses are similar with each other, except the week of mid-course test. In 2012, the mid-course test was held in the 10\textsuperscript{th} week, while it was in the 6\textsuperscript{th} week in 2013. The proposed method initializes the matrices in 2013 with those in 2012, making correspondence of the learning items in every week. Column "id" indicates the
index of the corresponding learning items in 2012 and 2013.

We have found that students of Ritsumeikan university strongly care for tests and scores [46, 49]. The difference of the mid-course test would give some effects on the calculation result of the method.

Figure 5.5 depicts the rate of students who turned out active with the proposed method during the course of 2013. The horizontal axis represents the index of the corresponding learning items in 2012 and 2013.

![Figure 5.5: Rates of active students](image)

Active students decrease in the week in learning item c5. Since the mid-course test is held just after c5 in 2013, the students tentatively took pessimistic learning behavior, such as viewing sample codes for assignments frequently. However, the students in 2012 did not change their learning behavior in the week of c5. Because of it, active students in 2013 decreased up to less than 20% in figure 5.5. On the contrary, in the week corresponding to c9, many students in 2012 take pessimistic learning behavior, while students in 2013 presents no change in their behavior. It makes the method to regard approximately 90% of students active in 2013 in figure 5.5.
The graph indicates the dependency of the method on the similarity in the course settings. The method initializes the internal vectors of students in 2013, founding on the behavior similarity to students in 2012. The initial weight matrix of the c5 week in 2013 regarded many students as passive students. On the contrary, the initial gene matrix is determined based on the contextual inquiry results for the students in 2012. Its element values are set up with the average of learning behavior items of students strong for every internal factor. The course in 2012 still had students taking active behavior, even though the number is small. The mid-course test did not make significant difference on the initial gene matrix of the c5 week in 2013 from those of other weeks. The discussion above accounts for the change of the number of students in figure 5.5.

5.5.4 Accountability for accuracy

Every week, the method identified the persona of each student with the cosine similarity of the internal vector to the persona vectors. We separately determined their predominant personas through the contextual inquiry for students in 2013. Let \( n_m \) and \( n_c \) be the number of students who turned out active with the method and that with the contextual inquiry, respectively. Let \( n_b \) be the number of students who were active in the both cases. Suppose we can find students who are truly active with the contextual inquiry. The recall, \( R \), for the method to find active students is represented with \( n_b/n_c \), while the precision, \( P \), with \( n_b/n_m \). The f-measure, \( F \), is derived with \( 2PR/(P+R) \).

Figure 5.6 depicts the recall, the precision, and the f-measure along with the corresponding learning items in 2012 and 2013. Generally, as active students increase, the recall gets higher, while the precision gets lower. Since there were few active students in the week of c5 in 2013, the recall is low. In the week of c9, many students were active, the recall is high. However, the precision gets down in the both weeks.

The initial gene matrix is calculated with behavior of students strong in each internal
factor. It is less dependent even in the week just before the mid-course test, because students strong in each internal factor do not change their behavior temporally. Students strong in the intrinsic motivation factor always try to solve many assignments, while students with strong extrinsic motivation would check their own scores many times. Meanwhile, before the mid-course test, there are students who are active in themselves, but mind the mid-course test. They would take passive behavior in some learning behavior factors. But, they do not take typical passive behavior as inherently passive students do. Their learning behavior is a peculiar one specific to the week just before tests.

The gene matrix consists of gene vectors. The method tries to represent the behavior vector for each student with the inner products of gene vectors and internal vectors. The behavior vector just before the mid-course test is different from those of other weeks. Since gene vectors are less dependent on the mid-course test, elements in the internal vector take values specific to the week just before the mid-course test.

Every week, the method determines the persona of each student, to judge whether the student is active. In the determination, the method calculates the cosine similarity
of the student internal vector to every persona vector in 2012. Since the persona vectors represent typical predominant personas over whole weeks, they reflect almost no effect of the mid-course test. Therefore, the precision to determine the persona gets low, which degrades the precision to predict active students.

The proposed method assumes students in a specific educational institute take similar learning behavior under the courses. Based on the assumption, the mechanism of the method sufficiently accounts for the transition depicted in figure 5.5 and figure 5.6. The assumption works well to figure out learning status of students.

5.6 Conclusions

We propose a method to model the learning status of students every week of a programming course. The method uses NMF to decompose the matrix of student learning behavior which are automatically taken with a e-learning web site. Comprehensive motivation and learning strategies of previous course students taken with contextual inquiry is essential for initial data for the matrices of NMF approximation. The application to actual programming courses show the method can apply the method to figure out learning status of student under similar programing course settings. As future works, we are going to apply the method on a series of successive courses to find out persona vectors which cover various behavior of passive personas. It would bring high predictability for not only active but also passive personas.
5.6 Conclusions
Chapter 6

Conclusions and future works

6.1 Difficulties in building course plan

Programming is an indispensable ability for students whose major is IT. However, many students feel difficulties to obtain the ability. They should study many difficult concepts. In addition to that, they have to attain skills to be acquired through lots of experiences. Some of them give up learning itself.

Motivation and learning strategies are factors which play a vital role to make the students sustain their effort toward the programming learning. If a student is motivate to learn, he has inspiration and will to take learning activities to achieve the task. He tries to overcome difficulties until the goal is achieved. Motivation and learning strategies are determining factors resulting in learning success. Instructional Design and Motivational Design center on students to design the course and instructions. The methods try to examine motivation and learning strategies of individual students to make a course matching their preferences most.

However, motivation and learning strategies are not observable factors. The teachers cannot obtain the information of their students before the course when they design the
course for them. It is because the students of the on-coming course have not experienced
the course. They cannot express any opinions about the course. The teachers cannot
obtain the information during the course, either. Since the teachers must supervise many
students with few teaching staffs, they want to find students stuck in the learning, to focus
their power on the stuck students. The teachers often use questionnaires and interviews.
However, these ways to obtain the information take much time of students. It also imposes
tasks out of the learning on them.

Teachers should provide a good course which makes as many as students can achieve
the course goal, utilizing motivation and learning strategies of students. To establish a
good introductory programming course, there are two matters which the teachers have to
c onsider carefully. One matter is a course plan to make the students study to master the
weekly goal. The other is the formative assessments during the course.

6.2 Solutions

6.2.1 Design of a course plan with PMD method

It is difficult to make a course plan with a discipline leading all the students to study
well. It is hard to motivate all of the students because they are various motivation
components. We consider to group the students based on their motivation and learning
strategies. A group consists of students similar in term of degree of their motivation
and learning strategies. The number of groups would be much less than the number of
students. When designing the course, educators can consider specific groups to give careful
supervisions, e.g. groups of students who hold low intrinsic motivation and low task value
motivation. Persona is a virtual user representing the characteristic of motivation and
learning strategies of a group.

Students belonging in a persona have similar engagement in the learning and similar
approach to obtain the knowledge and skill the course requires. Factors motivating them would be similar. The scenarios of a persona specify what behavior students belonging to the persona would take. When making the course plan, the teachers examine the presence of all of the personas among the whole students. They build a course plan to take care of all of the personas during the course as much as possible. Along the course, even though the persona of a student changes, the course can take care any persona. For example, TAs are assigned to proper students who need supervisions to move forward their learning.

Personas among the students in the same university are quite similar in characteristics in some successive courses. To understand the students of the coming course, we conduct contextual inquiry on the students of the past course when they finished the course. Because the students have experienced the course, they express their programming learning experience in a comprehensive fashion, which bring about details of their motivation and learning strategies.

In Ritsumeikan University, we have found 5 personas from their contextual inquires in two successive programming courses, courses of years 2012 and 2013. We call them the easy-going persona, the industrious persona, the demanding persona, the obliged persona, and the unwilling persona. The easy-going one and the industrious one are active, because they can learn by themselves. The other three personas cannot overcome difficulties by themselves. When they encounter problems, they have little will to solve them. They are called passive personas.

We propose the PMD (Persona, Motivation, and Discipline) method to design a course plans. The design centers on characteristics of motivation and learning strategies of the personas in the students. The design directs all the personas to study to achieve the course with a discipline which is the incorporation of good learning strategies and affective components of the personas. The design motivates all personas with supporting services corresponding to motivation components of each of persona. Applying the PMD method to over 500 students supervised by 10 teachers and 100 TAs, we have improved the learning
time of the students. We have also succeeded in keeping their scores maintained horizontal in 2 successive years.

6.2.2 Formative assessment of personas with NMF

Motivation and learning strategies of a student express his learning status. The learning status changes during a course. The teachers often use questionnaire to get the information. However, it has serious drawbacks. It imposes burdens on students. If it is conducted many times, students get tired of giving truthful answers. We propose a method to figure out learning status of students in an introductory programming course. The method uses internal factors of motivation and learning strategy. It constantly takes student learning behavior on an e-learning site of programming exercises. It assumes each behavior factor appears as total accumulation of effects of multiple internal factors. Based on the assumption, it uses the non-negative matrix factorization (NMF), which numerically approximates decomposition of a matrix showing learning behavior of every student into a product of two matrices. One matrix is a gene matrix indicating how each internal factor affects into individual behavior. The other is a weight matrix representing weights individual students have for each of the internal factors.

The matrix elements have a correspondence to the degree of internal factors that reveal motivation and learning strategies of every student. The method induces the correspondence using initialization of the two matrices. Because personas are similar in successive programming course in a university, and topics to be learned are same in corresponding weeks of identical courses, the method derives initial values for the matrices of every week for current students from the past student data of the corresponding week.

We conducted contextual inquiry on students of the previous course to acquire their motivation and learning strategy comprehensively. The contextual inquiry reveals students strong for every internal factor. It enables us to figure out the objective associa-
tions of internal factors with behavior factors. It also brings the internal factors of each student. Since the internal factors are representative ones over the weeks in the course, we decompose the behavior matrix in every week in the past course into a product of a weight matrix and a gene matrix, with initialization utilizing representative internal factors. The matrices derived in the decomposition represent the correspondence from matrix elements to the internal factors in the past course. The matrices of every week of the previous course are used to initialize those of the corresponding week of the current course. NMF is used again to achieve the precise weight matrix representing learning status of individual students in the current course.

We applied the method to figure out learning status of students in actual C programming course of Ritsumeikan University in 2013. We compare results with learning status obtained by the contextual inquiry conducted after the course. The activeness of individual students can be predicted with more than 70% accuracy in the latter half of the course.

6.3 Future works

NMF decomposes students’ behaviors factors into their motivation and learning strategies factors. The formative assessment which uses NMF does not impose any burdens on the teachers and on the students. It brings the learning status of the students in time.

However, the formative assessment of personas which we have proposed has a weak point. It assumes the current course has settings similar to those of the past one. The result we have presented is the application of the method on the course which has very similar settings with the past one. There is only small difference between the two courses. In actual training situation, the teachers would improve the course settings course by course. If there are significant changes, it is hard to apply two-step NMF as this method did. To avoid the initial values for the weight matrix and the gene matrix using the data
of the past course, we are considering initializing values for the matrices of the students of a week utilizing the data of the previous week.

We have found more than half of the students belonging to the same persona in a week tend to belong to the same persona in the successive week based on their behavior [47]. These students would have the similar movement of persona in their motivation and learning strategies. If we can confirm their internal factor in a certain week of the course, we may be able to figure out the learning status for the following weeks.
Bibliography


[61] Yasuhiro Takemura, Hideo Nagumo, Kuo-Li Huang, Kenichi Matsumoto. Analysis of the Relation between the Teaching Materials and Motivation in Programming


LIST OF PUBLICATIONS

A. Journal


B. Conference


