PAPER *Special Issue on Intelligent CAI and Hypermedia*

An Inductive Student Modeling Method which Deals with Student Contradictions

Yasuyuki KONO[†], *Nonmember*, Mitsuru IKEDA[†] *and* Riichiro MIZOGUCHI[†], *Members*

SUMMARY Student contradictions are the essentials of concepts and knowledge acquisition processes of a student, in the course of tutoring. This paper presents a new perspective to represent student contradictions and a student modeling architecture to capture them. The formulation of a student modeling mechanism enables flexible decision making by using information obtained from students. A nonmonotonic and inductive student model inference system HSMIS has been developed and formulated to cope with modeling contradictions, which basically embodies advanced representation power, sufficiently high adaptability and generality. The HSMIS is evaluated and compared with other representative systems in order to demonstrate its effectiveness.

key words: *modeling and simulation, education, student model, model revision, modeling contradiction, student contradiction*

1. Introduction

A student who is in the fixing stage* of the acquired knowledge often shows contradictory behaviors. This means a student is apt to unstably apply problem solving methods, because he** has not built them or since he has not completed the formulation of related concepts, etc. It is clear that he shows such nonmonotonic learning processes in fixing his knowledge in the process of acquiring correct and stable knowledge. There are other types of contradictions to be considered in designing a student modeling system. A modeling system often gets an answer from him which is consistent with his current belief but inconsistent with his past answers, because he changes his mind nonmonotonically as his learning proceeds.

Contradictions which a modeling system should cope with are classified into the following two types:

1. *modeling contradictions* which should be resolved by revising student model, and
2. *student contradictions* which should be regarded to be inherent in the students and be utilized for education.

The former problem is the essential for modeling processes, because a practical ITS should follow a student's nonmonotonic change. Major requirements for a method dealing with modeling contradictions are as follows:

- revise current student model to enhance its accuracy,
- follow a student's change in understanding,
- ignore his slips, and
- control the load of both him and the system.

To this end, a student modeling system should always make belief revisions to keep data for inference consistent. Surprisingly, only Huang *et al* [8] have tackled this problem, except that Woolf *et al* [21] pointed out the significance. The authors have been tackling this issue and developing an inductive student model inference algorithm HSMIS [16],[17]. The HSMIS employs the ATMS [4] to maintain consistency of the student modeling process. In addition, sophisticated mechanisms to adaptively control the focus of modeling and the reliance on the student are incorporated into HSMIS to meet the last requirement. It enables the HSMIS to ask him questions appropriate in the sense of tutoring.

The second problem, to capture *student contradictions*, seems more important from educational viewpoints. The Socratic method, for example, is a contradiction-based tutoring strategy which teachers especially use to help students in the fixing stage. It is a well-known and already verified method that gives such a student a strong impression that he misapplied his knowledge. Although building high-fidelity student models is an intractable problem [18], an ITS should have a student model which is precise enough to handle tutoring strategies integrated into the ITS [6]. In order to generate sophisticated tutoring behavior like the Socratic method, student modeling methods should be able to cover student contradictions. His knowledge acquisition and fixing processes should be captured by modeling him as he is, even if he has contradictory knowledge [11].

This paper presents a new methodology for handling student contradictions in conjunction with the

Manuscript received July 29, 1993.
Manuscript revised September 21, 1993.

[†]The authors are with the The Institute of Scientific and Industrial Research, Osaka University, Ibaraki-shi, 567 Japan.

*"Fixing stage" means an intermediate learning stage where acquired knowledge is not completely established yet.

**For simplicity "she" is used to refer to the teacher and "he" to refer to the student in general.

Question 1
A sphere has passed through the origin with velocity 19.6m/s to the right at $t = 0$s. It continues linear and uniformly accelerated motion along x-axis of a horizontal plane, and stopped at $t = 2$s. It mass = 2kg. Get its position at $t = 1$s. Determine both the direction and the magnitude of the applied force.
Student's answer
$a = \frac{v_t - v_0}{T_t - T_0} = \frac{0-19.6}{2-0} = -9.8\text{m/s}^2$
$S_t = S_0 + v_0 t + \frac{1}{2}at^2 = 0 + 19.6 * 1 + \frac{-9.8 * 1^2}{2} = 14.7$
The position is 14.7m from the origin.
$F = ma = 2 * -9.8 = -19.6\text{kg·m/s}^2$
It receives 19.6 kg·m/s^2 of force to the left.

Question 2
A sphere has passed through the origin with velocity 19.6m/s to the right at $t = 0$s along x-axis of a horizontal plane. P is moving with linear motion with its velocity decreasing at a uniformed rate. It stopped at $t = 2$s. Determine its position at $t = 1$s.
Student's answer
$S_t = S_0 + vt = 0 + 19.6 * 1 = 19.6$ The position is 19.6m.

Question 3
An sphere is thrown directly upwards with initial velocity 19.6m/s at t=0s. It reaches at the maximum hight at t=2s. Determine the direction of the applied force at t=1s.
Student's answer
$v_t = \frac{v_{t'} - v_0}{T_{t'} - T_0} * t + v_0 = \frac{19.6-0}{2-0} * 1 + 0$
$= 9.8\text{m/s}$
It is still moving upward at $t = 1$s.
Therefore, it is still receiving upward force at $t = 1$s.

Fig. 1 Examples of behavior of a student who has undifferentiated concepts.

Socratic tutoring. There seems to be some dilemma between a method to capture student contradictions as they are and one to resolve modeling contradictions. The authors successfully integrated the two methods by introducing a multi-world mechanism into the HSMIS architecture. In the reformulated HSMIS, the ATMS plays another important role of managing multiple worlds which model students with contradictions.

2. What is Student Contradiction?

Assume that a student is in the stage of acquiring a certain new concept and that he has not fully discriminated it from other related concepts he has already acquired. Such a student is apt to behave unstably in applying knowledge to solve problems which contain the undifferentiated concept. Figure 1 indicates the behavior of a student who has not yet differentiated concepts, those are the concept of "uniform motion" and the concept of "uniformly accelerated motion." He correctly calculated the position of P in Question 1, which specifies the type of motion as "linear and uniformly accelerated motion." In Question 2, however, he mistook a uniformly accelerated motion for a uniform motion, and so applied problem solving knowledge for uniform motion. Such a situation occurs due to his confusion between the two concepts. As a result, his problem solving ability becomes unstable.

A student can choose certain problem solving methods appropriate for the problem given, if he has well-discriminated concepts and has adequate knowledge of their attributes. If he has not, however, he might misapply a procedure which belongs to another world by taking no notice of particular attributes of the problem. For instance, methods to "calculate the position of a moving object" are associated with both concepts, such as uniformly accelerated linear motion ($S_t = S_0 + v_0 t + at^2$ or $S_t = S_0 + (v_0 + v_t) * t/2$) and uniform motion ($S_t = S_0 + vt$). In solving Question 2 in Fig. 1, he retrieves the method defined in the concept of uniform motion, while he should apply the method defined in the concept of linear and uniformly accelerated motion.

A more interesting example of student contradiction is found in Question 3 in Fig. 1. The student who had correctly calculated the force that the sphere receives in Question 1, but could not determine the correct direction of the force to Question 3 in spite of that the two motions are physically identical except for the direction of the motion.

Such conflicts among his answers suggests the "multi-world reasoning" assumption that he partitions his whole storage and reasoning space into many ones. Each small partition in his reasoning space with relevant storage is called a "world." He stores problem solving methods and rules which he can handle at once in each world. He can retrieve and utilize these methods in a certain world, only when he makes inference in the world. A contradiction can be found when he utilized two different worlds in solving problems. One is the world of well-formalized physics for Question 1 in which he stores the knowledge learned through the curriculum of physics, e.g., formulas, definitions, and another is his naive physical world for Question 3 which he has been deeply engraved on his memory since his childhood, for instance. He has a "motion implies a force" misconception [3] in the naive one in this case. It is inconsistent with the knowledge for "uniformly accelerated motion" in the well-formalized one. He has answered that the force is directed upward because he used the naively misconceptualized world.

"Student contradictions" are defined in this paper as his status which causes behaviors which can be regarded as a contradiction viewed from the stand point of an observer. Typical interpretation of contradiction is as follows:

- He places more than two series of problem solving methods, which is originally placed in different worlds of concepts, in the same world regardless of their attributes. This is caused by his failure in differentiating them from each other.

- He answers different truth values to a certain fact within a limited time, since his knowledge is unstable.

3. Contradictions in Student Modeling

This section discusses contradictions in student modeling process by clarifying human concept discrimination structures and problem solving method retrieval in respective reasoning worlds.

A student's answer to a question is represented by a pair of a fact and its truth value, and is called an oracle. The set of oracles acquired by the observation of student's behavior within a certain period of time tends to be inconsistent for several reasons. Such student's inconsistent behavior is classified into the following three types of contradictions:

(1) *Oracle contradictions caused by change of student's mind*: Student's learning process is essentially attained with change of his mind. The consistency of his answers can be easily lost, because he behaves based on his current knowledge independent of his previous knowledge.

(2) *Oracle contradictions caused by slips*: A student often makes careless mistakes. Oracles based on them are inconsistent with his actual knowledge, hence, the set of oracles that contains slips is inconsistent.

(3) *Student contradictions*: A student sometimes has inconsistent knowledge in his head which also causes contradictory oracles.

In cases of type (1) and (2), a modeling system is able to construct a consistent student model that represents the current student knowledge by revising the set of oracles appropriately to resolve the contradictions, since we can assume student knowledge is consistent on each period in this case. In cases of type (3) of contradictions, however, Student's knowledge itself is inconsistent. Therefore, any models, that are represented and applied in a single reasoning space, can not completely capture his status. A multi-world formalization of a student model should be introduced to cope with this problem.

It can be explained that human beings partition their whole storage and reasoning space into multiple "worlds" and organize a kind of structure to retrieve their knowledge efficiently and with less load by

1. retrieving which world (concept) the given problem belongs to along a certain discriminating structure at first, and

2. retrieving a method that contributes to problem solving in the discovered world,

as mentioned in Sect. 2. The first element, i.e., decision of the target world, can be regarded as a search on a *concept discrimination tree* from its root, a node of which corresponds to a concept. The given problem is articulated into a vector of primitive attributes, which decides the conceptual world the problem belongs to by seeking on the concept discrimination tree. To go forward through a path from one conceptual node to another requires to satisfy some conditions which characterize the destined node as conditions C_0, C_1 and C_2 in Fig. 2. For instance, the node "linear motion" has children nodes "uniform motion" and "uniformly accelerated motion." A motion which belongs to the node "linear motion" also belongs to the node "uniform motion", if the motion satisfies discrimination conditions "the velocity is fixed" or "the acceleration is zero," etc. Problem solving methods which conventional modeling systems have handled is retrieved in the decided world and executed.

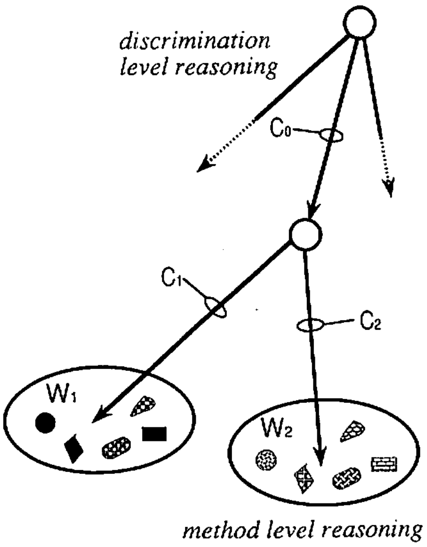


Fig. 2 Concept discrimination tree.

Student contradictions are closely related to such concept discrimination structures and reasoning mechanisms as mentioned above, therefore, they can be well formulated by the "multi-world reasoning" assumption. The status of a student who has not yet discriminated two concepts, for instance "uniform motion" and "uniformly accelerated motion", can be captured as not having built such discrimination conditions. The lack of discrimination conditions easily causes student's confusion of concepts as lack of C_1 and C_2 causes his confusion of methods in Fig. 3. Such a student is apt to unstably misapply methods which should belong to a different concept from the expected one. He conversely slips up to seek the naive world in the case of Question 3.

Student contradictions are caused by erroneous concept discrimination trees, because a student who has such erroneous tree cannot manage consistency in retrieving problem solving methods as mentioned above.

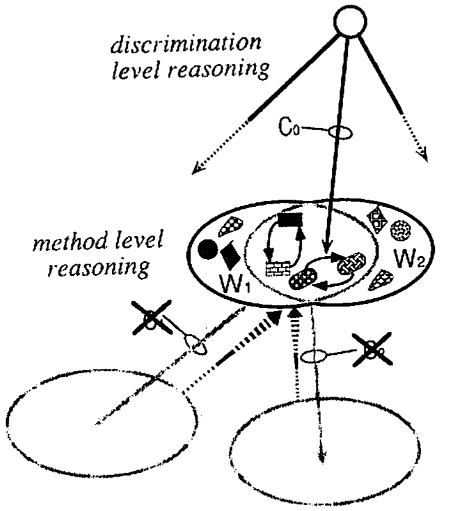


Fig. 3 Concept discrimination tree of a student who has undifferentiations.

Such kind of contradictions can be represented by the overlay of discrimination conditions.

Student modeling process is essentially hypothetical, hence, the completeness of inferred student model is not always guaranteed. The expectation of student's answer deduced from the current student model is often different from newly obtained oracles, when the current model does not completely represent his current status. Assumptions, which were assumed when the current model was inferred, become inconsistent with the set of oracles. Such a type of contradictions is called "assumption contradiction in modeling." They are able to be formulated in a similar way of types (1) and (2) of contradictions, because all of them should be resolved to represent consistent knowledge inside a single world. To resolve such contradictions requires a nonmonotonic process which is essential to student modeling processes, i.e., to assume the student's knowledge which satisfies the given set of oracles, and to revise the current model going together with changes of oracles, etc. "Modeling contradiction" is a general term of such three types of contradictions which occurs in a single world. We can design a student modeling architecture that is able to maintain the consistency between the student model and the student behavior by defining methodologies to resolve them. The SMDS, subsystem of the HSMIS, is able to algorithmically backtrack the contradiction in the student model and oracles. The HSMIS has inherently the capability to adaptively revise the current model in cases of assumption contradictions. Furthermore, the HSMIS can cope with contradictions (1) and (2) by nonmonotonically revising the assumptions of the ATMS, which are roots of all its modeling processes. Such a mechanism to deal with modeling contradictions is described in Sect. 4.

It is difficult for not only modeling systems but also human teachers to distinguish and detect the four types of contradictions, i.e., type (1), (2), (3) and assumption contradictions in modeling, because all of their indications are very similar. They are triggered by a difference between the expectation of student answer deduced from the current student model and his actual answer. One of the research goals of this paper is to produce a generic and formalized modeling mechanism which is able to cope with these kinds of contradictions. Although a generic methodology to distinguish them is not fully developed, some heuristics are employed as shown below.

Assume that the reliability of each given oracle or each clause in the student model can be available. Both modeling contradictions and student contradictions are detectable by quite similar triggers, i.e., the expectation from the model and the actual oracles. Contradiction resolving procedures of those contradictions are quite different from each other. Modeling contradictions require to be resolved by revising the set of oracles or current model in general. Contradiction resolution procedure for each type of modeling contradiction is a bit different, and hence detection processes of them are different from each other. In the heuristics, student contradictions are first distinguished from modeling contradictions.

Student contradictions should not be resolved, because student's inconsistencies should be modeled as he is. Student contradictions require to revise neither oracle set nor clauses that are inconsistent with oracles, but to revise discrimination structure to permit to contain the inconsistency in it. Such a difference in treatment of student contradictions and modeling contradictions suggests the following way of discriminating them. If either the reliability of a clause which is inconsistent with valid oracles or that of the oracles is less than a certain threshold, the inconsistency should be considered to be a modeling contradiction and hence should be resolved. On the other hand, if both of the reliabilities are high enough, the inconsistency is considered to be a student contradiction. They are not revised but put into some worlds, i.e., all the reliable data can be alive in the multi-world formalization. The following heuristics to detect contradictions of each sub category in modeling contradictions are incorporated.

The change of student's knowledge which causes type (1) of modeling contradictions occurs especially right after his errors are corrected. He then generally changes his understanding from erroneous status to correct one. It is appropriate to apply revision procedures for type (1), when correct oracles are obtained right after tutoring, i.e., the system resolves the contradiction by excluding the past oracles inconsistent with correct clauses, or by asking him truth values of the oracles. The revision of oracles results in the revision of the model, i.e., erroneous clauses are dismissed and correct clauses are appended. In addition, it is available to directly ask him if he has changed his knowledge.

Independently of the correctness, generally speaking, the student ought to have consistently applied the clauses that are inconsistent with newly obtained oracles throughout a certain period, in the case that he makes careless mistakes which cause type (2) of modeling contradictions. Thus such type of contradictions are detected in similar criteria as those for student contradictions, i.e., the inconsistent oracles and clauses would be both reliable enough. There are two ways to distinguish them; one is to consider a situation as a type (2) only when the situation could not be treated as a student contradiction, and another is to ask him a very similar question to get a confirmation.

These contradictions can be more sufficiently distinguished by introducing and enriching domain dependent heuristics, e.g., "Students tend to mistake a uniformly accelerated motion for a uniformed motion if the motion is vertical," in addition to the domain independent heuristics explained above.

4. Nonmonotonic Student Modeling

4.1 SMIS: Inductive Inference Engine

A student model description language is required to be able to represent the teacher's understanding of the student's knowledge. From this viewpoint, the language should take four truth values for a statement, those are, true, false, unknown and fail. The SMDL, which is an extended version of Prolog, is designed to treat these truth values.

Student modeling is essentially an inductive inference. The SMIS, an inductive inference engine that the authors have developed based on MIS [20], describes student's knowledge as an SMDL program from a set of oracles. An oracle symbolizes a student's answer which takes the form of a fact and its truth value. SMIS applies the following procedure repeatedly to the model:
(1) If there is a difference between an oracle and the fact derived from a student model, activate the student model diagnosis system, SMDS, to identify the cause of the difference.
(2) According to the diagnosis, SMIS selects an appropriate operation, either removal of an inappropriate clause or addition of a new clause.
Generality of the student modeling method with SMIS is sufficiently high, because it can construct models which can be described in terms of SMDL. Further details are described in [10] and [12].

4.2 Control of Model Building Process

A student modeling system should embody a teacher's educational insights. Requirements to the architecture of the modeling system from educational viewpoints are in the following:

1. To follow a student's change in understanding, and
2. to ask questions with regard to their appropriateness in the sense of tutoring.

The SMIS automatically asks questions that contribute to disambiguation of alternative model selection. When the teacher is confident that she has a good grasp of her student's knowledge, she asks him fewer questions on his behavior as long as it supports her confidence. The model inference procedure should ask questions with regard to their appropriateness in the sense of tutoring.

These two problems mentioned above suggest that the inference procedure should cope with nonmonotonic modeling processes. In an HSMIS, an ATMS is employed for this purpose. The HSMIS consists of the SMIS, the ATMS, the Virtual oracle generator and the CRS. The main task of the ATMS is to manage the consistency of a set of assumptions (environment) used by the problem solver, the SMIS in our case. The SMIS informs the ATMS of all the reasoning processes. The control mechanism of student modeling is newly formulated, in order to make the quality and quantity of questions reasonable in the sense of tutoring. In this formulation, the following three types of nonmonotonicities, with which a modeling system should cope, are listed:

(1) student nonmonotonicity, which is explained above,

(2) topic nonmonotonicity which is related to both topics for model diagnosis and questions, and

(3) reliance nonmonotonicity which is related to the reliance on a student's knowledge.

Oracles are justified by three kinds of assumptions such as (1)*student*, (2)*consider* and (3)*trust* in order. The status of each oracle is controlled from viewpoints of these nonmonotonicities. Further details of the formulation are given in [12] and [10].

An example of a modeling conversation and student model constructed by the HSMIS is indicated in Fig. 4. This geographic domain is selected, because it can notably demonstrate the efficiency of the control mechanism. In the conversation (a), the system obtains an oracle `grow(rice,osaka)::true`. Since the oracle is correct, the HSMIS trusts that the student has correct knowledge about the growth of plants, which corresponds to the clause (1). Therefore, the system makes virtual oracles, which are correct facts
`suitable_temperature(rice,osaka)` :: `true`,
`suitable_soil(rice,osaka)` :: `true`,
`suitable_lay(rice,osaka)` :: `true`, and
`has_irrigation(osaka)` :: `true`,
to induce the correct clause (1) without making questions on them.

(a) Does rice grow in Osaka ?
>> Yes.[Yes]
(1)
(b) Does rice grow in Kiev ?
>> Yes.[No]
(2)
(c) Does rice grow in Mongolia ?
>> Yes.[No]
(d) Is the temperature of Mongolia suitable for rice ?
>> Yes.[Yes]
(e) Is the soil of Mongolia suitable for rice ?
>> Yes.[Yes]
(f) Is the lay of Mongolia suitable for rice ?
>> Yes.[Yes]
(g) Does Mongolia have an irrigation ?
>> No.[No]
(3)
(h) Is the temperature of Kiev suitable for rice ?
...
...

(Student's answers are underlined. Right ones are in [].)

```
                    (1)
grow(Plant,Place) :: -
        suitable_temperature(Plant,Place),
        suitable_soil(Plant,Place),
        suitable_lay(Plant,Place),
        has_irrigation(Place).

                    (2)
grow(Plant,Place) :: -
        suitable_soil(Plant,Place),
        suitable_lay(Plant,Place),
        has_irrigation(Place).

                    (3)
grow(Plant,Place) :: -
        suitable_soil(Plant,Place),
        suitable_lay(Plant,Place),
        has_irrigation(Place).
```

(Model at each scene of conversation.)

Fig. 4 An example of modeling conversation and model at each scene of it.

In the conversation (b), the student gives a wrong answer to the system's question. As the corresponding oracle can be explained by correct facts on Kiev and the prepared clause (2), which implies that the student lacks the originally needed condition suitable_temperature(Plant, Place), the system makes virtual oracles on Kiev to derive the clause as in the previous conversation. Virtual oracles on Osaka continue to exist and support the clause (2). In the conversation below (c), the system fails to explain the wrong oracle grow(rice,mongolia) :: true by any of the plausible clauses which are prepared in advance. Therefore, the system must obtain "real" oracles on Mongolia. Moreover, virtual oracles on Kiev and Osaka are dismissed, because he betrays the system's trust in clause (2). The system makes questions on Kiev to get "real" oracles but does not make questions on Osaka, because the system avoids touching the old subject of Osaka by changing the status of assumption *consider*(*Osaka*).

5. Student Contradiction

5.1 Formulation of Student Problem Solving

Student problem solving which includes the latter two phases mentioned in Sect. 2 can be represented using a logic-based language such as Prolog. Predicate solve($G, \tilde{X}_{\mathrm{in}}, \tilde{X}_{\mathrm{out}}$) denotes problem solving knowledge. G denotes the goal of the problem, that is, what should be determined under what constraints. $\tilde{X}_{\mathrm{in}}$ is a vector of input variables which are instantiated and $\tilde{X}_{\mathrm{out}}$ is a vector of output variables which are not instantiated when the predicate is called. $\{\tilde{X}_{\mathrm{in}}, \tilde{X}_{\mathrm{out}}\}$ represents whole articulation of the problem space. For instance, the problem space "motion" is represented as $\{m, (s(t), A_s), (v(t), A_v), (a(t), A_a), (f(t), A_f), [(T_0, S_0, V_0, a_0, F_0), \cdots]\}$, where the elements are the mass of the moving object, displacement, velocity, acceleration and applied force as functions of time erasped, and sets of the elements of the motion, respectively. Each function of time erasped is denoted as a couple of the function itself and the attribute of the function. The problem space, which is adopted in Question 1 in Fig. 1, is represented as $\{m, (s(t)), (v(t)), (a(t), \mathit{fixed}), (f(t)), [(2, S_0, (0,0), a_0, F_0), (0, (0,0), (19.6, 0), a_1, F_1)]\}$. When the problem solving begins, input variables are given in the formula of instantiated variables. For instance, the displacement and the velocity on $t = 0$ are instantiated as $(0, 0)$ and $(19.6, 0)$, because they are given in the problem. Problem solving is a retrieval and an execution of methods described in the problem solving knowledge base, and to get the output parameters list $\{\tilde{X}_{\mathrm{out}}\}$ instantiated from the given input parameters list $\{\tilde{X}_{\mathrm{in}}\}$.

A problem solving knowledge consists of the estimating part and the procedural part, and is represented by the following formula:

```
solve(G, X̃in, X̃out) :-
        predicates_for_estimation(X̃in),
        procedural_bodies(G, X̃in, X̃out).
```

The estimating part specifies relevant domain of the knowledge, and the procedural part describes the method used to get to the goal. Predicates for estimation are called "*world predicates*," which correspond to discrimination conditions on a concept discrimination tree. Conventional bugs are defined as wrong predicates, lack of predicates and additional predicates from the correct problem solving knowledge in the procedural part. An undifferentiation of concept can be modeled as a lack of world predicate condi-

```
solve(get_position_of_moving_object(T1,S1), M,S1,V1,A1,F1, [(T0,S0,V0,A0,F0),(T1,S1,V1,A1,F1)]) :-
        solve_in_formulated_physics ({Xin}),
        uniformly_accelerated_motion ({Xin}),
        get_position_of_moving_object ([A1], {Xin}, {Xout} ).
    get__position_of_moving_object ([A1], {Xin}, {Xout}) :-
        get_acceleration ([A1], {Xin}, {Xout}),
         subtract (T1,T0,T), multiply (V0,T,V1), squere (T,TT). multiply (A0,TT,ATT). add (S0,VT,ATT,S1).
```

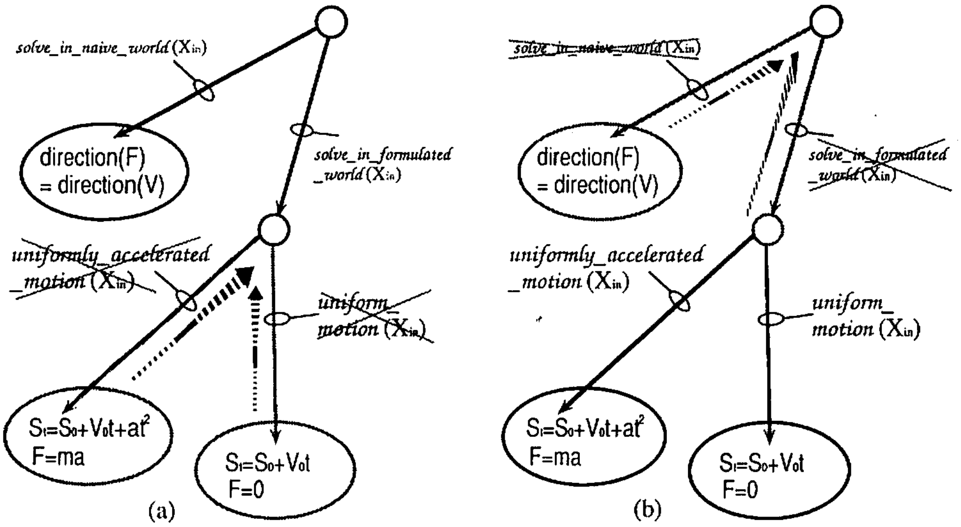

solve_in_naive_world (Xin)
direction(F) = direction(V)
solve_in_formulated_world (Xin)
~~uniformly_accelerated_motion (Xin)~~
~~uniform_motion (Xin)~~
S1=S0+V0t+at²
F=ma
S1=S0+V0t
F=0
(a)
~~solve_in_naive_world (Xin)~~
direction(F) = direction(V)
~~solve_in_formulated_world (Xin)~~
uniformly_accelerated_motion (Xin)
uniform_motion (Xin)
S1=S0+V0t+at²
F=ma
S1=S0+V0t
F=0
(b)

Fig. 5 Interpretation of the student contradiction.

tions. Lack of a world predicate extends the coverage of concept, so that the knowledge, which should be inapplicable outside a certain concept, becomes incorrectly applicable in other concepts. As a result, both the original problem solving knowledge and the knowledge with wrong predicates for estimation are applicable in some different concepts. The knowledge which he actually applies changes according to circumstances in such cases. Therefore, the status of the knowledge applications can be observed to be unstable. Teachers recognize these circumstances as contradiction. The correct problem solving knowledge to determine the position of P in Fig. 1 is described as clauses in Fig. 5. Conditions *solve_in_formulated_physics* and *uniformly_accelerated_motion* are world predicates, and the last predicate in the body of the former clause belong to the procedural part, which calls the problem solving method, i.e., the latter clause. A student who has not completely formed the concept of "uniformly accelerated motion" lacks the latter world predicate as depicted in Fig. 5 (a), so that he can apply the knowledge even in the case of uniform motions. Thus he unstably chooses the correct method for calculating the position or the incorrect one, because of the lack of the world predicate having no definite view of his own.

5.2 Modeling Student Contradictions

The formulation of student contradictions described in Sect. 5.1 works well as a student modeling method by utilizing the heuristics in Sect. 3. A Multi-World Controller is incorporated into HSMIS to control multi-world reasoning. Concept discrimination trees are given in advance as a part of domain knowledge, and model diagnosis and revisions are driven on the structure of these trees. Each world corresponds to an ATMS environment in the integrated HSMIS. Clauses for problem solving are induced in each world from oracles belonging to the world. The Multi-World Controller manages the set of worlds, each of which is a certain ATMS environment. The ATMS makes the rounds of such worlds and SMIS induces the clause level student model of the world. When a modeling contradiction is detected and is resolved in a certain world, the corresponding elements of the set of environments are revised. Each clause level student model can be consistently inferred using such a mechanism.

The construction process of the student model that represents student contradictions is as follows:

1. The system assumes a student contradiction, when it turns out that a reliable clause has to be deleted to explain new oracles.

2. The system tests whether the oracles are satisfied by the clauses that *exist* in another world by using the world in order of similarity to the correct world on the structure of the tree.

3. When a satisfiable world is found, discrimination conditions that contributes to differentiation of the

two worlds are marked as neglected as depicted in Fig. 5.

4. If any satisfiable worlds are not found, the system considers the situation as a modeling contradiction and tries to revise the model in the correct world.

I_W, a set of instances each of which should originally belongs to a certain conceptual world W, can be declared by applying C_W which is the world predicate of W to I, the whole set of instances in the domain. Method level representations and oracles which justify the model are generated and stored in each world individually. Suppose that there are two worlds W_1 and W_2 which are brothers and that they have already obtained and involved oracle sets $\tilde{O}_{W1}$ and $\tilde{O}_{W2}$. Methods $\mathcal{M}_1$ and $\mathcal{M}_2$ have been staying in W_1 and W_2, respectively. If $\mathcal{M}_1$, that currently has high CF value, is denied by the oracle set $\tilde{O}_p$, which is newly obtained from him, to the problem that naturally belongs to W_1, then the system hesitates to dismiss $\mathcal{M}_1$ and tries to interpret the status of his conceptual discrimination as undifferentiated. If $\mathcal{M}_2$ satisfies $\tilde{O}_p$, $\tilde{O}_p$ is moved into W_2 and supports $\mathcal{M}_2$ there. Discriminal conditions C_1 and C_2, which are world predicates of W_1 and W_2, are marked as neglected. If any methods in any other worlds in the tree except the "another world" do not support $\tilde{O}_p$, it is assumed that he was thinking in the world that contains naive buggy knowledge and has solved the problem unformulatedly. $\tilde{O}_p$ is moved- into the "another world" which is prepared to cover his unformulated problem solving, if one of the buggy clauses, prepared in the world in advance, satisfies $\tilde{O}_p$.

In our example, the student correctly answered Question 1 in the past, so that the student model had the correct clause to get the force which an object in uniformly accelerated motion receives. He made the wrong answer utilizing his naive physical world to Question 3 later. The HSMIS receives oracles made from his answer, e.g., oracle(subtract(19.6,0,19.6),true), oracle(subtract(2,0,2),true). The above clause in the student model does not satisfy these oracles and derives an answer different from that of him, so that HSMIS engages in generating another clause whose head is solve(get_direction_of_force(···), ···). It is inappropriate to dismiss the existing clause from the student model, however. The clause is reliable enough and it is hard to interpret his behavior as lacking in it, because previously he was able to apply it to obtain a correct answer.

In such cases, the HSMIS dares to interpret his concepts as undifferentiated, attempting to make a student model which implies the student contradiction. To this end, the HSMIS at first infers the method that the student applied from the above oracles in a data driven way as usual: subtract(V_1, V_0, DV), subtract(T_1, T_0, DT), divide(DV,DT,AV) and direction(AV, D_f), in order. Next, it searches a world which has already contained the method. If the search fails, model inference is carried out in the "another world." The HSMIS searches the existing clause in the world which satisfies his erroneous answer. The system finds out the clause which represents his erroneous "motion implies force" misconception, so that it marks discrimination conditions that can naturally partitions the concept of motion into the world of formulated physics and the world of naive physics as neglected. It can explain his discrimination status, that he unstably applies physical formulas and naive methods.

5.3 Correcting Student Contradictions

To help the student build naive physical world in his brain is one of the essential goals of tutoring, especially for helping his conceptualization. This suggests a very effective tutoring behavior as follows:

1. Give him a problem such that he tries to solve in his naive one and fails to get a correct solution.
2. Remind him of the correct answer to the problem he obtained in Question 1.
3. Point out the inconsistency between the two results.
4. Explain the causes and guide him to build a correct concept.

In this way, he can correctly identify attributes necessary for building the concept, which we call world predicates, and establish relationships between them, thus he can appropriately conceptualize the knowledge in both worlds.

Tutor: Solve this problem. (Give Question 1 again)
The student correctly answers.

Tutor: You answered in Question 3 that the direction of the force which the sphere receives is upward at t=1s, because it is still moving upwards then. If that was correct, why didn't you say that the sphere in Question 1 receives force to the right?

Student: Because it was moving upwards, so it is hardly possible that it continued receiving force downward.

Tutor: The two problems are completely the same, e.g., speed at each time, etc., except for the direction of motion. If direction of force were to be implied from motion, you should have naturally said that the direction of the force is to the left in Question 1, but you didn't. (You should have ''motion implies force'' misconception in your naive physical world. ···)

Fig. 6 An example of tutoring behavior.

Obtaining a student model that represents the student contradiction, in this case the cause of it is represented by the lack of the condition "*solve_in_naive_physics*", the system becomes able to generate a effective tutoring dialogue as Fig. 6. His incorrect reasoning methods, such as the use of abduction in his problem solving, can be formalized as contradictions, however, the topic has been kept for future work.

The mechanism for modeling contradictions explained in Sect. 4 and the framework to handle student contradictions explained in the previous section are integrated into the reformulated HSMIS. It gives up constructing a unified consistent model and dares to build a model in multiple spaces, when newly obtained data denies the reliable current model. The control mechanism guarantees the fidelity and the accuracy of the model in each world.

6. Discussions

Here discuss some basic issues of comparative student modeling methodologies.

The Overlay model [2] is relatively easy to control the modeling processes with. Since it cannot represent the student's incorrect knowledge, the performance of overlay-model-based ITSs is limited. IDEBUGGY [1] has a lot of modular chunks of buggy procedures. It has the capability to cope with the noise problem, however, it is not complete. Its searching strategies depend on the assumption that each individual buggy procedure can be extracted from a combined buggy procedure. The burden on cataloging bugs is still left.

Both ACM [14] and our method HSMIS are based on the idea that student modeling is viewed as inductive learning from a set of examples. They can model not only mal-functions but *mal-structures* in terms of declared primitives. Both methods act as domain-independent engines and their capabilities contain those of both the overlay and buggy methods. Besides, both of them can cope with noisy data.

The major difference is their searching and diagnostic strategies for ascertaining what part of the model conflicts with the student. HSMIS can prune search space for new clauses in a directed tree, although ACM originally makes a blind search of all its search space. To embody efficient searching, Langley *et al* [15] have been developing DPF (Dynamic Path Finder).

The HSMIS can incrementally revise the model following newly obtained data. Although the path-finding algorithm is relatively formalized by DPF, the rule-finding process is not formally defined. HSMIS can distinguish between noisy data and bug migration [12], while ACM cannot currently [15]. HSMIS asks questions which are logically required in consideration of educational appropriateness, while ACM does not.

It can be claimed that it is significant for student modeling systems to handle inductive learning (inference) and nonmonotonic reasoning [16], [17]. Woolf *et al* [21] also pointed out the significance recently. Incorporation of inductive learning theories into student modeling is discussed by Hoppe [7] and Dillenbourg [6], which utilize EBG and LEX, respectively. Both systems are successful to handle their simple didactic strategies.

Self [19] gives the characterization of student modeling as a diagnostic task, making an application of GDE [5]. The knowledge to be learned is expressed as a system with a defined structure of components with defined behavior. GDE generates hypotheses of student's misconceptions. Each hypothesis is represented as a set of failed components. GDE identifies and refines the set of hypotheses consistent with the observations thus far. However, GDE has a fatal problem. It can express only mal-functions.

Huang [9] proposes a logic to capture the student's inconsistent knowledge. Surprisingly, only his study, except the HSMIS, has so far been made at modeling student contradictions. Although it is well-defined in terms of propositional calculus, it would be difficult to extend it to deal with the first order predicate calculus.

7. Concluding Remarks

This paper has presented a student modeling methodology and its use in an ITS. Contradictions of the student to model students with contradictory knowledge are first defined. The modeling system of such students has to model undifferentiated concepts and inconsistency as it is. A sophisticated control mechanism to deal with modeling contradictions for the HSMIS has been developed.

Next, HSMIS has been compared with other modeling systems from various viewpoints to demonstrate it is a well-defined and generic student modeling algorithm, which can build a student model of high representation power. The HSMIS has been fully implemented in Common-ESP(Extended Self-contained Prolog), and embedded in FITS, Framework for ITS. Two ITSs have been built using FITS, one is on geography and the other is on chemical reactions.

Acknowledgments

The authors are greatful to Geoffry Webb for his comment which motivated this research. The authors are also thankful to Heinz Ulrich Hoppe for his valuable comments. This work is supported in part by Grant-in Aid for Scientific Research on Priority Areas of the Ministry of Education, Science and Culture of JAPAN under Grant: (No.03245106).

References

[1] Burton, R.R., "Diagnosing bugs in a simple procedural skill", In Sleeman, D., et al. (eds.), *Intelligent Tutoring Systems*, Academic Press, London, 1982.

[2] Carr, B. and Goldstein, I., "Overlays:A theory of modeling for computer aided instruction", *MIT AI Memo*, 1977.

[3] Clement, J., "Students' preconceptions in introductory mechanics", *Am. J. Phys.*, vol.50, pp.66–71, 1982.

[4] de Kleer, J., "An assumption-based TMS", *Artificial Intelligence*, vol.28, pp.127–162, 1986.

[5] de Kleer, J. and Williams, B.C., "Diagnosing multiple faults", *Artificial Intelligence*, vol.32, pp.97–130, 1987.

[6] Dillenbourg, P., "Designing a self-improving tutor: PROTO-TEG", *Instructional Science*, vol.18, pp.193–216, 1989.

[7] Hoppe, H.U., "An annalysis of EBG and its relation to partial evaluation", *Arbeitspapiere der GMD*, 1991.

[8] Huang, X., McCalla, G.I., Greer, J.E. and Neufeld, E., "Revising deductive knowledge and stereotypical knowledge in a student model", *User Modeling and User-Adapted Interaction*, vol.1, pp.87–115, 1991.

[9] Huang, X., "Inconsistent beliefs, attention, and student modeling", *J. AI in Edu.*, vol.3, no.4, pp.417–428, 1993.

[10] Ikeda, M., Kono, Y. and Mizoguchi, R., "Nonmonotonic model inference", *Proc. IJCAI'93*, Chambery, France, pp.467–473, 1993.

[11] Kono, Y., Ikeda, M. and Mizoguchi, R., "To contradict is human", In Frasson, C., et al. (eds.), *Intelligent Tutoring Systems*, ITS'92 Proceedings, pp.451–458, Springer-Verlag, 1992.

[12] Kono, Y., Tokimori, T., Ikeda, M., Nomura, Y. and Mizoguchi, R., "A student model building method based on formalization of nonmonotonicity", *J. JSAI*, vol.8, no.4, 1993, (In Japanese).

[13] Kono, Y., Ikeda, M. & Mizoguchi, R., "A modeling method for students with contradictions", *Proc. AI-ED 93*, Edinburgh, Scotland. pp.481–488, 1993.

[14] Langley, P. and Ohlsson, S., "Automated cognitive modeling", *Proc. AAAI-84*, pp.193–197, 1984.

[15] Langley, P., Wogulis, J. and Ohlsson, S., "Rules and principles in cognitive diagnosis", In Frederiksen, N., et al. (eds.), *Diagnostic Monitoring of Skill and Knowledge Acquisition*, pp.217–250, Lawrence Erlbaum, 1990.

[16] Mizoguchi, R., Ikeda, M. and Kakusho, O., "An innovative framework for intelligent tutoring systems", *Proc. IFIP TC3 Working Conference on AI Tools in Education*, pp.105–120, Fascati, Italy, 1987.

[17] Mizoguchi, R. and Ikeda, M., "A generic framework for ITS and its evaluation", In Lewis, R., et al. (eds.), *Advanced Research on Computers in Education*, pp.63–72, North-Holland, 1991.

[18] Self, J.A., "Bypassing the intractable problem of student modelling", *Proc. ITS'88*, pp.18–24, 1988.

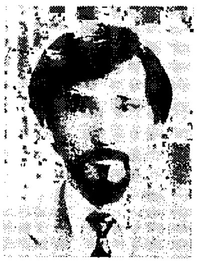
[19] Self, J.A., "Supporting the disembedding of learning", *Proc. East-West Conference on Emerging Computer Technologies in Education*, pp.281–286, 1992.

[20] Shapiro, E.Y., *Algorithmic Program Debugging*, MIT Press, 1982.

[21] Woolf, B.P. and Murray, T., "Using machine learning to advice a student modeling", *J. Artificial Intelligence in Education*, vol.3, no.4, pp.401–416, 1993.

Yasuyuki Kono was born in Hyogo, Japan, on January 8 1967. He received B.S. and M.S. degrees from Osaka Univedsity, Osaka, Japan, in 1989 and 1991, respectively. He is currently a Ph.D. student at the Department of Information and Computer Science, Faculty of Engineering Science, Osaka University. He is engaged in research on student modeling for intelligent tutoring systems. Mr. Kono is a member of Japanese Society for Artificial Intelligence (JSAI), Information Processing Society of Japan (IPSJ), and Association for the Advancement of Computing in Education (AACE).

Mitsuru Ikeda was born in Aomori, Japan, on March 29, 1962. He received B.S. and M.S. degrees from Utsunomiya University, Tochigi, Japan, in 1984 and 1986, respectively. He obtained a Ph.D. of Eng. degree from Osaka University in 1989. From 1989 to 1991 he was a Research Associate in the Faculty of Engineering, Utsunomiya University. He is currently a Research Associate in the Institute of Scientific and Industrial Research, Osaka University. He has been engaged in research on formal language theory, hypothetical inference, inductive inference, and intelligent tutoring systems. Dr. Ikeda is a member of JSAI, IPSJ, CAI Society, IEEE, and AAAI.



Riichiro Mizoguchi was born in Tokyo, Japan, on October 13, 1948. He received B.S., M.S., and Ph.D. of Eng. degrees from Osaka University in 1972, 1974, and 1977, respectively. In 1977 he joined the Osaka Electro-Communication University. From 1978 to 1986 he was a Research Associate in the Institute of Scientific and Industrial Research, Osaka University, where he is presently a Professor. He has been engaged in research on learning of pattern discriminating function, cluster analysis, speech analysis/recognition/understanding, expert systems, and intelligent tutoring systems. He received paper awards from Pattern Recognition Society and IEICE of Japan in 1983 and 1987, respectively. Dr. Mizoguchi is a member of JSAI, IPSJ, Acoustical Society of Japan, CAI Society, Japanese Society for Cognitive Science, AACE, IEEE, and AAAI.