PHYSIMC
An Intelligent Assistant for Case-Based Learning

Jane Yung-jen Hsu   Chien-Jung Ting
Department of Computer Science and Information Engineering
National Taiwan University Taipei, Taiwan 106, R.O.C.

Abstract

This paper presents the design and implementation of PHYSIMC, which supports *case-based learning* of elementary physics in a computer-assisted simulation environment. The PHYSIMC system facilitates physics problem solving by providing 1) a user-friendly interface for problem specification via direct manipulation of physical objects, 2) 2-D motion simulation of primitive physical objects, 3) a case library of successful problem-solving episodes, and 4) a browsing tool for relevant problems and their corresponding solutions. As a result, a student can make use of past problem-solving experiences in attempting to solve a new problem. In a knowledge-based simulation environment, such case-based learning tools help narrow the gaps due to incomplete domain knowledge.

1 Introduction

Modern computer technologies have helped create a wide range of computer-aided instruction (CAI) software. Many current CAI systems present textbook knowledge through a multimedia interface so as to boost a student's interests in the subject matter. Examples and tests are usually extracted from a prestored database of problems based on certain instructional strategy. With proper student modeling, the problem sets can be further tailored to the level of focus, motivation, and comprehension of the individual student.

1.1 Motivation

In most CAI systems, the learner plays a passive role in her interaction with the computer. Since CAI systems are often designed from the viewpoint of a teacher, too much emphasis has been placed on the teaching process rather than on the learning process. PHYSIM is a knowledge simulator for elementary physics using an object-oriented approach (Hou 1995). Its object knowledge base is partitioned into domains, each of which corresponds to a specific subject in high school physics. In addition to the object manager, the system has an event-driven simulation manager as well as a GUI manager. Instead of using pre-stored problem sets, the learner can experiment with an arbitrary scenario involving a variety of common physical objects from a set of primitives defined for the specific subject. Such a system can complement structured teaching in classrooms by providing solutions in the form of simulation to problems specified and manipulated by the learner.

Although physics is among the few "good" domains in which there exist well-defined laws governing objects, it is difficult or impossible to articulate the knowledge about any complex domain. As a result, a strictly knowledge-based system will perform poorly for domains involving many dynamic objects or inexact reasoning. Early cognitive experiments suggested that a human can often reason by remembering previous situations similar to the current one and uses them to help solve the new problem. Case-based reasoning is used to solve problems all around us. For example, attorneys are trained to use cases as precedents for constructing arguments in new cases. Medical doctors are trained by studying and analyzing cases of past patients for diagnosing new patients. Similarly, going over the solution steps for solving analogous physics problems can suggest plausible solutions to a new problem.

This paper presents the design and implementation of an intelligent teaching assistant (ITA) that facilitates a student to learn by actively exploring the subject domain in an interactive environment. The ITA starts out with relatively little knowledge about solving problems in the subject domain. Each successful problem solving episode is recorded, parameterized, and indexed into a case library. The primary objectives for PHYSIMC are to provide

- an environment for learning by active exploration,
- a user-friendly interface,
- guidance during the problem-solving process, and
- facilities for reviewing interesting cases.

The resulting system can help focus the student's attention on important features of a problem.

1.2 Case-Based Learning for ICAL

Case-based reasoning has been an area of active research in AI (Kolodner 1993; Leake 1996), and it has been considered as a promising technology for intelligent tutoring systems (Edelson 1996; Schank 1996).

0-7803-5214-9/98/$10.00 © 1998 IEEE.

Instead of using predefined cases for teaching, this research aims to integrate case-based learning with a knowledge simulation environment.

To supplement the incompleteness of domain knowledge, the system records each successful problem solving process performed by the teacher in a case library. When the student attempts to solve a new problem that doesn't have a readily applicable solution method, the system can retrieve and present to the student relevant problems from its case library. By browsing through similar problems, a student can often infer a plausible approach to work on the problem at hand. By working with the student step by step, performing necessary checking and derivation toward the solution, the system can assist the student in trying out different approaches to solving an unfamiliar problem. In addition, the system should provide suitable on-line help during the problem solving process on demand.

The case-based teaching assistant is designed to work in two exclusive modes:

1. *Teacher mode* — The teacher starts by specifying a problem scenario by selecting and dragging icons from the object menu into proper geometrical locations. She then defines the relationships among objects. Each object is then associated with a physical phenomenon and its relevant physical laws. The system performs the necessary substitutions of known values and calculations of unknown values. The system then proceeds to simulate the corresponding physical phenomenon exhibited by all the objects as a whole. Whenever there is a change of state, e.g. an object hitting the ground, the system pauses for the teacher to enter instructions for the next step. The process is repeated until all objects reach the termination state (e.g.\ when a system reaches equilibrium).
2. *Student mode* When the student works on a new problem without the guidance of a human teacher, the ITA can retrieve relevant cases according to a well-defined similarity metrics. The cases are presented to the student who will perform the necessary reasoning for adapting the case to the new situation. A more sophisticated system may propose plausible adaptation(s) of the old cases for the new problem. However, the adaptation problem is much harder and not explored here. The ITA module will continuously learn from new cases so that its performance should be improved over time.

In what follows, let us start by observing the steps involved in solving simple problems of projectile motion.

> *Suppose that a rigid ball is thrown at an angle θ above the horizon. Its initial velocity is v. Determine the distance between the point where the ball was thrown and the point where the ball touches the ground.*

The projectile motion is a two dimensional motion, which can be decomposed into a horizontal motion and a vertical one. The following laws of physics are applied to solve the problem.

$$\vec{v}_y = \vec{v}sin\theta + \frac{1}{2}\vec{a}t^2$$

$$t_1 = 2t$$

$$\vec{v}_x = \vec{v}cos\theta$$

$$\vec{s} = \vec{v}_x t_1$$

Given a new but *similar* problem:

> *Suppose that a rock is thrown at an angle θ above the horizon a time t_0. Its initial velocity is v. Determine the time t for the rock to touch the ground.*

Knowledge about the 2-D decomposition as well as the physics laws can be readily applied to solving the new problem. The key idea of PHYSIMC is to support such transfer of problem-solving experiences.

2 System Architecture

The overall architecture of the PHYSIMC system is shown in Figure 1. In this system, a number of distinct physics subjects share the knowledge base and case library. When the user designs physics problems for a specified subject, the system builds four managers for that subject automatically. The object manager also plays an important role as the intermediate interface to the knowledge base and the case library for the other modules.

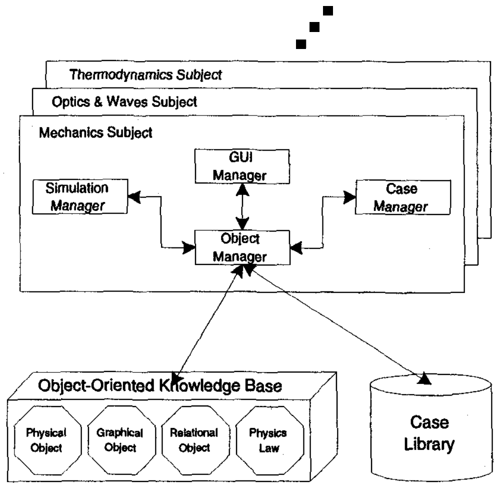


Figure 1: System Architecture of PHYSIMC

The major components include the following:

- The **GUI Manager** provides the interface between the system and the user.
- The **Object Manager** handles the creation and deletion of object instances.
- The **Simulation Manager** is in charge of the simulation scheme for the given subject.
- The **Case Manager** supports case based reasoning.
- The **Object-Oriented Knowledge Base** consists of objects for the specifications of physics problems and their simulation.

The system is designed to be used by both teachers and students. The user can describe an arbitrary physics problem in a given domain by defining the objects and their relationships in the GUI window. A text description of the problem can be recorded in the authoring window.

3 Case-Based Reasoning

Case-based reasoning is based on the idea that one can make use of past experiences in solving similar problems. There is much evidence that people do, in fact, use case-based reasoning in their daily activities. Adapting a memorized case to fit the new problem situation is an intuitive method for generating reasonable answers to the new problem. Physics problem solving requires knowledge about the domain as well as its solution strategy. When a new problem is presented, a student can often benefit from examining past experiences in solving similar problems. The case-based tools in PhySimC support the recording and browsing of prior problem solving episodes. The basic issues of case-based reasoning include case representation, case indexing, case retrieval and case adaptation. In this section, we propose to combine CBR with an *object-oriented knowledge model.* In particular, the method has been implemented to support case-based learning in PhySimC. The CBR process involves the following:

a. Use the problem description to identify a "similar" old case.

b. Retrieve the solutions to the old case and present them to the problem solver.

c. Adapt the results from the old case to the new situation.

d. Apply the adapted results to the new situation.

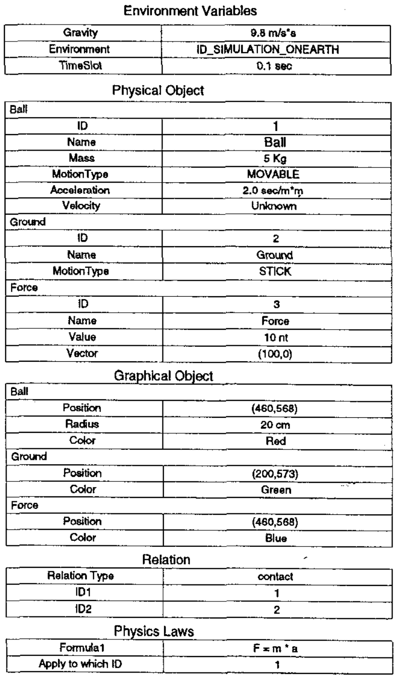
Due to its intuitive appeal, CBR has been applied to a wide range of problems such as robot navigation (Moorman & Ram 1992), manufacturing automation (Pu 1992), planning (Hammond 1989) and story understanding (Ram 1993; Burke & Kass 1996) etc. CBR is an effective method since it does not involve deep reasoning, which relies on a detailed theory of the domain.

3.1 Case Representation

Given that there are infinitely many problems with arbitrarily high complexities, it is impossible to index and store the solution for *every* physics problem in the case library. Fortunately, each problem can usually be decomposed into a finite number of subproblems. Therefore, the case library should consist of primitive problems and their solutions. A student can learn to solve a complex problem by combining the solutions of the constituent problems.

The PhySimC system employs an object-oriented knowledge base to simulate a variety of phenomena in simple physics domain. Each case contains a collection of objects corresponding to four types of knowledge resources (physics objects, graphical objects, physics laws, relation objects) in the knowledge base. Each case in PhySimC is organized into a flat sequential structure, which provides an effective method for case representation in an object-oriented environment. The case representation for the simple physics problem below is shown in Table 2.

> *A small ball with a mass of 5Kg is placed on the ground. Given a 10NT force in the horizontal direction to the ball, determine the acceleration of this ball.*



Environment Variables

Gravity	9.8 m/s*s
Environment	ID_SIMULATION_ONEARTH
TimeSlot	0.1 sec

Physical Object

Ball	
ID	1
Name	Ball
Mass	5 Kg
MotionType	MOVABLE
Acceleration	2.0 sec/m*m
Velocity	Unknown
Ground	
ID	2
Name	Ground
MotionType	STICK
Force	
ID	3
Name	Force
Value	10 nt
Vector	(100,0)

Graphical Object

Ball	
Position	(460,568)
Radius	20 cm
Color	Red
Ground	
Position	(200,573)
Color	Green
Force	
Position	(460,568)
Color	Blue

Relation

Relation Type	contact
ID1	1
ID2	2

Physics Laws

Formula1	F = m * a
Apply to which ID	1

Figure 2: A sample case as a list of objects

Several alternative structures, such as memory organized packets hierarchy (Schank 1982). conceptual graphs (Fung & Adam 1995), and explanation patterns (Schank 1986) have been used in CBR systems. Although these sophisticated structures can facilitate reasoning in case adaptation, adding additional information into these structures is a complex task.

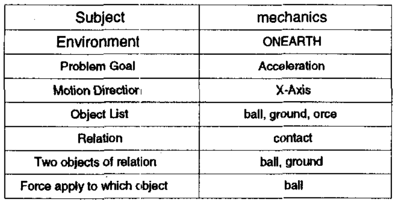
In addition to the problem description, a case representation should also contain the following information.

- **Solution:** the stated or derived solution to the problem;
- **Outcome:** the resulting world state when the solution is carried out.

Case specifications in CBR may be incomplete.

3.2 Case Indexing Scheme

Indices of cases in a case library have a significant impact on the efficiency of a case-based system. In this work, the indices for a case is selected by the system builder, who is responsible for analyzing a domain in order to create a good index list. Since there are different subject domains, the indices are selected according to characteristics of the domains. For example, the indices for the physics example earlier in this section are shown in Figure 3. The PhySimC system provides a tool for index selection. A list of indices is separated from the corresponding case in the implementation. The system does not have to load the entire case into memory when selecting similar cases. As a result, memory space is minimized and search speed is enhanced.



Subject	mechanics
Environment	ONEARTH
Problem Goal	Acceleration
Motion Directior	X-Axis
Object List	ball, ground, orce
Relation	contact
Two objects of relation	ball, ground
Force apply to which object	ball

Figure 3: Sample Indices

3.3 Case Retrieval

The most important issue in PhySimC, and indeed in any case-based system, is to select the exact or partial matches from the case library. A good CBR system should have a retrieval algorithm that is both *correct* and *efficient* for handling a large number of cases. Efficient retrieval algorithms take into account priority among features of a case. For example, if the reasoner is attempting to choose a plan to achieve a goal, then a match between the current *goal* and a *goal* in an old case is the most important; the next is to match the *constraints* that guide how the goal must be achieved; and then is to match the *features* selected by the old case. Matches on other features are ranked considerably lower in importance.

3.4 Retrieval Algorithm

The current case library contains only the fundamental problems in kinematics and optics. A flat memory model is adequate when the case library is of a relatively size. Cases are located by a sequential search of the memory for matches. However, when PhySimC is extended to include more cases, it is necessary to partition the case library. One possibility is to partition the case library according to the physics subjects. The retrieval algorithm will keep cases from being retrieved unless the subject is matched. For example, in identifying matches for an input problem on kinematics, it is not necessary to spend time in searching through cases on optics. Search through the entire case library can easily be avoided.

Unlike database query processing, one may wish to retrieve cases that are only partial matches. It is possible that there doesn't exist any case in the case library that matches the new situation exactly, so search must result in retrieval of a close partial match. Partial match algorithms are usually quite expensive (Kolodner 1993). Because of the cost, the retrieval algorithm must be directed in some way so that cases with high potential relevance to the new situation can be selected. Therefore, the methods of retrieval are often directed to partition the search space on only a small number of cases.

To choose the best, or most useful cases, a *distance metric* is used in the partial-matching process. Once PhySimC has decided to look for a new case, it uses the *distance metric*, which pre-defines the distance between any two different values of the same feature, to find the case that best matches the current situation.

Algorithm for case retrieval

```
Load all subject domains $S = \{S_1, S_2, \ldots, S_m\}$
Fetch the subject domain $C$ of the query case
FOR (each element $S_i \in S$) DO
    IF ($C == S_i$) THEN
        Load index sets $\{I_1, I_2, \ldots, I_n\}$
IF (no subject domain $C$ in the case library) THEN
    build subject domain in case library
    exit
Fetch query case index set $Q = \{Q_1, Q_2, \ldots, Q_k\}$
Load index set $X = \{X_1, X_2, \ldots, X_k\}$ of $D_i$
distance = $\sum_1^k$ distance($Q_i, X_i$)
Select case with the smallest distance.
```

In order to help the student understand characteristics of the physics problem he tries to solve, the PhySimC system provides a tool to select the priority of features about the problem. Retrieval algorithm works according to the priority of features. If solutions of the previous case are not fitted to the current situation, the student can change the priority of features to search for the most familiar case.

3.5 Case Adaptation

Case adaptation is an important issue for case-based reasoning. In general, previous solutions are rarely exact matches for a new problem. Old solutions usually need to be adapted in order to fit the new situations. In the physics domains, there are too many different features in every subject to define a good adaptation rule to control the process of case adaptation. The simulation environment facilitates learning by doing, which is more effective than being told the result of automation adaptation. By providing similar cases in PHYSIMC, the student can adapt the solutions from previous cases to the current situation. Working out the solutions step by step will enhance the student's understanding of the fundamental concepts about the physics problem.

3.6 Serialization

The PHYSIMC system provides a laboratory environment for the teacher and student to experiment with physics problems. The user can specify a physics problem using graphical representation and assign physics laws to this problem. At the end of a problem solving episode, there are many active objects in the memory. The key question is how to store these live objects and be able to load them back from the hard disk. In order to allow objects to persist in the hard disk, the serialization mechanism provided in the Microsoft Foundation Class Library (MFC) is used in PHYSIMC.

The basic idea of serialization is that an object should be able to write its current state, usually indicated by the value of its member variables to some persistent storage. Later, the object can be re-created by reading, or de-serializing, the object's state from the storage. When an object is serialized, the object itself is responsible for reading and writing its own state. Thus, for a class to be serializable, it must implement the basic serialization operations.

4 Using PHYSIMC

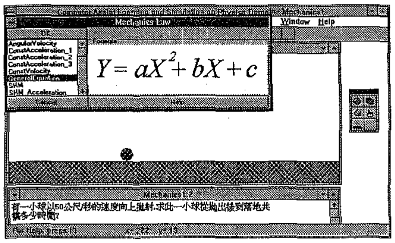
When the PHYSIMC system is started, many tools for designing and entering a new case into the case library are available. This section goes through working with the following physics problem to demonstrate using these tools.

A small ball is placed on the ground. Given a initial velocity 50 m/sec in the vertical direction to the ball. Determine the time it takes for the ball to fall down and reach the ground.

4.1 Create a New Case

Specify the Physics Problem The teacher chooses menu command *Kinematics* under the *Subject* menu. The system will pop up a simulation window and the corresponding object palette. Physical objects can be placed on arbitrary positions within the simulation window by clicking the corresponding icons in the palette and dragging them to the proper positions. A teacher can also describe the statement of physics problem in text mode. He can open an authoring window under the *Window* menu and directly writes the statement describing the problem.

In order to solve a physics problem, the teacher needs to identify a subset of physics laws which are relevant to the solution of this problem. The system provides a default set of equations for each subject. In Figure 4, the teacher has selected the command *Kinematics* under the *Law* menu. Two equations, for constant acceleration motion and the general equation, are then assigned to the ball.



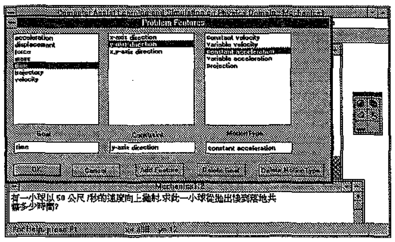
$Y = aX^2 + bX + c$

有一小球以50公尺/秒的速度向上拋射.求此一小球從拋出後到落地共需多少時間?

Figure 4: Selecting a relevant General Equation from the mechanics domain

Enter the Case into the Case Library If the teacher wants to enter the instructional materials described above into the case library, he must define the features of this problem. These features play an important role that make the process of case retrieval effective. In Figure 5, the teacher chooses the command *Problem features* under the *Case* menu, and the system creates a "Features" dialog. The teacher selects features for the physics problem through the dialog box. In addition to the features provided by the system, the teacher can also add new features for the specific problems.

After the teacher makes sure that all work has been completed correctly, he chooses the command *Save case* under the *Case* menu. The features and representation of the case are then stored in the case library automatically.



Goal: time
Constraint: y-axis direction
Motion Type: constant acceleration

有一小球以50公尺/秒的速度向上拋射.求此一小球從拋出後到落地共需多少時間?

Figure 5: Selecting features for a case

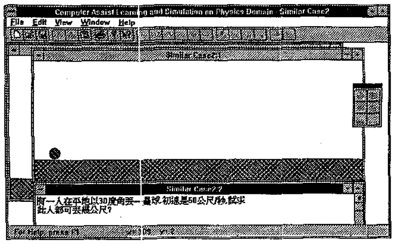
Review and Update Cases in the Case Library Supposed that the system has collected a large amount of instructional materials, which may contain errors, in the case library from many teachers. It is necessary to have some functionality for reviewing and updating cases by the teachers. After the teacher selects the command *case* of some subject under the *Subject* menu, the system will create a window to show the first case in the case library. The teacher can review the other cases by clicking the *Next* and *Back* buttons. If the teacher has modified certain cases, choosing the command *Update case* under the *Case* menu will update its content. There is also the functionality of deleting cases when the teacher decides to discard bad or useless cases.

4.2 Select a Similar Case

Suppose that PHYSIMC has collected instructional materials from many teachers which can be arranged into a complete course. A student can utilize the support of similar cases in problem solving using the case library built by the teachers. When the student tries to solve a new physics problem, he can specify the problem and assign the necessary features about the new problem.

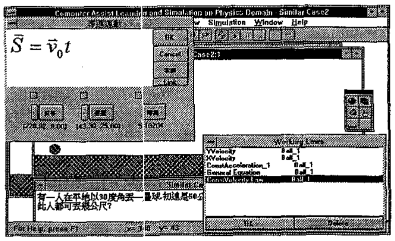
There are two ways to retrieve similar cases in the PHYSIMC system. One is to select cases automatically based on the default priority among features. Alternatively, the student can select similar cases according to his own priority of features.

In Figure 6, the system has retrieved four similar cases according to the priority of the features and presented the most similar case in the simulation window. Then the student can open the physics law dialogs to derive solutions to fit the current problem from the similar case. In Figure 7, the dialogs show the process of assigning physics laws to solving this problem. Finally, the student can view the result in simulation.


Computer Assist Learning and Simulation on Physics Domain - Similar Case2
File
Edit
View
Window
Help
Similar Case2:1
Similar Case2:2
For Help, press F1

有一人在平地以30度角丟一壘球,初速是50公尺/秒,試求
此人都可丟幾公尺?

Figure 6: Showing the most similar case in the simulation window



$\bar{S} = \vec{v}_0 t$

YVelocity Ball_1
XVelocity Ball_1
ConstAcceleration_1 Ball_1
General Equation Ball_1
ConstVelocity law Ball_1

有一人在平地以30度角丟一壘球,初速是50公
此人都可丟幾公尺?

Figure 7: Computing the displacement

5 Conclusion

The paper has presented a case-based learning tool in PHYSIMC, a computer-assisted simulation environment. The case-based component was implemented as an extension to PHYSIM (Hou 1995). In PHYSIMC, a teacher can construct arbitrary problems and solutions to be cached into the case library, and a student can obtain hints for solving a new problem by retrieving similar problems from the case library. Preliminary experience with the system shows that CBR can facilitate learning of problem-solving knowledge.

References

Burke, R., and Kass, A. 1996. Retrieving stories for case-based teaching. In Leake, D. B., ed., *Case-Based Reasoning: Experiences, Lessons, & Future Directions.* AAAI Press/MIT Press. chapter 5, 93–109.

Edelson, D. 1996. Learning from stories:indexing and reminding in a socratic case-based teaching system for elementary school biology. Technical report, Institution of Learning Science in Northwestern University.

Fung, P.-W., and Adam, A. 1995. Conceptual graphs for similarity mearsure in a case-based physics problem solver. In *Artificial Intelligence in Education*, 170–177.

Hammond, K. 1989. *Case-Based Planning: Viewing Planning as A Memory Task.* San Diego, CA: Academic Press, Inc.

Hou, C.-C. 1995. Object-oriented modeling and simulation for ical. Ms thesis, Department of Computer Science and Information Engineering, National Taiwan University.

Kolodner, J. 1993. *Case-Based Reasoning.* Morgan Kaufmann Publisher, Inc.

Leake, D. B. 1996. Cbr in context: The present and future. In Leake, D. B., ed., *Case-Based Reasoning: Experiences, Lessons, & Future Directions.* AAAI Press/MIT Press. chapter 1, 3–30.

Moorman, K., and Ram, A. 1992. A case-based approach to reactive control for autonomous robots. In *AI for Real-World Autonomous Mobile Robots.*

Pu, P. 1992. An assembly sequence generation algorithm using case-based search techniques. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 2425–2430.

Ram, A. 1993. Indexing, elaboration and refinement: Incremental learning of explanatory cases. *Machine Learning* 10(3):201–248.

Schank, R. 1982. *Dynamic Memory.* the Press Syndicate of the University of Cambridge. chapter 6, 95–109.

Schank, R. 1986. *Explanation Patterns: Understanding Mechanically and Creatively.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C. 1996. Goal-based scenarios: Case-based reasoning meets learning by doing. In Leake, D. B., ed., *Case-Based Reasoning: Experiences, Lessons, & Future Directions.* AAAI Press/MIT Press. chapter 15, 295–347.