A Personalized Emotional Intelligent Tutoring System

Based on AI Planning

Dissertation zur Erlangung des Doktorgrades
Doktor der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Ingenieurwissenschaften und Informatik
der Universität Ulm

vorgelegt von

Heba Mohamed Atef Elbeh
aus Menoufia, Ägypten
Institut für Künstliche Intelligenz
Fakultät für Ingenieurwissenschaften und Informatik
Universität Ulm

Institutsleitung
Prof. Dr. Susanne Biundo-Stephan

Ulm, Deutschland
June 2012
Amtierender Dekan der Fakultät für Ingenieurwissenschaften und Informatik:
Prof. Dr.-Ing. Klaus Dietmayer

Vorsitzender des Promotionsausschusses:
Prof. Dr. Uwe Schöning

Mitglieder des Promotionsausschusses:
Prof. Dr. Michael Weber
Prof. Dr. Wolfgang Minker

Die Gutachter der Dissertation:
Prof. Dr. Susanne Biundo-Stephan
Prof. Dr. Harald Traue

Tag der Promotion: 27 September 2012
A Personalized Emotional Intelligent Tutoring System

Based on AI Planning

A PhD thesis submitted to the Faculty of Engineering and Computer Science, Ulm University in fulfillment of the requirements for the degree of Doktor rerum naturarum (Dr. rer. nat.)

by

Heba Mohamed Atef Elbeh from Menoufyia, Egypt Institute of Artificial Intelligence Faculty of Engineering and Computer Science Ulm University

Advisor

Prof. Dr. Susanne Biundo-Stephan

Ulm, Germany June 2012
Dean of the Faculty of Engineering and Computer Science:  
Prof. Dr.-Ing. Klaus Dietmayer

Chairman of the doctoral committee:  
Prof. Dr. Uwe Schöning

Members of the doctoral committee:  
Prof. Dr. Michael Weber  
Prof. Dr. Wolfgang Minker

Reviewers of the dissertation:  
Prof. Dr. Susanne Biundo-Stephan  
Prof. Dr. Harald Traue

Day of Conferral of Doctorate: 27 September 2012
ABSTRACT

The Intelligent Tutoring System (ITS) is a computer based learning system which assists students in their learning process. It has the ability to be adaptable according to the needs of students. In ITS, it is equally important to consider not only the cognitive level of the student but also the emotional state of the student. Psychological researches indicate that emotions have a deep influence on the efficiency of memory storing and retrieving processes. Also, it is important to individualize the learning process and to teach students according to their personality types. Thus, students with various personality types differ in their cognitive capabilities, emotional and motivational states, learning style, appraisal, and coping way with incoming events.

Although several approaches have been constructed for ITS, enhancing students emotional intelligence has not been considered so far. Emotional intelligence (EI) is considered as an important factor for increasing the student’s performance. The researches show that IQ (intelligence Quotient) contributes only about 20% to success in life, the rest of 80% of success depends on the EQ (Emotional Quotient). Thus, in our educational systems, we are not taught how to handle frustration, anxieties, stress, and failure or depression problems during the learning process. EI is defined as the ability to recognize the meanings of emotion and their relationships, to reason about emotions, to enhance thought.

With the consideration of previous factors, we present a novel framework for modeling an independent authoring/course generation system named PANDA.TUTOR. This system allows the author to prepare the course structure and content enriched with classification information without need to define the adaptation rules or specify configurations for each student, aiming to keep all the authoring effort low in terms of time and effort. The course generation system, which is basically a hierarchical planning system, generates a personal course for the student. The system enriches with different learning scenarios, teaching strategies and learning styles. The student’s personality type in our system serves as a guide for selecting an appropriate scenario, a teaching strategy, and a learning style in order to regulate with current emotional and motivational states of student. For that purpose, we developed general ontologies for representing student module and course module that can be used in different domains. These ontologies help the planning system to generate a personalized course. In addition, in order to enhance the student’s emotional intelligence, we developed a dialog based HTN planning system. It is shown to improve the emotional intelligence of the student. This can be done by grasping the student’s appraisal to build up an individual’s interpretation of how external events relate to the student goal and desires, as well as overcoming their emotional reasoning. Emotional reasoning is defined as a thinking error that occurs when a student believes that his/her feeling is true regardless of the evidence, which prevents one looking at alternative, more balanced information or evidence. Different styles of dialog can be generated according to the student’s personality type. Moreover, we regard the student’s emotional appraisal and coping to regulate his/her emotion during the dialog and course generation. Thus, the objective of our approach is to teach the student during the learning process how to adjust or maintain the emotion, motivation, and cognitive states. Therefore, the proposed system is able to understand the student’s emotion and the reason behind this emotion to shape the students’ behavior.
First of all, I would like to express my special appreciation and thanks to my advisor Prof. Dr. Susanne Biundo-Stephan, you have been a tremendous mentor for me. I would like to thank you for your continuous support and for all your invaluable guidance that makes this period of study enjoyable. Regardless of how much and what she had to do, she was always available for discussion and support, and a never dwindling source of ideas and suggestions.

Beside my supervisor, I would like to thank the rest of my defense committee for their guidance. Special thanks to Dr. Bernd Schattenberg for his valuable support.

I am grateful for the fruitful discussions with our friends in the Artificial Intelligence department, whose blessings and wishes have helped me come this far.

Words cannot express the profoundness of my gratitude to my husband Dr. Mohamed Elkawkagy, my son Yousuf and my daughter Salma. Thank you for your support and patience during the whole time of my thesis. Without them, I wouldn’t be standing here.

My sincere thanks go to my parents, sisters, for supporting me in everything, and especially I can’t thank you enough for encouraging me throughout this experience.

Finally I thank my God, for letting me through all the difficulties. I have experienced Your guidance day by day. You are the one who let me finish my degree. Thank you, Allah.

Ulm, Germany, 2012

Heba Elbeh
DEDICATION

To my parents, my husband and my kids (Yousuf and Salma).
# Contents

Table of Contents xi

List of Figures xv

List of Tables xvii

1 Introduction 1
   1.1 Motivation and Outline ........................................ 1
   1.2 Structure of The Dissertation ................................. 5

2 Background 9
   2.1 Intelligent Tutoring Systems .................................... 9
   2.2 Intelligent Tutoring System Architecture ..................... 13
      2.2.1 Course Module ............................................. 15
      2.2.2 Student Module ........................................... 19
      2.2.3 Pedagogical Module ....................................... 23
      2.2.4 Course Generation ......................................... 25
   2.3 Learning Theories .............................................. 30
      2.3.1 Interface Module ......................................... 34
   2.4 Authoring Systems .............................................. 34
      2.4.1 Authoring Tools for Intelligent Tutoring Systems ...... 36

3 Planning 39
   3.1 AI Planning ....................................................... 40
      3.1.1 Classical Planning ......................................... 40
      3.1.2 Partial Order Plan ......................................... 43
      3.1.3 Hierarchical Planning ..................................... 46
      3.1.4 Hybrid planning ........................................... 52
   3.2 Formal Framework .............................................. 54
      3.2.1 Logical language ........................................... 55
      3.2.2 Tasks ....................................................... 56
      3.2.3 Domain Model .............................................. 57
3.2.4 Plans .................................................. 58
3.2.5 Planning Problems and Solutions ...................... 59
3.2.6 Plan Generation ....................................... 61
  Flaws .................................................... 62
  Plan Modification ....................................... 62
3.2.7 Search Strategy ....................................... 63
  Refinement Algorithm .................................... 64
4  System Architecture ........................................ 67
  4.1 Affective Computing .................................... 67
  4.2 Emotional Intelligence .................................. 70
  4.3 System Architecture .................................... 73
    4.3.1 The Goal of the Proposed Architecture ............. 75
    4.3.2 Course Module ...................................... 76
    4.3.3 Student Module ..................................... 77
    4.3.4 Pedagogical Module ................................ 79
    4.3.5 Task-Centered Instruction Strategy ................. 80
    4.3.6 PANDA.TUTOR Scenarios ............................ 83
    4.3.7 Generated Plan ...................................... 87
    4.3.8 Student Interface ................................... 88
    4.3.9 Author interface .................................... 88
5  Course Module ............................................... 89
  5.1 Ontology and Semantic Web .............................. 89
    5.1.1 Ontology and E-Learning .......................... 94
  5.2 Concept Map Based on Adaptive Educational System .... 97
  5.3 Learning Objects ....................................... 99
    5.3.1 Knowledge Objects and Instruction Strategies .... 102
  5.4 Course Module Ontology In PANDA.TUTOR ............... 104
  5.5 Authoring Part ......................................... 114
6  Student Module ............................................... 117
  6.1 The Human Brain ....................................... 118
  6.2 The Cognitive Process .................................. 121
  6.3 Personality Mood and Emotion .......................... 122
  6.4 Student Personality .................................... 123
    6.4.1 Types of personality .............................. 124
  6.5 Student Mood ........................................... 127
  6.6 Student Emotion ....................................... 128
    6.6.1 Emotion and Personality ........................... 131
  6.7 Motivation in Educational Environment .................. 132
    6.7.1 Personality and motivation ......................... 133
## CONTENTS

6.8 Learning Styles ............................................................ 134  
  6.8.1 Personality and Learning Styles ................................. 135  
6.9 Student Control ....................................................... 135  
6.10 Student Module Ontology in PANDA.TUTOR ..................... 136  
  6.10.1 Static Information .............................................. 137  
  6.10.2 Dynamic Information ...................................... 137  

7 Emotional Intelligence and Apprising Process .................... 141  
  7.1 Why we focus on Emotional Intelligence in PANDA TUTOR?  ......................................................................... 142  
  7.1.1 The Emotional Intelligence Branches ....................... 144  
  7.2 How to Improve the Emotional Intelligence .................. 146  
  7.2.1 Emotion Election ................................................ 149  
  7.3 Emotional Reasoning ................................................ 150  
  7.4 Appraising and Coping Theory ................................... 151  
  7.4.1 Appraising Process ............................................. 151  
  .............................................. 153  
  .............................................. 155  
  .............................................. 157  
  7.4.2 Coping process .................................................. 159  
  7.5 Personality and Coping Process ................................... 160  
  7.6 Emotion Regulation .................................................. 164  
  7.7 Emotional Regulation Strategies ................................. 166  
  7.8 Appraisal Coping and Regulation in PANDA.TUTOR ........... 167  
  7.9 Emotional Ontology .................................................. 169  

8 Affective Planning .......................................................... 171  
  8.1 PANDA.TUTOR .......................................................... 171  
  8.2 Pedagogical Module .................................................. 174  
  8.3 Emotional Intelligence and Appraisal process in PANDA.TUTOR ......................................................... 175  
  8.3.1 Appraisal variables ............................................. 178  
  8.4 Dialog Planning in PANDA.TUTOR ............................... 180  
  8.5 Affective Planning for Lesson Generation ...................... 191  
  8.5.1 Regulation Process in PANDA.TUTOR ....................... 194  
  8.5.2 Affective Planning Problem ................................... 196  
  8.5.3 Affective Planning Domain ................................... 196  
  8.5.4 Sorts and Relations ............................................. 199  
  8.5.5 Tasks and Methods ............................................. 202  

9 Conclusion ....................................................................... 215  
  9.1 Research Contributions ............................................ 215  
  9.2 Future Work ............................................................ 217
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Dependencies among the chapters.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Basic architecture of an intelligent tutoring system</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Adaptive hypermedia application model</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>Overlay student models</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>Bloom taxonomy</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>State space search versus plan space search</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Planning by action abstraction.</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Implementations of tasks (primitive and abstract)</td>
<td>57</td>
</tr>
<tr>
<td>4.1</td>
<td>The basics of emotional intelligence</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>The system architecture</td>
<td>74</td>
</tr>
<tr>
<td>4.3</td>
<td>Task-Centered instruction strategy</td>
<td>81</td>
</tr>
<tr>
<td>4.4</td>
<td>First principles of instruction Merril.</td>
<td>82</td>
</tr>
<tr>
<td>4.5</td>
<td>A conceptual learning model</td>
<td>83</td>
</tr>
<tr>
<td>5.1</td>
<td>Semantic web stack</td>
<td>90</td>
</tr>
<tr>
<td>5.2</td>
<td>Mapping from OWL ontology elements to concept map elements</td>
<td>98</td>
</tr>
<tr>
<td>5.3</td>
<td>Layers of course structure</td>
<td>106</td>
</tr>
<tr>
<td>5.4</td>
<td>Artificial intelligence course</td>
<td>107</td>
</tr>
<tr>
<td>5.5</td>
<td>Knowledge object of planning concept in artificial intelligence course</td>
<td>109</td>
</tr>
<tr>
<td>5.6</td>
<td>Instruction component strategies of planning concept</td>
<td>110</td>
</tr>
<tr>
<td>6.1</td>
<td>The layers of human Brain</td>
<td>118</td>
</tr>
<tr>
<td>6.2</td>
<td>Relations between personality mood and emotion</td>
<td>123</td>
</tr>
<tr>
<td>7.1</td>
<td>The four branches of emotional intelligence</td>
<td>145</td>
</tr>
<tr>
<td>7.2</td>
<td>Emotional intelligence wheel</td>
<td>147</td>
</tr>
<tr>
<td>7.3</td>
<td>Conceptualization of emotional intelligence</td>
<td>149</td>
</tr>
<tr>
<td>7.4</td>
<td>The cognitive motivational emotive system</td>
<td>152</td>
</tr>
<tr>
<td>7.5</td>
<td>The original structure of emotions of the OCC model</td>
<td>155</td>
</tr>
<tr>
<td>7.6</td>
<td>Attribution theory</td>
<td>156</td>
</tr>
</tbody>
</table>
7.7 Emotion regulation model by Gross .............................................. 165
7.8 Appraisal and coping theory in our approach ................................. 168
7.9 Emotional ontology ................................................................. 170

8.1 Phases of pedagogical module ..................................................... 174
8.2 Reason of emotion ................................................................... 178
8.3 Dialog based HTN-planning in PANDA.TUTOR ............................ 181
8.4 Diagnoses emotion state in PANDA.TUTOR: Example-1 ................. 188
8.5 Diagnoses emotion state in PANDA.TUTOR: Example-2 ................. 189
8.6 A fragment of the PANDA.TUTOR sort hierarchy .......................... 209
8.7 The decomposition hierarchy for the complex task schema PANDA − TUTOR − Course ................................................................. 210
8.8 The decomposition hierarchy for the complex task schema Generate_Lesson_Scenario and its definition ................................................. 211
8.9 The decomposition hierarchy for the learning process ..................... 212
8.10 The decomposition hierarchy for the learning process ..................... 213
List of Tables

2.1 Dexter hypertext reference model ............................................. 15
3.1 Different kinds of flaw classes and the corresponding modification classes ........... 64
5.1 Instructional strategies for component skills (Merrill) ........................................... 103
7.1 The emotion type specifications of the OCC model. ............................................ 154
7.2 The emotion type specifications of the OCC model. ............................................ 166
8.1 Some regulation strategies. ........................................................................ 197
8.2 Sort elements of the logical language $\mathbb{L}$. .............................................. 200
Chapter 1

Introduction

1.1 Motivation and Outline

Intelligent Tutoring System ITS is a computer based learning technique which assists students in their learning process. The Course Generation System (CGS) or Course sequence is a well established technology in the field of intelligent tutoring systems. The idea of course generation is to generate an individualized course for each student by selecting the most optimal teach operation. It has the ability to be adaptable according to the needs of students. Therefore, students will be able to achieve the learning goal more efficiently when the pedagogical procedure is adapted to their individual differences. Thus, students with various personality types differ in their behavior. In addition, it is equally important to consider not only the cognitive level of the student but also the emotional state of the student. Psychological researches indicate that emotions have a deep influence on the efficiency of the memory storing and retrieving process. However, our educational system concentrates on improving IQ (intelligence Quotient) of EQ (Emotional Quotient). The researches show that IQ contributes only about 20% to success in life, while the remaining 80% of success depends on our EQ [1]. In other words, in our school, we are not taught how to handle
frustration anxieties, stress, failure or depression problems during the learning process. In addition, we are not told to learn how to manage emotions (i.e. interaction, coordination, adjustment, communication). At the later stages of our lives we are told to master emotional competencies to be successful.

So, it is important to enhance the emotional intelligence of the student as well as intellectual intelligence. Emotional intelligence (EI) refers to an ability to recognize the meanings of emotion with their relationships, and the capacity to reason about emotions to enhance thinking; to reflectively regulate emotions, and promote emotional and intellectual growth.

However, the process of arranging personalized adaptations is usually complex, the current platforms do not usually provide more than a relatively simple way for personalization and adaptation. Although most course generation systems generate a suitable course for the student’s state considering the student's cognitive ability, some considerations are given to the motivational and emotional state of student [2] [3], the student’s personality type, emotional appraisal and coping ways and different regulation strategies for each personality type are not considered. [4, 5].

In this thesis we will introduce PANDA.TUTOR, a model for a new approach for course generation in ITS, based on hybrid planning. The considered planning system is a combination between HTN planning and partial order causal link [6]. HTN planning is based on the concepts of tasks and methods [7]. Tasks are either abstract and are implemented by different methods, or primitive, and correspond to operators in classical planning. Furthermore, a planning problem is an initial partial plan which is solved by incrementally decomposing abstract tasks until a plan is generated consists only of primitive tasks. The hierarchal capability of HTN planning enhances the modeling by considering different classification criteria(partial order causal link).

In PANDA.TUTOR a new approach for the dialog based HTN planning is presented in order to improve the emotional intelligence of the student and to grasp the student's appraisal. This dialog can help the planner to build up an individual’s interpretation of how external events relate to the
student’s goal and desires, as well as overcoming their emotional reasoning. Emotional reasoning is defined as a thinking error that occurs when a student believes that his/her feeling is true regardless of any evidence. This prevents the student to look at alternative, more balanced information or evidence.

In addition, the student’s personality type is considered as a main factor to generate an appropriate dialog and course for the student. Moreover, we regard the student’s emotional appraisal and coping way to regulate his/her emotion during the dialog and course generation.

The dialog produces a set of predicates as diagnostic facts about the student. The course generation uses these facts in its planning problem to generate an appropriate course for the student. These predicates contain the emotional and motivational states of the student.

The system uses different emotional theories for the appraisal process, such as cognitive model of emotion OCC [8], an attribution theory [9], and a control-value theory [10]. In addition, in PANDA.TUTOR a new independent Authoring/Course Generation approach is presented. The course content consists of a textual description, examples, exercises etc., and is prepared by the author, who enriches it with classification information. The course generation system which is a hierarchical planning system generates the course by tailoring the learning content to an individual learner, considering the student’s goal, the current emotional and motivational state of the student.

The authoring part in our approach is defined by the course module and the course generation part is defined by the pedagogical module. The authoring phase can help a non-programmer teacher to configure different aspects of the course domain model. Thus, we aim to keep the needed amount of time and effort for the authoring process low. By using our proposed architecture, the author can modify the course structure and incorporate new learning materials without the need of intervention from the system developers. Moreover, the authors in this approach do not need to define a set of adaptation rules or specify a specific teaching strategy for each student type. Also, there is no need to create or modify the teaching method or strategies for each student. In other words, in
contrast to other systems in which the author uses the authoring tool to define the Course Module, Student Module, and rules of pedagogical Module. The author in our approach should only define the course structure.

On the other side, the course generation system which is an HTN planning system selects the appropriate scenario for the student and generates an appropriate course. Thus, the course generation system aims to tailor the instruction on a micro level by diagnosing the student’s specific learning needs and providing instructional strategies for those needs. The pedagogical module generates the course regarding the course module and student module. The planning system generates a course without an intervention from the author in determining an appropriate strategy for each student. PANDA.TUTOR includes different learning scenarios, strategies, learning styles, and regulation strategies which help the planner to construct highly individualized courses, and help students to build up their knowledge.

So, we modeled ontologies for the course module and student module. The development of these ontologies is a step toward creation shared reusable and adaptive educational systems. These ontologies will help the planning system to generate a personalized course for a student and provide adaptive learning environments. The ontology also permits the retrieval of learning materials after the planner has generated the course. The proposed course ontology model is a general educational ontology based on knowledge objects and the Instruction Design theory, as introduced by Merrill [11]. Furthermore, we have considered a separation between the course content reusability and the learning objects reusability. This ontology can be used by the author for constructing the course. Afterwords, the authorized course is considered during the pedagogical module to construct a personalized lesson for the student.

Accordingly, a personalized course can be delivered to a student as a structured representation of the subject, as well as helping the student to adjust or maintain his/her emotion.

There is no doubt that the emergence of web technology gave rise to a number of web based ITSs,
as a type of Web-Based Intelligent Tutoring Systems. Therefore, we considered the new generation of educational systems which combine the semantic web technique with educational technique in our course generation system.

1.2 Structure of The Dissertation

This dissertation is broken down into eight chapters (see Figure 1.1 which describes the dependency between these chapters):

- **Background**
  Through this chapter we will introduce an overview about ITS with the typical architecture of ITSs as well as the related work and techniques to our work.

- **Formal Framework**
  This chapter presents the underlying formal framework that covers our hierarchical planner. First, the syntax and semantics of the planning data structures will be explained, as well as the formalization of the planning problem. Second, we will present an outline of the hybrid planning system, considering HTN planning and partial order planning.
Chapter 1 Introduction

• **System Architecture**

  In this chapter the architecture of our system PANDA.TUTOR will be introduced with a brief description of the architecture parts.

• **Course Module**

  The purpose of this chapter is to describe the course module. Thus, we describe our course module considering the ontology technique and semantic web, which provides an adaptive learning environment and facilitates sharing and reusing learning materials. The ontology model is based on the Task-Centered instruction strategy and the knowledge object approach. In addition, we have considered realizing separately re-usability of the course content and the re-usability of learning objects. The authoring part is defined in this module.

• **Student Module**

  The student personality type plays an authoritative role in our system, we connect all the educational, motivational and emotional feature to the personality type. This information is used by the pedagogical module to generate a personalized course for a student. Furthermore, we discuss research into the connection between the personality type and the strong of emotion, achieving motivation, and learning styles. We explain different aspects that play a role in this module and describe the student module ontology.

• **Emotional Intelligence and apprising process**

  The aim of our system is to enhance the emotional intelligence of a student as well as the intellectual intelligence. In this chapter we will explain how we can we improve the student’s emotional intelligence. Likewise we will explain how we introduced the appraising, coping and regulating process in our system regrading different personality types.

• **Pedagogical module**

  The pedagogical module is a predominate module in PANDA.TUTOR. We used hybrid plan-
ning for modeling in this module regarding two phases dialog based planning phase and course generation phase.

- Conclusion

Finally, we look into the possibility to extend our approach in the future development.
Chapter 2

Background

Most of empirical studies investigate the importance of considering Intelligent Tutoring Systems (ITS) as an affective tool for the learning process. ITSs are educational systems based on computer technology, for the purpose of improving the learning process. They provide a scaffolding for constructing a teaching environment geared towards helping a student practice skills. There are different methods and techniques by which artificial intelligence can be used to improve the performance of educational systems. In this chapter we would like to give an overview over ITS architecture and the techniques that play an important role in the field of education.

2.1 Intelligent Tutoring Systems

Using the computer as an aid in the educational process has been started three decades ago. Numerous computer-based education systems have been developed during these decades.

The 1950s saw the first ITS in the form of Computer Aided Instruction (CAI) systems [12] with simple 'linear programs'. These systems were quite effective in helping students. In these systems the selected material is arranged and presented in a series of 'frames' to take the student step by step toward the desired behavior. Most frames used very simple questions (e.g. involving only
the filling in of a missing spaces), and the student was told immediately whether the answer was right or wrong. The student’s answer was compared to the correct answer, then the computer give an appropriate feedback. If the answer was correct, a new problem was selected and presented, but if the student answered incorrectly, a simple kind of remediation was delivered to the student. In addition, CAIs could not understand or define the student’s misconception, and these systems have no ability to adapt to the student’s state. Thus, they present the same instructional materials in exactly the same sequence and the same feedback as in a ”one-size-fits-all” approach, although students with different levels of knowledge and different preferences require different types of instruction.

Later, CAI systems have been improved as Intelligent Computer Aided Instruction systems(ICAI) [13,14]. These systems have the ability to control the material shown to the student. Hence, ICAIs have the ability to interact with the learner and can also adapt the dialog by manipulating different forms and viewpoints of information as an expert system model. In more detail, some information from the problem-solving expertise component is presented, the student responds in some way to the material that was presented, then the system analyzes the student’s responses and decides the course of action. However, ICAIs do not take into account different aspects of the student module. Also, the teaching material became too large to be manageable through straightforward programming, so a special branch of programming languages, called ’author languages’, were developed for creating ICAI material.

Generally, the central problem with early systems was that they were unable to provide individualization or rich feedback, because they were not designed to consider and understand the teaching course, the student states, or the teaching strategies. These drawbacks gave rise to a new generation of education systems including more intelligent factors, in order to increase their effectiveness. They are called Intelligent Tutoring Systems (ITSs) [15]. Intelligent Tutoring Systems have the ability to take into account the course domain and the student knowledge and skill level. Depending
on this knowledge, the pedagogical module of ITS can provide the individualization of the leaning process and the adaptation of presented materials to the needs and abilities of the student. Different Artificial Intelligence (AI) techniques are used to represent pedagogical decisions as well as the course domain knowledge and information regarding each student. ITSs were usually developed as stand-alone systems. The goal of various ITS is the utilization of knowledge about the domain, the student, and about teaching strategies, to support a flexible and individual learning process.

Adaptive hypermedia was a relatively new direction of research concerning the crossover of hypermedia and user modeling. Adaptive Hypermedia research can be traced back to the early 1990s. The combination between Adaptive Hypermedia Systems (note that hypermedia usually combines hypertext and multimedia techniques) (AHS) and intelligent tutoring systems is known as adaptive Educational Hypermedia System (AEHS). At that time, the two main research topics Hypertext and User Modeling had mastered to a level that allowed the cooperation of research ideas. It started to overcome the limitations of traditional “static” hypermedia applications of providing the same page’s content and the same set of links to all users. Thus, the traditional educational hypermedia system presents the same knowledge of the subject, static explanation and suggests the same next page to students without considering the student’s goal or level.

The adaptive hypermedia is an alternative to the traditional “one-size-fits-all” approach in the development of hypermedia systems. It builds a model of goals, preferences and knowledge of each individual user, and uses this model throughout the interaction with the user, in order to adapt to the needs of that user.

The year of 1996 can be considered an impact point in adaptive hypermedia research. Before this time the research in this area was performed by a few isolated teams. However, since 1996 adaptive hypermedia has gone through a period of rapid growth [16]. Adaptive Hypermedia Systems (AHS) are defined as “[all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user. In other
words, the system should satisfy three criteria: it should be a hypertext or hypermedia system, it should have a user model, and it should be able to adapt the hypermedia using this model” [17].

The main services presented to the user are an adaptive presentation of the teaching content and adaptive navigation by adjusting the page hyperlinks. The adaptive education hypermedia systems provide guidance by presenting the next convenient learning step to follow, or by individualizing the creation of the learning sequence. In addition, it supports the student’s progress during the learning process. AEHS systems dynamically construct or adapt the educational pages in contrast to ITSs in which the contents of the educational pages are typically static.

The first ”pre-Web” generation of adaptive hypermedia systems explored mainly adaptive presentation and adaptive navigation support and concentrated on modeling user knowledge and goals [18]. Such that adaptive navigation can increase the speed of navigation and learning, whereas adaptive presentation can improve content understanding. The emergence of the web revolution gave rise to the e-learning generation. It becomes possible to provide unlimited access to teaching materials. However, the learner on the web faces the problem of doing confronted by a huge number of materials and hyper links which could be related or unrelated to the student’s goal. So, various approaches exist for supporting the e-learner’s navigation. The research on e-learning and web-based educational systems traditionally combines research and efforts from various fields. All these systems aim to personalize the course offered to the users. The fundamental research for the adaptive web has been manipulated in the area of adaptive hypertext and hypermedia [19].

The second ”Web” generation extended the approach of adaptive hypermedia by exploring adaptive content selection and adaptive recommendation based on modeling user interests [20]. Thus AEHS are an innovative approach to the web learning experience. The technique of Adaptive Educational Hypermedia [21] is especially developed for adapting a course on the web. They tried to overcome main shortcomings of the classical hypermedia e-learning applications of ”lost-in-hyperspace” phenomena by adapting the learning Content and its presentation to needs, goals, and
learning styles of every individual learner. Representatives of such systems are adaptive textbooks constructed with AHA! [22], InterBook [18] and Net-Coach [23], or adaptive courses within ELMART [24] and AIMS [25]. Semantic Web technologies achieve an improvement w.r.t. adaptation and flexibility of the courseware. Thus, Semantic Web technologies and ontologies have begun to be used in hypermedia applications. These technologies are becoming popular in the field of adaptive hypermedia because they provide the means of overcoming the interoperability problems that connected with the current adaptive systems (Interoperability means that the digital learning resources will plug and play on different platforms). Also, Semantic Web technologies provide collaboration, the exchange of content and the reuse of functionality.

### 2.2 Intelligent Tutoring System Architecture

The architecture of ITS may vary according to the designers’ needs [26]. The components of the architecture of the educational system in CAI first were combined in a single structure. This caused a problem in the modification process. The improved developing architecture consists of three main modules (as depicted in figure 2.1):

1. The course module, containing the course content and information about the teaching subjects with different relations between these subjects.

2. The student module, which records information regarding the user.

3. The pedagogical module, containing knowledge about pedagogical aspects.

These three components play a major role in the functionality of the ITS. Some system add a fourth one as a communication or an interface module between student and the ITS. Each of these modules will be described in the following subsections.
The Adaptive hypermedia system is based on the basic form of hypertext systems of the Dexter Hypertext Reference Model [19]. The architecture of adaptive hypermedia systems has two main layers: the Runtime layer and the Storage layer (2.1). In the Runtime layer the learning material is represented to the user’s and the user performance is observed. The storage layer controls all of the adaptation process. The Adaptive Hypermedia Application Model AHAM [27] (see Figure 2.2), provides a framework to express the functionality of adaptive hypermedia systems by dividing the storage layer into three parts:

- **Domain model**: defines what to be delivered and what to adapt? (contains the content structure description).

- **User model**: contains information about the student, according to which the system can be adapted.

- **Adaptation model**: defines how the adaptation should be performed (contains adaptation rules which define how the adaptation is performed). AHMS uses condition-action rules to define its rules.
Table 2.1 Dexter hypertext reference model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Layer</td>
<td>Presentation of the hypertext; user interaction; dynamics</td>
</tr>
<tr>
<td>Presentation Specifications</td>
<td></td>
</tr>
<tr>
<td>Storage Layer</td>
<td>a &quot;database&quot; containing a network of nodes and links</td>
</tr>
<tr>
<td>Anchoring</td>
<td></td>
</tr>
<tr>
<td>Within Component Layer</td>
<td>the content / structure inside the nodes</td>
</tr>
</tbody>
</table>

2.2.1 Course Module

The Course module contains knowledge regarding the course being taught as well as the actual teaching materials. There are two structures have been considered; the course structure and course content. The course structure refers to the basic concepts of the course concerning the domain. Every concept has a number of general attributes such as the concept_name, concept_level etc. Also, these concepts can be linked together by different relations among them representing the network structure of the course. Whereas the course content contains the teaching materials presented to the students. The pedagogical module uses the course knowledge to select appropriate parts to the student. For instance, knowledge may indicate the difficulty of relative curriculum sections or exercises.

The course module was represented or modeled by different forms, for instance; an AND-OR graph is used by Vassileva [28]. It is the simpler representation schema containing (AND) links defining conjunctive prerequisite relations and (OR) links defining disjunctive prerequisite relation. In hypertext technique the course module consists of a set of text nodes which connected by
links. Each node (as text) contains information and can be connected to other node(s) using link(s).

The hypermedia technique is an extension to hypertext which uses multiple forms of media, such as text, video, audio, graphics, etc. In AEHS there are two kinds of adaptation: adaptive navigation and adaptive presentation. **Adaptive presentation** adapts the contents of an educational page to the user characteristics. In this method, the various pieces of information formulating the course materials and associated with conditions. When conditions are achieved, the corresponding information is presented. This way some information is hidden if it is incompatible with the user’s characteristics. One of the popular methods of adaptive presentation is known as *additional explanation* [29]. This method provides different text or media with additional details or prerequisite explanations with different sorting of information. Another method of the adapting presentation technique concerns the explanation variants that stores variations of the educational content and
selects the most suitable one according to the user model. This method can be implemented in two ways. The simplest way is to use page variants. In this way, the system contains variants of the same page with different presentations for the same subject. A more specialized way is to use fragment variants. This way is useful when a page refers to more than one knowledge concepts. These two ways can be combined in order to enable adaptation according to multiple user characteristics. For example, the appropriate page can be selected from the page variants according to the user’s background and the suitable fragments according to the user’s knowledge level. The course generation approach can be classified as an adaptive hypermedia system.

In Adaptive navigation the links are adjusted towards the optimal path through the learning material for a given user [29]. In other words, the links are adapted in a personalized way when the user navigates through the hypermedia, which assists users through navigating the hyperspace of the educational system. The usual methods applied for adaptive navigation are the following:

- **Direct guidance**: This is the simplest adaptive navigation method. It shows the next best course material to access.

- **Link sorting**: The links of a specific page are sorted in order of decreasing relevance.

- **Link annotation**: To select the appropriate page, links are annotated using linked colors, icons, ect.

- **Link hiding, removing, disabling**: The links with low interest are hidden and presented as simple text, totally removed or disabled.

To facilitate the selection and ordering of the course content, it is important to represent relationships between hyperdocuments as well as metadata descriptions. There are standards for the metadata description of the course content such as IEEE LTSC Learning Object Metadata\(^1\) or

\(^1\)(http://ltsc.ieee.org/)
Chapter 2 Background

Doblin Core\(^2\).

However, the course module structure in AHAM has strong limitations when it comes to describing concepts and relationships between them. In more details, the domain model is exclusively composed of concepts (atomic or composite). This appears to be a very strong limitation since (i) it seems that no typology of concepts exists to sustain this model and (ii) that there is no way to describe any related elements that is not a concept. Another strong limitation resides in the kind of links that are used only "part-of" or a "prerequisite". Several reference architectures for adaptive hypermedia systems do exists, for instance in adaptive hypermedia systems like InterBook [16], MetaLinks [30], and NetCoach [31], the values of meta-data attributes denote the document space. Thus, the document is annotated with a set of documents that have a specific role (e.g. prerequisites, successor, part-of), such that the document’s space follows the meta-data annotation. When new documents are added, deleted, or modified, the document’s annotated meta-data regarding its size could be invalid because of a change in the document’s length.

The difference between modeling the course module in ITSs and e-learning systems is the use of learning objects in e-learning systems. In these systems the concepts are first defined by the author and then linked to teaching resources [32]. The teaching resources contain the knowledge and skills defined in the course module. This helps the system to choose the appropriate learning objects to achieve a particular goal. But making this connection is difficult. To solve this problem an ontology is used to specify the concepts in the course and the relationships between these concepts. This ontology is shared with the learning objects [33]. By employing semantic web technologies, the knowledge space of domain models can be represented by using domain ontologies as a formal specification of the conceptualization of the shared domain. Currently, the Semantic Web community is adapting a standardized language (the Web Ontology Language, OWL) to sharing ontologies.

\(^2\)(http://purl.org/DC/)
In this thesis, we will consider different components of the adaptive hypermedia application model, and we will apply ontology and semantic web technologies to improve their interoperability. Note that, the document space in semantic web is dynamic. By using OWL, adaptive hypermedia systems can share their domain models, as we will explain in more detail in chapter 4.

2.2.2 Student Module

The student module records information regarding the student and represents the beliefs of the system about the student. There are different possibilities for the user’s characteristics that can be recorded in the user model. Each system chooses the most appropriate ones to its case. But if the student model is incomplete, the system’s adaptability will be unsuccessful.

The student module is responsible for representing the student’s understanding of the subject being taught, the student’s preferences, her knowledge, goals, performance history and other relevant aspects.

The process of inferring a student’s states from observable behavior is called diagnosis. This kind of information is stored for each learner. Based on this information the system gives an individualized instruction to the student. In general, the student’s characteristics that are recorded in the student module are as follows:

- **Student performance history:**

  Information about the student’s interaction results is known as student’s performance history. The student’s performance history is being noted and is used by the pedagogical module to take the appropriate teaching actions. For example, if the student’s performance on some topic is poor, the ITS selects similar problems until the student’s performance increases. The system needs to build a learner model and update it while the tutoring process goes on. What the student knows, what s/he does not know, and what any the misconceptions are, is also used to access the student’s knowledge. The system keeps track of the student’s learning
abilities w.r.t. the topic being taught.

- **Cognitive state:**
  Describes the learning ability of the student and the cognitive level during the learning process.

- **Student’s goals:**
  There are two types of goals that can be considered during the learning process. High level goals e.g. learning goals and low level goals e.g problem solving goals. These goals could be changed during the learning process. Also, the student’s goals can be modeled as a set of concepts.

- **Student’s preferences:**
  This term refers to all of the student’s preferences during the learning process. Usually, it refers to the preferred learning styles or media types (e.g. text, image, audio or video). Sometimes, also the student’s preferred teaching strategy as well as the preferred language.

- **Background or experience:**
  The student’s background refers to the experiences regarding the goal of the teaching course, which is important to student to concentrate on. This information includes the student experience in other relevant fields, experience in using computers, or familiarity with the system, etc.

- **Knowledge related to the teaching course:**
  The student’s knowledge is regarded as the most important characteristic for ITS or AEHS. The different models for the student module are:

  - **Overlay model:** This is the most popular way of representing the user’s knowledge.

    This model is based on the pedagogical structure of the domain knowledge (i.e knol-
2.2 Intelligent Tutoring System Architecture

This model is designed to represent the student’s knowledge state as a subset of a teacher’s knowledge state [34]. For each concept, the model returns a value representing the user’s knowledge level regarding this concept. This value is Boolean (known, not known) or scalar. Based on the overlay model the system presents the educational content until the student’s level of knowledge reaches expert’s level. The overlay model has been used very often in ITSs. A disadvantage of this model is its inability to represent student’s misconceptions.

- **The buggy model**: It represents the student’s misconceptions as variants of the teacher’s knowledge [34]. This model represents the user’s knowledge as the union of the subset of the domain knowledge and the set of misconceptions Figure 2.3. The buggy model provides great support in the correcting the user’s mistakes. There are two variations of the buggy model: the bug catalog and the bug-parts-library model. In the bug

![Figure 2.3 Overlay student models](image)

---

In thebug-parts-library the student’s misconceptions are created during the teaching
process from a library to bug parts. The library usually contains symbolic rules with conditions and actions executed when the condition is satisfied. So, the second type is simpler and easier to be constructed.

– **Stereotypes:**

In this model a predefined classes of students is defined to categorize the student. It represents as a set of pairs (stereotype, value) where the value is a Boolean value specifying if the student belongs to a specific stereotype or not. This model is simpler than the other models and is easier initialized and maintained. A disadvantage of this model is the difficulty in setting boundaries between different stereotypes.

– **Historic model:**

This model trade of the learner’s visiting state of individual pages. This model is a commonly used in adaptive hypermedia systems.

The combination of these models can be considered when modeling the student model. For example, the combination of the stereotype and overlay models are often used. Here the student is initially categorized by stereotype and then the model is generally modified according to the overlay model.

Different models are used to model the student modules. For instance, uncertainty models are employed to model the student’s knowledge [35], these models use fuzzy logic, Bayesian networks, neurofuzzy representations, etc. Also, constraint-based modeling [36], which represents the student’s knowledge as a set of constraints. There is a standard related to the user modeling in general, but also a domain specific one, as IEEE Public and Private Information or IMS Learner Information Package. The ontology language is used to enable sharing of the user models, but most of the applied systems are based on the application specific user model needs and more likely they will not be suitable for other application domains.
2.2.3 Pedagogical Module

The pedagogical module acts as the backbone of the tutoring system. It represents the teaching process and supports the infrastructure knowledge for tailoring the course structure and content according to the information contained in the student’s module. It decides what to teach, how to teach, when to review, when to present a new topic, and which topic can be present, in accordance to the student module.

Different courses can be generated according to the needs and the characteristics of different students and depending on various relations between concepts and the description of the course materials. In addition, the pedagogical module considers tutoring strategies in its modeling which define how the teaching subject should be taught. Thus, it is important for an ITS to support various types of tutoring strategies as not to be restricted in its pedagogical view. It is also important to integrate an effective mechanism or effective plan for selecting the appropriate strategy during the teaching process. Different scales for selecting the appropriate strategy can be considered such as the student’s achievement, preferences of course materials, etc.

The pedagogical plan ranges from low level tasks (such as selecting questions, the difficulty of a problem and presenting hints) to high-level tasks (such as selecting the next concept for the student). Another important aspect of the pedagogical model is the evaluation of the user’s performance based mainly on the student’s answers.

Different types of ITS technologies are identified: curriculum sequencing, intelligent analysis of the student’s solutions, interactive problem solving support and example-based problem solving support.

The goal of curriculum sequencing (or course generation) technology is providing the student with the most suitable, individually planned sequence of concepts and learning tasks (examples, questions, problems, etc.) to work with. It helps the student finding an optimal path through the learning material (more details about the curriculum sequencing technique will be introduced in
Intelligent solution analysis deals with the student’s solutions of educational problems (which can range from a simple question to a complex programming problem). Unlike non-intelligent checkers which can only tell whether the solution is correct or not, intelligent analyzers can tell what is wrong or incomplete, and which pieces of knowledge are missing or incorrect. Intelligent analyzers can provide the student with extensive error feedback and update the student model.

The goal of interactive problem solving support is to provide the student with an intelligent help on each step of problem solving - from giving a hint to executing the next step for the student.

Finally, The example-based problem solving technology is the newest one. This technology helps students solving new problems not by articulating their errors, but by suggesting the relevant successful problem solving cases from their earlier experience. The main techniques presented to the user in AHMS are adaptive presentation of the teaching content and adaptive navigation by adapting the page hyperlinks. Thus, in addition to adaptive hypermedia technologies, the ITS technologies are applied in the context of Web-based education (WBE). Curriculum sequencing technology becomes very important for its ability to guide the student through the hyperspace of available information. Curriculum sequencing was one of the first techniques implemented in such Adaptive Intelligent Web based Educational System (AIWBES), as ELM-ART [31] and KBS-Hyperbook [37]. Intelligent solution analysis technology was also one of the first techniques on the Web in such AIWBES, as ELM-ART [31], the systems SQL-Tutor [38] or German Tutor [39]. Interactive problem solving support technology is not as popular in Web-based systems as in standalone intelligent tutoring systems mainly due to implementation problems.

Finally, in addition to that, ELM-ART [31] provides a unique example of example-based problem solving support.

Other different adaptive hypermedia technologies are applied, such as adaptive information filtering, intelligent collaborative learning, adaptive group formation and peer help, adaptive collabora-
In adaptive hypermedia systems the condition actions rules are used as adaptation rules. Recently, the semantic web rule language (SWRL) has been used for representing and sharing the rules in the semantic Web [40]. The related subject to this part are the course generation systems and instruction design theories.

### 2.2.4 Course Generation

Course generation as a research field has long been also known as curriculum sequencing. The course generation process is based on curriculum sequencing technology (the term also refers to instructional planning technology as we will explain in the next subsection). The main idea of course generation is to generate a course in accordance to the teaching material, student states and goals to generate an adapted sequence of resources that provide the student in achieving her learning goals. The curriculum sequencing technique aims to provide the student with a suitable and individual plan. This plan consists of a sequence of course units and learning tasks (example, exercise, definition, etc.) which helps the student find the optimal path through the learning process. Based on the constructed plan, the course materials are selected, ordered and presented to the student. Several empirical studies investigate that students benefit more from the organized contents that form according to pedagogical principles.

In course generation systems, the course module describes the course concepts and the related materials for each concept. The student module records the developing performance progress and other information about the student. The pedagogical module contains teaching and learning strategies which define how to assemble a course by using information provided by the course module and student module.

Mainly, there are two kinds of sequencing: Active and Passive. The *active sequencing* concerns the learning goal (the course concept(s) needed to be learned). Thus, the system can build the best
Chapter 2 Background

individual path to achieve this goal. Systems in active sequencing can be categorized into those with fixed and adjustable learning goals. Most of the existing systems can guide the student to the fixed learning goal, however a few systems let the teacher or student selects a specific goal as a subset of the whole set of concepts as the current learning goal. On the other hand, the passive sequencing is a reactive technology and does not require an active learning goal. Thus, it aims to help the student to select a subset of the available teaching material, which can fill the gap in the student’s knowledge of achieving a learning task; for instance it can help the user when s/he is not able to solve a problem or answer a question correctly.

Currently, most web-based educational systems use sequencing. Active sequencing is the most common type of sequencing. Only a few systems (InterBook, CALAT, Prolog Tutor) can perform passive sequencing. Among active those using sequencing, only a few systems are comping between high and low level of sequences as ELM-ART, KBS-Hyperbook or ART-web. Finally, most of the systems provide fixed learning goals. Only a few systems provide adaptive learning goals giving the opportunity to the teacher (DCG) or student (InterBook, KBS-Hyperbook) to select an individual goal.

Course generation is considered as a middle ground between two types of systems: On one side the pre-authoring systems, ”one-size-fits-all”, in which the same course is delivered to all students regardless of individual differences, goals or levels. On the other side individual look-up learning objects as hypermedia systems in which the student searches for resources by his/her own. Thus, the student should be self-aware and organized enough to be able to evaluate and structure the retrieved content.

Two main approaches are defined in the Course generation field [32]: the adaptive course generation and the dynamic course generation. In adaptive course generation the goal is to generate a personalized course considering the student goal and state. The generated course is generated before it is delivered to the student. That could help the student to realize the complete view about the
2.2 Intelligent Tutoring System Architecture

requested concept which gives the student the free navigation ability through the presented part. On the other hand, in the dynamic course generation approach the system observes the student’s progress during the interaction with the presented course part and adapts the presented part or delivers the next part according to the student goals and previous knowledge. However this thereby is fit the changeable state of student, but at the same time may cause confusion to student when s/he move from one part to another and may not get the complete idea.

Note that two levels of sequences are defined in most ITS: high and low level. High-level sequencing or knowledge sequencing determines the next learning subgoal; as next concept, set of concepts, or lesson to be taught. While low-level sequencing determines next learning task (example, exercise, test) within current subgoal. These levels are often performed by different mechanisms. In many ITS only one of these mechanisms are performed.

Most of the earlier approaches used a number of AI techniques. Early approaches on course generation back to the eighties [41]. They introduced techniques which is considered as the base for course generation systems. They encode the pedagogical knowledge as planning operators to determine sequences of the instructional course. The learning material is structured in concepts, and prerequisite knowledge, which states the causal relationships between different concepts is defined. Then planning techniques are used in order to find plans that achieve the learning goals using a STRIPS-based planner. Moreover, Murry [42] used learning materials repositories to support the tutoring process. This technique later become the basic technique for course generation systems. In addition, formal plans are retrieved from a plan library in [43], or expert systems in [44].

Van Marcke [45] introduced the Generic Tutoring Environment (GTE) system. The instructional knowledge in this system was divided into instructional tasks, instructional methods, and instructional objects. Tasks represent achieves that have to be accomplished during the teaching process. Methods decompose tasks into subtasks until a level is reached where only primitive tasks remain. The generated solution from the decomposition process is known as a task structure. GTE imple-
ments its algorithm based on precede signals, but the complexity of this technique is not known and the selection of methods is done using ratings encoded within the methods. Vassileva [46] introduced a system similar to GTE called Dynamic Course Generation (DCG), as one of the earlier systems. She introduced the approach and an architecture of Dynamic Course Generation based on AI planning to structured representation of the domain knowledge and allowing explicit representation of teaching rules as if then rules. DCG differentiates between domain concepts and educational resources. The domain structure is represented by an AND-OR graph. The education resources are stored as HTML pages and linked to the domain concepts. Different if-then rules are considered for adapting the pedagogical module which contains teaching strategies for course sequencing. The course generation in DCG is divided into two main processes **Content Planning** and **Delivery Planning** based on Wasson [47]. Such that, according to the concept goal, the **Content Planning** generates paths through the AND-OR graph using AI-planning to search for the paths and generate a plan. Then, **Delivery Planning** or presentation planning selects the educational resources and the order of these resources that are linked to the concepts which are generated by the content planning process. DCG also allows the author to shape the structure of the course and the related educational materials, and to define a set of adaptation rules to determine an appropriate teaching strategy for each student (or group of students). However, the generation of the learning path is based on simplistic planning techniques and heuristics while the variety of learning objects is quite limited (i.e. HTML resources). Also, as a result of the separation of content and presentation planning, presentation planning considers the selected concept. For instance of considering the definition of the prerequisite concepts.

The Adaptive Course-ware Environment (ACE) [48] presents a similar approach as DCG and suffers from the same drawbacks w.r.t. separation between the content planning and presentation planning. An additional drawback is that the rules of presentation planning that are attached to the individual concepts are not general rules for generating the learning objects. Related to the
same drawback, the author needs to specify the path through the domain structure which it is not generated automatically.

In addition, today’s course generation focuses less on pedagogical knowledge, but more on semantic web and metadata, e.g., on using an ontology of the subject domain to automatically calculate the best path through the learning material [2]. These approaches use rather simplified pedagogical knowledge, e.g., to select those learning objects with the lowest typical learning time. However, generating a course which is adapted to the individual learner’s goals and needs and which is based on state of the art of pedagogical strategies requires more elaborate expertise. So, there are a handful course generation systems that are based on HTN planning. Me’ndez et al. [49] used HTN planning for course generation in an e-learning environment. Each concept is considered to be a complex task that can be expanded by a method $m$. The preconditions of $m$ are the prerequisite of the concept. While the subtask of $m$ are the concepts that linked to $c$ by part-of relation. Furthermore, for each educational resource, a method and an operator are generated. The precondition of the method is the student’s learning style, and the single subtask consists of the educational resource. While, the precondition of the operator is the prerequisites of the educational resource, and the add list consists of the concept associated with the educational resource. Although the domain structure is defined by the way that helps the system to generate different courses for the given concepts with different learning styles, the pedagogical knowledge that is encoded in planning domain is limited. The PASER system is introduced by Sicilia et al. [50] they show clearly that course generation techniques can be used for the generation of learning designs. They present an idea of using HTN-planning to generate IMS LD instances based on a very abstract level representing some basic operators and methods. However, they do not provide different adaptation forms in the generated course for student. Finally, Ulrich [51] introduced a course generation system which is able to generate a course adapted to different learning goals based on the HTN planning framework using the Shop2 planner. In this system, pedagogical objectives are represented as tasks and the
ways of achieving these tasks are represented by methods. Also different learning scenarios are considered and the generated course automatically assembles learning objects retrieved from one or several repositories.

2.3 Learning Theories

Learning theories address how people learn thereby helping us understanding the complex process of learning. There are two main types of learning theories that describe specific methods of instruction for different cases of a student’s status: Descriptive and Prescriptive theories. Descriptive learning theories describe how learning occurs and contrives models that can be used to explain and predict learning results. There are three famous categories of descriptive learning theories: behaviorist, cognitive, and constructivist.

On the other hand, Prescriptive learning theories are considered guidelines that describe what to do in order to achieve specific outcomes based on descriptive theories. The instructional design is categorized under prescriptive category.

First, we will describe the learning theories:

- **Behaviorism** This approach was started by Pavlov and improved by Thorndike, Waston and others [52]. Behaviorism focuses on the human behavior that can be observed and measured, without taking into account mental processes. The mind is seen as a black box, regarding the learning process. The response to an action can be measured quantitatively. In the tutoring system notion is reflected condition-action pairs. The positive reactions have to be reinforced, and the undesired ones avoided. Viewed as a question answer system: If the answer is right the system continues with the next question, if not it gives a hint or ships the question, without measuring the student’s understanding level.

- **Cognitivism** In the early 1920’s people began to investigate the limitations of the behaviorist
approach in trying to understand the learning process. Cognitivism emphasizes on the acquisition of knowledge and the internal mental structure. In other words, it focuses on how information is received, organized, stored, and retrieved inside the mind. Cognitivists consider how human memory works trying to promote learning. The physiological processes of sorting and encoding information or events from short term memory to long term memory are important for educators working under the cognitive theory. Learning takes place through organizing and linking the new structures to old knowledge.

- **Constructivism** Constructivism is based on the principle that knowledge cannot be transmitted but has to be constructed by the individual. Constructivism views learning as a process in which the learner actively constructs or builds new ideas or concepts based upon the current and past knowledge or experience. Therefore, learning is an active process of integrating information with pre-existing knowledge. The constructivist sees the learner as more than just an active processor of information. Learners create their own meaning of knowledge. Therefore, the goal of instruction is not to know particular facts but to elaborate and interpret information. The prescriptive learning theories can be described as follows;

- **Instruction Design** Instruction design is defined as "a construct referring to principles and procedures by which instructional materials, lessons, and whole systems can be developed in a consistent, reliable fashion" [53]. Instruction design provides guidelines for authoring the learning material. Thus it supports the student in achieving learning goals. Instruction design in the context of intelligent tutoring is seen as instructional planning. The instructional planning is the process of planning the global sequence of instructional goals and actions that provide consistency, continuity and coherence in the instructional process. The goal of instructional planning is to generate individual sequences of learning material using appropriate presentation formats. The instruction planning approach was proposed by Wasson [47]. Two levels are defined in instruction planning: Content Planning and De-
Content planning is the process of deciding the content of the delivered part [47, 54]. While Delivery Planning is based on selecting and sequencing tutorial instructions based on the content already chosen [55]. The design of instruction strategy involves a number of decisions including selecting the content’s segments, sequencing these segments, selecting the appropriate instructional objects (example, definition, exercise,..), sequencing these instructional objects, and configuring a given architecture for a given student. The instructional planner needs different kinds of knowledge: First, knowledge about the concept to be learned and the relationships between these concepts. Second, Knowledge about student. Andy, knowledge about the pedagogical rules.

Clearly, the students could benefit a lot if there was a system that supported them to perform learning tasks as memorization, evaluation, retrieving information. Providing students with such support would help the students to construct their knowledge and help them to learn in an efficient way. There are different architectures of instructional design which are designed to help students learning more effectively [53]. For instance, Bloom [56] developed a taxonomy for providing a foundation of instructional design. Bloom identified six levels of the cognitive domain, from simple recall or recognition of facts to the highest level, the evaluation of material. The most important outcome is the learner’s ability to move up in the ”pyramid” and begin to achieve new knowledge and deeper levels of understanding. See figure 2.4.

Based on the taxonomy proposed by Bloom, Gagne [57] proposed a taxonomy of learning outcomes based on the cognitive theory of learning. Robert Gange was a leading scientists regarding learning theory. He lists of nine instructional events for planning lessons: 1) Gaining attention, 2) Informing students about the objective, 3) Stimulating recall of prior learning, 4) Presenting the Content, 5) Providing learning guidance, 6) Eliciting performance, 7) Providing feedback, 8) Assessing performance and 9) Enhancing retention and transfer.
Merrill [58] later elaborated and extended this taxonomy towards the component display theory (CDT) based on constructive learning theory. CDT uses a performance content matrix to specify outcomes categories. The matrix has five dimensions (fact, concept, procedure, principle, and skill) and three performance levels (remember, use, and find). Hence, specific relationships between the content dimension and instruction dimension are defined in CDT. Such that, the content dimension has two values: generality and instance. The instruction dimension has presentation (expository) and application (inquisitory) values. These led to four primary presentation forms (PPF): Present Information = expository generality (IG), Remember Information = inquisitory generality (IG); demonstration = expository instance (EG); application = inquisitory instance (IEG).

Over the years, the author has come to adopt less terms as will be explained in chapter 4. Whichever approach is used to specify learning outcomes, it is important to consider learning outcomes during ontology design which is our goal, as will be illustrated in this thesis.
2.3.1 Interface Module

The interface module is an intermediary between the student and the tutoring system. Using the interface the system can display information, teaching materials or feedback. It can interact with the student, accept the student’s inputs or solve exercises.

2.4 Authoring Systems

In ITSs, the course module is a result of a collaborative effort of the knowledge engineer, the programmer, and the teacher as an expert. The knowledge engineer is responsible for interacting with the teacher to extract the main architecture and knowledge. Then, the knowledge engineer and the programmer work together to produce the course model. A common drawback that was observed in ITSs is the inflexibility of these systems and the complexity of producing an educational system. This makes it difficult for the developer to change or modify the teaching materials and led to a new category, called Intelligent Tutoring System Authoring System (ITSAS). These authoring systems solve the problem of domain dependency in ITSs. Accordingly, the instructor as an author plays an important role in increasing the effectiveness of the produced ITS.

The authoring system is a computer based system that allows the authors as teachers (non-programmer) to use the authoring tool for constructing or authoring content and for configuring different aspects of the course model in intelligent tutoring systems. Thus, the author as a teacher does not need to have programming skills.

Murray [59] in his paper on the state of the art of authoring systems classifies the existing systems according to their functionalities and capabilities. He explains the strengths, limitations and variations of each category. Generally, he categories the ITS authoring systems into two categories. The pedagogical-oriented and the performance-oriented. The pedagogical-oriented systems concentrate on representation of the teaching strategies and techniques in ITS, which specify the ways
of sequencing and teaching the course content. This includes systems for sequencing and planning a curriculum, authoring tutorial strategies, composing multiple knowledge types (e.g., facts, concepts and procedures) and adaptive hypermedia. While, the performance-oriented systems concentrate on providing a valuable learning environment with more practice, skills and feedbacks, providing rich learning environments where students can learn by solving problems and receive dynamic feedback. Students use these systems to solve exercises or problems and to receive feedback depending on their answer. These systems need a deep model of expertise, which enables the system to correctly evaluate the student’s answer as well as to provide assistance on solving a problem. Most authoring systems for these systems focus on generating rules that form the knowledge model or the course model. It is worth mentioning that Murray has not referenced a system that combines the two categories.

However there is a deep conceptual gap between authoring systems and authors. So that building an instructional course still requires a lot of effort. Mainly because of the need to define the course material and to consider different instruction design theories and learning strategies. The course content, the student’s cognitive state, the appropriate teaching strategy, level of help for each student and other different configuration details for each student [60] have to be defined. Also, it is difficult for the author to build different courses for different of the student’s states. In addition, most of these systems are special purpose systems and lack the sophisticated requirement to build intelligent tutoring [59]. On the other side commercial authoring systems have a shallow representation of pedagogical strategies and theories. Some special purpose tutoring systems provide quite extended authoring guidance. However to adapt or meta changes to these systems is not very easy and not all the content can be reused for other educational purposes [59].

In the following subsection we will describe only a few authoring systems along with considerations concerning our aim to decrease the difficulty and effort of the authoring process.
2.4.1 Authoring Tools for Intelligent Tutoring Systems

Intelligent tutoring systems as we have described them before consist of four main components [26]: The course module, the student module, the pedagogical module, and the user interface module. Thus, ITS authoring tools offer one or more of these components. Therefore, the differentiation among various authoring tools are ordinarily based on the components that the system allows to author [61] and also depends on the type of ITS that they produce [59]. For instance, the REDEEM tutor [60] is an authoring environment which helps the author to convert an existing course into an ITS that provides distinguished and adaptive instructions concerning the pedagogical module of ITS. However, the REDEEM focuses on the representation of instructional strategies, not on the generation of these strategies. In other words, the author can select their own pedagogical parameter setting from fixed rules defining the pedagogical behavior or meta-strategies. Such as the amount of help, preference style (general, specific), the necessity of prerequisite, etc.

Moreover, in the REDEEM the author has to describe existing teaching material (tutorial pages) in terms of their difficulty, their generality, etc. In order to construct the parameters of the teaching strategies. In spite of REDEEM has the feature of reusing domain material, but it does not consider the standard metadata which increase the reusability of the learning materials. In addition, classification of students was done using a simple model, considering the student’s performance level (for example, novices, advanced learner) and the author should define the parameters for each category, for instance the advanced student might start with general information following by specific information with help being only available in case of an error. However other aspects of the student are not considered.

Eon [62] allows authoring of every component of the ITS. The student module, the course module, the pedagogical module, and the interface module. In Eon the course knowledge is authored by defining the set of topics of the course considering the property for each topic (importance, difficulty, etc.) and the relationships between topics (prerequisite, part of, etc). Accordingly, the author
2.4 Authoring Systems

could create the topic network. In addition, the author can define teaching strategies using a flow line-based graphical programming language that allows the author to author arbitrary instructional processes. The author can also specify rules about the teaching material sequence, e.g. when and how to test the student, the explanation or hints availability, student evaluation or assessments. However, the problem with this system is that there is no guidance given to help the author creating effective tutoring strategies.

COCA [63] uses a rule-based representational method, and the author uses pull-down menus to specify the right and left-hand components of IF-THEN rules. Based on AHAM [27] a different system was defined resulting in; LAOS [64]. In this system the following layers are defined:

- **The Conceptual layer** expresses the domain model (with sub-layers: atomic concepts and composite concepts - with their respective attributes),

- **The Lesson layer** has multiple possible lessons for each concept map or a combination of concept maps,

- **The Student adaptation and presentation layer** which is based on the adaptation model and the presentation model.

The objective of this system is to support the reusability at various levels, focusing mainly on adaptation strategies and techniques. LAOS draws a distinction between the primitive information (content) and presentation-goal related information.

The Ontology-based Intelligent Authoring Tool [65] uses an intelligent training system in the E-Learning scenario. It uses four ontologies (domain, teaching strategies, learner model, and interfaces ontology) for the construction of the learning model and the teaching strategy model, but it fails in exploiting modern web technologies. From the above discussions it is clear that the design of an ITS authoring system should be generic, so that courses from a large set of domains can be configured. It can also be inferred that authoring systems which can reuse existing material can
be more effective, since authoring of the tutorial easier and less time-consuming. In addition, we need a generic authoring tool for all domains so that courses from a larger set of domains can be configured. The authoring system should also consider different aspects of the student performance and characteristics. It is also important to consider the reusability of teaching material since authoring of the tutorials will be easier and less time-consuming. Thus, it enhances the adaptability of the authored tutorial. All these lead to the requirement of the authoring tool having to reduce the complexity of the authoring tasks and to reduce the effort of the author. In addition, currently the direction of research is toward ontologies and focuses on knowledge sharing and reusability which should be considered as a new generation of authoring system.
Chapter 3

Planning

Planning is an old and well established field. It is considered a central active field of research since the beginning of Artificial Intelligence (AI), not only because of its connection between fundamental issues in AI knowledge representation and computation, but also because of its practical importance. The output of planning system is represented by a sequence of actions that can change the current world state to desired state (goal state). In general, a planning problem is represented by an initial world state, goal state descriptions and a domain model. The latter typically is a knowledge base containing action specifications. In order to find a solution plan for a planning problem, we need a planning system, a so-called Planner. A planning system searches in the space of states for action sequences that generate a state transformation that reaches the goal state. This is a hard problem because the possible sequences in the search space can be very large and sometimes infinite. If the search space has been exhausted or the allowed maximum CPU time has been reached, no solution is found.

A large variety of approaches have been developed in the past to study the problem of constructing plans. This chapter introduces an overview of different planning approaches and then we will introduce the formal framework of our approach.
3.1 AI Planning

In principle, the AI planning research falls into four categories:

1. Classical planning
2. Partial order planning
3. Hierarchical planning
4. Hybrid planning

We will focus on these categories in the following sections.

3.1.1 Classical Planning

The first approach that has been used to study the problem of constructing plans in the classical paradigm is the STRIPS system\footnote{STRIPS \(\Rightarrow\) STanford Research Institute Problem Solver} [66]. It is a well-known planning framework and the formal representative language that was used for STRIPS is common to most classical frameworks. The STRIPS planning uses first order predicate logic language in order to encode the world state. The STRIPS planning framework has been formalized and analyzed by Bernhard Nebel [67].

In the STRIPS paradigm, a domain model formalism contains general information such as types, predicates and operators (actions) A predicate is a statement or a relation between different objects in the application domain. It may take on the values true or false depending on its parameters. A state \(s\) in STRIPS paradigm is represented by a collection of binary variables so-called facts. Facts are obtained from the predicates by instantiating their parameters with constant symbols. Note that the true facts are represented explicitly in the state, while the false facts are unspecified in the state according to the so-called Closed World Assumption [68].
An operator is the system’s representation of actions that may be executed in the application domain. Each operator $o$ consists of four parts: (i) the operator name, (ii) a set of parameters, (iii) the pre-condition ($pre(o)$), and (iv) the post-condition or effect ($eff(o)$). The pre-condition ($pre(o)$) is a conjunction of predicates. They must be true in the world state for the action to be applicable.

The effect ($eff(o)$) consists of two terms: a conjunction of the positive effects ($Add(o)$), and a conjunction of the deleted effects ($Del(o)$). However, effects (either $Add(o)$ or $Del(o)$) represent the changes in the world state as a direct result of the operator execution. It is interesting to note that $Del(o) \cap Add(o) = \emptyset$.

For example the action of traveling between two towns might be represented as follows:

Action: $LearnConcept(?b, ?d)$

Preconditions: $Registered(?b) \land \neg learned(?b, ?d)$

Effects: $learned(?b, ?d)$

where $LearnConcept$ is the action name. $?b$ and $?d$ are variable arguments representing Student name and concept name, where $Registered(?b)$ and $\neg learned(?b, ?d)$ are boolean valued state literals. They represent that the student must be registered and not learned the required concept.

In principle, the action is applied to the current state by using the transition function:

$$Result : S \times A \rightarrow S$$

Where, $S$ is the set of states, and $A$ is the set of instantiated actions.

An instantiated action is obtained from the operator by instantiating all its parameters with constant symbols. If all pre-conditions $pre(a)$ of action $a$ are true in the state, then action $a$ is applicable to state $s$, and the $Result(s, a)$ is defined. The result of applying an action $a$ in the state $s$ will remove the literals which exist in the negative effect ($Del(a)$) from the current state as well as adding the literals in the positive effect ($Add(a)$) to generated a new state.
Definition 1 (Applicable Action). The result of applying a STRIPS action \(a\) to the state \(s\) is defined as the following:

\[
\text{Result}(s, a) = \begin{cases} 
(s \cup (\text{Add}(a))) / \text{Del}(a) & \text{if } \text{pre}(a) \subseteq s \\
\text{s} & \text{Otherwise}
\end{cases}
\]

The result of applying a sequence of more than one action \(\langle a_1, a_2, \cdots, a_n \rangle\) to the state \(s\) recursively defined as the following:

\[
\text{Result}(s, \langle a_1, a_2, \cdots, a_n \rangle) = \text{Result}(\text{Result}(s, \langle a_1, a_2, \cdots, a_{n-1} \rangle), a_n).
\]

The planning problem in classical paradigm is represented by describing the known part of the world state, the so-called initial state, as well as the desired state, the so-called goal state. An initial state is a set of ground positive atoms which identify what conditions are true initially. In general, the planner aims to find out the sequence of actions in order to generate the goal state by performing these actions in the initial state. Therefore, in STRIPS formalism the plan is a solution to a given planning problem if the goals of the planning problem match a subset of the world state immediately after the last action in the plan is executed.

STRIPS planners proceed in one of two ways: The backward (regression) search strategy or the forward (progression) search strategy. The backward search strategy starts from a goal state specification, by choosing the suitable action that satisfies the current sub-goal. Applying the selected action will generate new sub-goals, and then the algorithm is called recursively with these new sub-goals. The backward search technique terminates successfully when it reaches an action that is performed directly in the initial state. The forward search strategy starts at the initial world state and search forward in time. Actions are added to the end of the plan until a state that matches the goal state is established.

In general, the STRIPS algorithm assumes that the sub-goals in the intermediate states are independent from each other and can be performed in any order. Therefore, the STRIPS algorithm
produces a totally ordered \( (\text{linear}) \) plan. A lot of approaches have been introduced to solve the dependency between sub-goals such as INTERPLAN \[69\] which tries to analyze the sub-goals in the intermediate states and then find out a sequence of sub-goals which solve the interaction between them. Afterwards, a new direction formulates the planning process as a search in the space or constructs partial plans \[70–73\].

### 3.1.2 Partial Order Plan

The STRIPS planner preserves a total order list of all actions in its plan a so-called \textit{total-order-planner} or \textit{linear planner}. As opposed to this, a partial order planner (\textit{non-linear planner}) preserves temporal constraints between pairs of actions. These temporal constraints means that an action \( a_j \) comes after an action \( a_i \), but not necessarily immediately after it \[74\].

A partial order plan can be represented as a tuple \( \langle \gamma, \zeta \rangle \) where \( \gamma \) is a set of plan steps (\textit{an instance of one of the action}) in the plan, and \( \zeta \) represents a set of temporal constraints between plan steps.

It is furthermore important to mention that, there are two special plan steps in the partial plan: the initial plan step that does not have pre-conditions and considers the initial state as post-conditions, and the goal plan step that has goal literals as pre-conditions and does not have post-conditions.

Corkill \[75\] introduced a new data structure, the so-called \textit{Procedural net} that formulates a plan as a partial ordering of actions. He inspired a new search technique in his NOAH system\(^2\), the search in \textit{plan space} instead of the search in \textit{state space} as in STRIPS planner.

As depicted in Figure 3.1 (b), the search space of a plan space planner is a set of partial plans and the plan itself is handled by adding new plan steps or constraints in order to generate new plans in the search space. As opposed to this, a state space planner (See figure 3.1 (a)) searches through the space of possible states of the world. This means, state space planners search for a path that solves the problem by using forward search or backward search techniques.

\(^2\text{NOAH} \Rightarrow \text{Nets Of Action Hierarchies}\)
Figure 3.1 State space search versus plan space search

Afterwards, Socerdoti [76] proposed new constraints between plan steps in partial order planning, the causal link constraints. A causal link constraint has the form $\langle s_i, \varphi, s_j \rangle$ where $s_i$ and $s_j$ are plan steps, $s_i$ is ordered before $a_j$ and the literal $\varphi$ is a pre-condition of the plan step $s_j$ and effect of the plan step $s_i$. $s_i$ and $s_j$ are referred to as the producer and consumer plan steps of the literal $\varphi$ respectively. Generally, causal link constraints are formulated to preserve the literals that are accomplished so far. A planner can benefit from causal link constraints in two different ways: First, it can introduce an Establishment that handles an unsolved pre-condition (open pre-condition) $p_j$ of plan step $s_j$. This can be done by building a new causal link $c_j$ from a suitable plan step $s_i$ to produce the required pre-condition $p_j$ for plan step $s_j$ ($c_j : \langle s_i, p_j, s_j \rangle$), or by inserting a new plan step that carries the required pre-condition in its post-condition. Second, clobbering (Threat Removal) that removes the causal link threat. The causal link $c_j : \langle s_i, p_j, s_j \rangle$ is threatened by another plan step $s_k$ when it deletes the protected literal $p_j$ of the causal link $c_j$, and the plan step $s_k$ is ordered between the two plan steps of the causal link. The threat is solved by propagating the order constraint between the threat task (i.e., plan step $s_k$) and the causal link components, that means, the order constraint between the consumer plan step and the threat task ($s_j \prec s_k$) or between the threat task and the producer plan step ($s_k \prec s_i$) is propagated.

One of the earliest planners that lifted STRIPS “total order planners” and maintains a partial order
on plan steps was the NONLIN planner [74]. It derived from NOAH planner in order to complete the step towards plan space planning. It introduced the partial plan data structure which includes plan steps, ordering constraints on the plan steps, and parameter bindings constraints. In addition, it applied a general backtracking schema over the plan generation process to re-construct alternative ways in case a particular choice lead to a dead end. After that, Chapman [71] demonstrates in his planner TWEAK an alternative way, the so-called *model-truth-criterion*, for determining whether a partial plan can achieve a given pre-condition at a given step. However, it uses the *model-truth-criterion* in order to generate a plan by incrementally adding new plan steps and adding or modifying constraints. The produced plan will continue to achieve the next subgoals and so on. When all goals of the plan are achieved, the given planning problem is solved. TWEAK depends on the result of the model truth criterion in order to choose the appropriate plan step $s_j$. The result of model truth criterion is true if and only if: (1) the pre-condition $p_j$ of the current plan step $s_j$ has been achieved by another plan step $s_i$, (2) the plan step $s_i$ is performed before the plan step $s_j$ ($s_i \prec s_j$), and (3) there is no other plan step $s_k$ preceding the plan step $s_j$ that might remove pre-condition $p_j$.

The most popular partial order causal link planners \(^3\) are SNLP \(^4\) [77] and UCPOP \(^5\) [73]. SNLP is considered a further development of Tate’s planner (*NONLIN*) [74, 76]. The technical difference between them lies in its threat formulation. NONLIN considers the plan step $s_k$ a threat to a causal link $(s_i, \varphi, s_j)$ only if $s_k$ deletes the protected literal $\varphi$. SNLP, however, is more restricted. It considers $s_k$ to be a threat when it deletes or adds the protected literal $\varphi$. This restriction provides a more systematic algorithm, because it does not allow to duplicate a plan in the plan space.

The UCPOP planner discusses the foundations of partial order planning. It is the first non-linear

\(^3\)All planners that apply causal link paradigm called Partial Order Causal Link (POCL)

\(^4\)SNLP $\Rightarrow$ Systematic Non-Linear Planner

\(^5\)UCPOP $\Rightarrow$ Universal Conditional Partial Order Planner
planner for which soundness and completeness were implemented. In addition, UCPOP’s domain model is formulated in ADL \cite{78}. The theoretical foundation of total order planning with ADL was inspired by the works of Pednault \cite{79,80}. Afterwards this foundation was developed by Pednault \cite{78}, in order to handle the partial order plans. A lot of planners have been established to extend UCPOP. Most of them focused on implementing efficient planning strategies such as the VHPOP\footnote{VHPOP \Rightarrow Versatile Heuristic Partial Order Planner} system \cite{81}. It extends the capabilities of POCL planners by also considering durative actions (i.e., \textit{temporally extended actions}) \cite{82}. In addition, it competed well with other heuristic state space planners at the 3rd \textbf{I}nternational \textbf{P}lanning \textbf{C}ompetition (IPC3). The common problems for all planning algorithms that have been discussed are the complexity of the planning problems and they do not have high flexibility to express more actions and states.

### 3.1.3 Hierarchical Planning

The common approach to improve the efficiency of planning is to use \textit{Hierarchical planning}. In general, hierarchical planning is categorized into two approaches, based on the kind of abstraction: \textbf{state abstraction} and \textbf{action abstraction}.

In hierarchical planning, the state abstraction is a powerful method for reducing the planning search space from exponential to linear time under specific conditions such as a hierarchy that satisfies DRP\footnote{DRP \Rightarrow Downward Refinement Property}. The DRP condition guarantees that no backtracking occurs between abstraction levels. DRP is formalized in ABSTRIPS-style hierarchies \cite{83}.

The ABSTRIPS planner is the earliest system that deals with state abstraction. It is built on top of the STRIPS planning system. The abstraction hierarchy for the problem space in ABSTRIPS is constructed by assigning a number of so-called \textit{criticality values} to the pre-conditions of each plan.

---

\textsuperscript{6}ADL \Rightarrow \textbf{A}ction \textbf{D}escription \textbf{L}anguage

\textsuperscript{7}VHPOP \Rightarrow \textbf{V}ersatile \textbf{H}euristic \textbf{P}artial \textbf{O}rder \textbf{P}lanner

\textsuperscript{8}DRP \Rightarrow \textbf{D}ownward \textbf{R}efinement \textbf{P}roperty
operator. Through these critical values, ABSTRIPS generates a plan by starting with the highest critical value. Afterwards, this abstract plan is refined by considering now the pre-conditions that have the next critical value and so forth until the lowest level of criticality is reached.

One of the most important hierarchical planners that generates abstraction for solving hierarchical problems automatically and has better abstraction than ABSTRIPS is ALPIN [84]. It relies on the ordered monotonicity property which ensures that the structure of the abstract plan will be preserved while the plan is refined i.e. all the refinement plans of the abstract plan leave the literals that have already been achieved in the abstract space without any change.

Although those planners that provide hierarchical planning by state space abstraction reduce the search space successfully, they lack in semantics because they are defined as a search algorithm and not modeled as a planning domain model.

As opposed to this, the second category of hierarchical planning depends on the hierarchies of the abstraction of actions, it is also known as Hierarchical Task Networks (HTN)⁹. In general, action abstraction systems organize the description of the actions in a hierarchical form. Specifically, the more abstract or complex action is placed on the top of hierarchy (high or abstract level), and the more specific one is placed on the lower level in the hierarchy and so forth, until the lowest level is reached, which is called the primitive level. Therefore, there are two types of tasks: primitive tasks that can be performed directly like operators in the STRIPS paradigm, and abstract tasks that must be decomposed into smaller sub-tasks (either abstract or primitive) during the planning process.

All information about the task hierarchy and how to implement these tasks in the plan are organized in the domain model. Besides the set of tasks in the hierarchical planning domain model, it also includes a data structure so-called methods. Each abstract task may have more than one applicable method, so the relevant pre-conditions and effects are not always known in advance. Each method specifies a pre-defined abstract solution or implementation of the corresponding abstract task. It is

⁹HTN is common the approach of “planning by action abstraction”
important to notice that, the conflicts which are introduced during the decomposition process are resolved by adding temporal or variable constraints.

Opposite to the STRIPS planning system which defines a planning problem as an initial state, a goal state and a set of actions that achieve a given goal state, HTN planning defines a planning problem as an initial state and an initial plan. The initial plan is a non-empty plan containing a set of tasks (abstract or primitive tasks) that need to be performed. The solution plan is found by incrementally decomposing the abstract tasks in the initial plan until a primitive plan that has only primitive tasks has been reached, that is executable in the initial state, and that has consistent constraint sets. The decomposition process is called refinement or expansion process. The decomposition process works by replacing the abstract task by a set of less abstract tasks. Attention should be paid to the fact that, each task may have a set of alternative expansions (see Figure 3.2).

Figure 3.2 Planning by action abstraction: Alternative expansions of abstract task “Deliver concept for Broader student”

One of the most commonly known HTN planners is O-plan [85]. O-plan is a domain independent general planning framework. It follows NONLIN [72] in using a tightly constrained method to generate plans that compose the search space. Therefore, O-plan represents abstract actions by introducing condition types in the abstract expansion schemata instead of propagating

\[
\text{M}_1: \text{Angry} \quad \text{M}_2: \text{Happy} \quad \text{M}_3: \text{Fear}
\]
pre-conditions and post-conditions in the abstract task [86]. These condition types are similar to causal link constraints. They describe the relationships between different actions in the plan. These conditions cause difficulties for the design of O-plan domain models because the domain designer has to satisfy the all conditions on actions.

Wilkins et al. [87] introduced an HTN-system called SIPE 10. This system places a restriction on the possible ways to order actions \textit{i.e. for a given two sub-plans that are unordered with respect to each other, SIPE orders them by putting one sub-plan before or after the other.}

Afterwards, SIPE was further developed into a practical HTN planning system, called SIPE-2 [88]. To achieve the given goals in diverse problem domains, SIPE-2 provides a domain independent formalism for describing actions, and utilizes knowledge encoded in these actions. This is combined with a heuristic search that handles the combinatorics of the problem. In addition, SIPE-2 can reason about resources. It can post and use constraints as well as employ a deductive causal theory to represent and reason about different world states. SIPE-2 has been applied to a lot of domains such as military operation [89] and manufacturing environment [90].

Erol was the first to present a formal syntax and semantics for HTN planning [91]. He introduced a hierarchical planner, the \textit{UMCP planner} 11 which depends on the task decomposition [7]. The UMCP planner represents world state and primitive tasks in a similar way as the STRIPS formalism, whereas the goal is represented by the complex tasks. The set of tasks in the plan are connected by a task network which is used to represent plans and sub-plans. The task network is a tuple \((A, T, B)\) where \(A\) is a set of tasks, either primitive or complex, \(T\) is a set of temporal constraints on the members of \(A\), and \(B\) denotes a set of variable binding constraints. UMCP works by recursively expanding each complex task by non-deterministically choosing a method until a primitive plan is produced \textit{i.e. all tasks in the plan are primitive tasks}. Afterwards, it starts to solve conflicts between tasks and extract a solution plan.

\(^{10}\text{SIPE } \implies \text{ System for Interactive Planning and Execution}\)

\(^{11}\text{UMCP } \implies \text{ Universal Method Composition Planner}\)
Algorithm 1: The Basic HTN Planning Algorithm

Input: \( P = \langle D, N \rangle \): Planning Problem

Output: Solution Plan \( P \) or Failure.

begin
  if (All tasks in \( P \) are primitive tasks) then
    Resolve the conflicts between the primitive tasks in \( P \)
    return Solution Plan
  end if

  if (conflicts unresolvable) then
    return Failure
  end if

  else
    Select an abstract task \( t \) in \( P \).
    Select an expansion for task \( t \).
    Replace \( t \) with the expansion sub-tasks.
    Detect conflicts between tasks in \( P \).
    Suggest the possible ways to resolve the detected conflicts.
    Apply one of the above suggestions (in line 12) on \( P \).
    return BasicHTN(\( P \))
  end if
end

Algorithm 1 introduces the basic HTN planning procedure which is considered to be a core of all HTN planning systems. The input of the basic HTN planning algorithm is a planning problem \( P = \langle D, N \rangle \), where \( D \) is a domain model and \( N \) is a task network. It returns a solution plan if the plan \( P \) is a primitive plan and consistent (i.e., all tasks in the plan are primitive, and all the conflicts between them are resolved) (lines 2 to 4). If the primitive plan \( P \) is inconsistent (i.e., the conflicts between tasks cannot be resolved), then a failure is returned (lines 5 and 6). Otherwise, in lines 8 to 9, an abstract task \( t \) in \( P \) and the suitable decomposition method that matches the task
t are selected. After that, the selected decomposition method is applied to the current plan $P$ by replacing the specified abstract task with the set of sub-tasks in the selected method (line 10). Worth mentioning is that adding new tasks in the plan $P$ may produce conflicts between tasks. These conflicts are addressed in line 11. All the possible ways to handle these conflicts are computed in line 12. In general, there is more than one way to decompose an abstract task and more than one way to resolve conflicts (e.g., task interactions) in a plan. Then, in line 13, we select one of the suggested ways (in the previous line) in order to solve the conflicts in the plan $P$. Finally, in line 14, the basic HTN algorithm is called recursively with the refined plan (i.e., plan after expansion) in order to make a new expansion.

Nau et al. introduced not only one of the most important HTN system, but also proved that it is sound and complete [92]. Their planner is called SHOP\textsuperscript{12}. SHOP reduces the complexity of reasoning by generating a plan for tasks in the same order in which they will be performed. It enables to build a complete world state at each iteration of the planning process. This is done by applying a forward-chaining search algorithm that starts by selecting the first abstract task according to the step ordering. Then it chooses the suitable method that decomposes the selected task. Note that in the SHOP system, each method has a pre-condition that has to hold in the current state before applying the corresponding method, while primitive tasks do not have pre-conditions. SHOP follows the if-then-else paradigm to select a method. Consequently, the appropriate method is checked and the first method for which the respective if-statement evaluates to “true” is selected for expanding the task. The process of creating a domain model in SHOP requires a greater effort than what is required for classical planners, because the domain model of SHOP propagates total order among sub-tasks in the decomposable method. Of course this renders it impossible to interleave sub-tasks of methods. For this reason, SHOP has been improved to allow each method to decompose into partially ordered sub-tasks i.e. in a produced plan it is allowed to interleave sub-tasks from differ-

\textsuperscript{12}SHOP $\Longrightarrow$ Simple Hierarchical Ordered Planner
The crucial difference between classical planning and HTN planning is the solution criterion. Whereas the goal in classical planning is to achieve a desired property, no matter which actions have to be used to accomplish this, the goal in HTN planning is to find a plan that is a valid decomposition for the initial abstract task, such that the resulting plan only contains primitive tasks.

### 3.1.4 Hybrid planning

The difficulty of solving problems in complex real-world application domains, such as emergency evacuation, crisis management [6, 94], and transportation/logistics problems [95] led to the appearance of a new planning paradigm, so-called hybrid planning.

In general, hybrid planning is a combination of hierarchical task network with classical planning approaches, each having been studied separately. This means that the produced system has advantages of both approaches i.e. it has good modeling and efficient search techniques.

There are a very few works that discuss hybrid systems, one of them is DPOCL [96]. DPOCL built on top of the SNLP [77] algorithm to handle partial action decompositions. DPOCL represents each action by two parts: First, an action schema is a tuple \( \langle A, V, P, E, B \rangle \), where \( A \) is an action type (primitive or composite\(^{14}\)), \( V \) is a set of variables, \( P \) and \( E \) represent the pre-conditions and effects of the current action respectively, and \( B \) is the set of variable bindings for the variables in \( V \). Second, each action has a set of decomposition schemata which represent alternative ways to decompose the complex action into more primitive actions. Intuitively, the set of decomposition schemata of primitive tasks are empty, because they are performed directly without any decomposition. The process of generating a plan in DPOCL includes deciding what action should be used

---

\(^{13}\)DPOCL = Decomposable Partial Order Causal Link

\(^{14}\)Composite action is a complex or abstract task
3.1 AI Planning

to achieve a sub-goal as well as solving interactions between plan steps (tasks).

Kambhampati et al. [97] introduced another hybrid system integrating hierarchical task network planning and refinement planning. To this end, the refinement planning framework for classical planning is extended to include complex actions and the decomposition schemata of these actions as part of the domain specification [98]. In the process of plan generation, the plan is refined by selecting an open pre-condition from the current plan and then closing it by selecting an appropriate task that generates this pre-condition. If the selected task does not close the current condition explicitly then this condition is handled again via a *phantom establishment* process which converts this open pre-condition into a constraint that solved later during plan development.

Recently, Schattenberg et al. [6,99] introduced a new hybrid planning system, the so-called PANDA system\(^\text{15}\). As mentioned before, POCL planning is a technique used to solve partial order planning problems. The objective is to achieve the goal state by applying actions in a correct order starting in a given initial state. In addition, the POCL technique explicitly shows causal dependencies between actions. The key feature of HTN planning is action abstraction which allows representing abstract tasks as well as pre-defined abstract solutions for these tasks. So, HTN planning reflects and employs abstraction hierarchies that are inherited in many domains. Therefore, the PANDA environment combines the key features of POCL and HTN planning techniques. In contrast to the previously discussed techniques, PANDA requires pre- and post-conditions for abstract tasks. It utilizes a mapping between abstract tasks and the alternative ways of implementing these abstract tasks by way of so-called *Decomposition methods*. PANDA defines a planning problem through a domain model as well as an initial partial order plan, and a tuple of initial state and goal state. The initial plan contains the set of the most abstract tasks that are required as well as two artificial tasks \(t_{init}\) and \(t_{goal}\) that represent the initial and the goal states respectively. The task \(t_{init}\) considers the initial state as post-condition and the task \(t_{goal}\) considers the goal state as pre-

\(^{15}\text{PANDA} \implies \text{Planning and Acting in a Network Decomposition Architecture}\)
condition. The initial partial plan is stepwise refined by adding new tasks and constraints - causal links, ordering and variable constraints. One of the most important features of the PANDA system is that it is implemented in separate modules. So that the implemented system can be employed as a platform to implement and evaluate various planning methods such as purely HTN planning as well as evaluating a lot of different search strategies [100, 101].

In contrast to other systems, which implicitly define their control search strategy by their search procedure, the PANDA planning environment explicitly defines the search strategy. A search strategy in the PANDA planning environment is a combination of the used modification and plan selection functions. Let us take a look at a simple example strategy for clarification: To perform a depth first strategy, the plan selection strategy has to be the identity (i.e., $f^{PlanSel}(\overline{P}) = \overline{P}$ for any plan sequence $\overline{P}$), whereas the modification selection strategy $f^{ModSel}$ can be arbitrarily chosen (but decides, which branches to visit first). Thus, the plan selection strategy is used to prioritize the plans; several strategies can be concatenated for tie-braking. The plan selection strategy uses its input sequence for tie-braking as well: If two plans are still invariant after applying the plan selection function, the order given in the input is used. A number of different plan and modification selection strategies have been described and evaluated in the work of Schattenberg [99–101]. Therefore, due to the great properties of the PANDA planning environment, our planning framework will be an adaptation of the hybrid formalization of PANDA.

### 3.2 Formal Framework

The planning part in our approach relies on a hybrid planning formalization [6], which combines HTN planning with concepts of partial-order-causal-link (POCL) planning. The resulting systems integrate task decomposition with explicit causal reasoning. Therefore, they are not only able to use pre-defined standard solutions as it is the case in pure HTN planning, but also to develop (parts
of) a plan from scratch or to modify a default solution in cases where the initial state deviates from the presumed standard. It is this flexibility that makes hybrid planning particularly well suited for real-world applications [94, 102].

In this section we will cover the underlying logical language of our framework, tasks, domain model entities, plans, planning problem and solutions as well as illustrate how to refine the current plan in order to generate the final solution plan.

### 3.2.1 Logical language

Our framework is adapted from the formalization of PANDA hybrid planning [6]. It relies on an sorted first-order logic [103]. The syntax of our framework is given by the logical language \( L = \langle Z, \prec, R, Const, V, O, T, E, L \rangle \).

In sorted logics, all variables and constants are of some sort \( z \in Z \). In addition, order-sorted logics impose a hierarchy on sorts which allows for more adequate and concise formalizations. This hierarchy on the sort symbols in \( Z \) is represented by defining the relation \( \prec \). Sorts in \( Z \) are super or sub sorts for each other.

The relation symbols \( R \) are used to represent properties of objects in the real world. The relation \( R \) is a \( Z^* \)–indexed family of finite disjoint sets of relation symbols. Each relation \( R \) can be either rigid or flexible. In general, the term rigid is applied to those literals or relations that cannot be modified or added during the planning process. While flexible terms are those literals that can be added or modified during the planning process. \( Const \) is \( Z \)-indexed family of finite disjoint sets of constant symbols which represent objects in the real world. The \( O \) and \( T \) represent finite disjoint sets of operator and task symbols respectively. The symbol \( E \) denotes a \( Z^* \)–indexed family, the so-called elementary operation symbols. They provide for each flexible symbol \( R \) a so-called add operation \( (+R) \) and a delete operation \( (−R) \). Finally, \( L \) is an infinite set of labels used for identifying different occurrences of identical tasks.
3.2.2 Tasks

In artificial intelligence planning, changes in the real world are represented by actions or tasks. In our framework, a task schema \( t(\bar{\tau}) \) is specified by a tuple \( \langle \text{type}, \text{Prec}(t(\bar{\tau})), \text{eff}(t(\bar{\tau})) \rangle \).

**Definition 2 (Task).** For a given logical language \( \mathbb{L} \), a task schema is defined by a structure
\[
t(\bar{\tau}) = \langle \text{type}, \text{Prec}(t(\bar{\tau})), \text{eff}(t(\bar{\tau})) \rangle
\]
where

- \( t \) is a task symbol,
- \( \text{type} \) is a kind of task: abstract task or primitive task.
- \( \bar{\tau} = \tau_1, \tau_2, \cdots, \tau_n \) are the list of variables which belong to the set of variables \( V \). They are called task parameters.
- \( \text{Prec}(t(\bar{\tau})) \) specifies the pre-conditions of the task \( t(\bar{\tau}) \).
- \( \text{eff}(t(\bar{\tau})) \) specifies the post-conditions of the task \( t(\bar{\tau}) \). It consists of two parts: \( \text{eff}^+(t(\bar{\tau})) \) which adds new relations or properties to the current state, the so-called positive effects, and \( \text{eff}^-(t(\bar{\tau})) \) which removes existing relations or properties from the current state, the so-called negative effects. Attention should be paid to the fact that the set of pre-conditions and post-conditions are sets of literals over the relation \( R \) in the logical language \( \mathbb{L} \).

In our approach, both primitive and abstract tasks show pre-conditions and effects.

A state \( s \) is a finite set of ground atoms\(^{16}\) in \( \mathbb{L} \). A state tells us which ground atoms are currently true: if \( \alpha \) is a ground atom, then \( \alpha \) is true in the state \( s \) if and only if \( \alpha \in s \). Therefore, a task \( t(\bar{\tau}) \) is called applicable in a state \( s \), if the literals\(^{17}\) of its pre-conditions \( \text{Prec}(t(\bar{\tau})) \) are present in the state \( s \). If a task \( t(\bar{\tau}) \) is applicable in a state \( s \), its application leads to the new state \( s' \).

---

\(^{16}\)An atom is a predicate symbol followed by a list of terms.

\(^{17}\)A literal is either an atom (in which case we say the literal is positive), or the negation of an atom (in which case we say the literal is negative)
\[ s' = (s \cup \text{\textit{eff}}(t(\bar{\tau}))/\text{\textit{eff}}(t(\bar{\tau}))) \]

**Figure 3.3** Implementations of tasks (primitive and abstract)

It is important to note that an instance of the task schema is a copy of the schema where all task parameters are substituted by new variables through a well sorted variable replacement. The semantics of abstract tasks are based on a sequence of tasks which are provided by the available primitive task schemata. As depicted in figure 3.3(a) and (b), a primitive task is performed directly, while an abstract task requires further decomposition to be executed.

### 3.2.3 Domain Model

As has been previously described, the term *domain model* refers to all the knowledge regarding the real world application area. This knowledge is necessary for generating a solution plan for the given planning problem.

**Definition 3 (Domain Model).** For a given logical language \( \mathbb{L} \), a domain model is defined as a tuple \( D = \langle Q, M, T \rangle \).

\( Q \) represents the set of all objects and relations which exist in the application domain, \( M \) is a model structure that represents decomposition methods and \( T \) represents a set of abstract and
primitive task schemata. As we said, abstract tasks do not correspond to a single primitive task in the real world and are thereby not directly executable by human users. Instead, abstract tasks can be seen as constraints for the plans that require and achieve pre-conditions and post-conditions of these abstract tasks and can thus be regarded as pre-defined standard solutions.

A decomposition method \( m = \langle t(\bar{\tau}), LVC, p \rangle \in M \) relates an abstract task \( t(\bar{\tau}) \) to a partial plan \( p \) that implements an abstract solution for a task \( t(\bar{\tau}) \). Additionally, each method includes a set of local variable constraints \( LVC \) to map variables in the abstract task \( t(\bar{\tau}) \) to variables in the task network i.e. set of tasks in a partial plan \( p \). The set of decomposition methods covers all possible state-refinements of the corresponding abstract task. Therefore, each abstract task has a number of different methods that can be used for its implementation.

Finally, a domain model \( D \) constitutes the terminology, concepts, and the relationships between objects for the corresponding course of actions. Now we are ready to introduce the notion of a plan in the next section.

### 3.2.4 Plans

In classical planning, a plan is a sequence of actions with completely ordered plan steps. The plan in our approach, however, is a partial order. A partial order plan points to plan with partially ordered plan steps.

**Definition 4 (Plan).** For a given logical language \( \mathbb{L} \), a domain model \( D \), a partial plan \( P \) is defined by the tuple \( P = \langle TE, \prec, VC, CL \rangle \)

where,

- \( TE \): is a finite set of plan steps or task expressions \( te = \ell : t(\bar{\tau}) \), where \( t \) is a partially grounded abstract or primitive task. \( \ell \in L \) is a unique label in order to distinguish different occurrences of the same task within the same plan. Intuitively, the list of parameter variables
\( \bar{\tau} = \tau_1, \tau_2, \ldots, \tau_n \) are assumed to be unique in \( TE \).

- \( \prec \): is a finite set of explicit ordering constraints on the plan steps \( TE \). They take the form \( te_i \prec te_j \) with task expressions \( te_i, te_j \in TE \). The ordering constraints specify that the task \( te_i \) must finish before the beginning of the task \( te_j \). Note that the set of ordering constraints in \( \prec \) are produced as a result of the planning process or pre-defined by the domain model.

- \( VC \): is a finite set of variable constraints. They are a set of co-designations or non-co-designations used for grounding tasks and to force equality or inequality between variables. More formally, for two tasks \( t \) and \( \bar{t} \), \( \tau_i(t) = \tau_j(\bar{t}) \) constraints \( \tau_i(t) \) and \( \tau_j(\bar{t}) \) to be identical and, for co-designating variables with constants, \( \tau_i(t) = c \) constraints the variable \( \tau_i(t) \) to be equal the constant \( c \in C_z \), where \( z \in Z \) is the sort of \( c \). Non-co-designations are defined similarly.

- \( CL \): is a finite set of causal link constraints. They have the form \( te_i \xrightarrow{\varphi} te_j \) or \( \langle te_i, \varphi, te_j \rangle \), indicating that the task expression \( te_i = \ell_i : t_i(\bar{\tau}_i) \) establishes a pre-condition \( \varphi \) of a task expression \( te_j = \ell_j : t_j(\bar{\tau}_j) \), where \( \varphi \in \text{eff}(te_i) \land \text{prec}(te_j) \).

Although causal link constraints impose a partial order between plan steps, our plan identifies explicit ordering constraints between plan steps if causal threats need to be resolved (by promotion or demotion) or if they are already present in a predefined plan of the domain model.

### 3.2.5 Planning Problems and Solutions

A planning problem in HTN planning is formulated over a domain model \( D \) and consists of the initial world state description \( s_{init} \) which is a set of positive atoms that represent what conditions are true initially, and the initial plan \( p_{init} \). Formally we define:
Definition 5 (Planning Problem). For a given logical language \( L \) and domain model \( D \), a planning problem \( \Pi \) is defined by the tuple \( \Pi = \langle D, s_{\text{init}}, p_{\text{init}} \rangle \), where, \( s_{\text{init}} \) is an initial state description and \( p_{\text{init}} \) is an initial partial plan.

The initial plan \( p_{\text{init}} = \langle TE_{\text{init}}, \prec_{\text{init}}, VC_{\text{init}}, CL_{\text{init}} \rangle \) is a consistent partial plan. Where, The task network \( TE_{\text{init}} \) contains two artificial tasks \( te_{\text{init}} \) and \( te_{\text{goal}} \) which are used to provide the initial and goal state, respectively. The facts in \( s_{\text{init}} \) are used as effects of the task \( te_{\text{init}} \) while task \( te_{\text{goal}} \) has the desired goal state as a pre-condition. All other tasks get ordered in between. The initial plan constrains are represented by \( \prec_{\text{init}}, VC_{\text{init}}, CL_{\text{init}} \). It is important to notice that our planner assumes any atom which is not in the initial state \( s_{\text{init}} \) to be false.

A plan \( p_{\text{sol}} = \langle TE_{\text{sol}}, \prec_{\text{sol}}, VC_{\text{sol}}, CL_{\text{sol}} \rangle \) is a solution to the planning problem \( \Pi = \langle D, s_{\text{init}}, p_{\text{init}} \rangle \) if the following solution criteria are met:

1. A plan \( p_{\text{sol}} \) is a refinement of \( p_{\text{init}} \). Informally, we call a plan \( p \) a refinement of \( p_{\text{init}} \) if the plan \( p \) results from applying plan modifications to the plan \( p_{\text{init}} \). A plan modification is the insertion of a plan element, i.e., an element from the set of task expressions, temporal orderings, variable constraints and causal links. The only modification that is not a pure insertion is the application of a method: it replaces an abstract task by implementing a task network and adapts the variable constraints and causal links. The formal description of the modification is discussed in details in section 3.2.7.

2. All plan steps in the task networks of a plan \( p_{\text{sol}} \) are primitive tasks.

3. All pre-conditions of every plan step in the task network of a plan \( p_{\text{sol}} \) are supported by a causal link, i.e., for each pre-condition \( \varphi \) of a plan step \( te_j \in TE_{\text{sol}} \) there exists a causal link \( \langle te_i, \varphi, te_j \rangle \in CL_{\text{sol}} \) with \( te_i \in TE_{\text{sol}} \).

4. none of the causal links in \( CL_{\text{sol}} \) is threatened, i.e., for each causal link \( \langle te_i, \varphi, te_j \rangle \in CL_{\text{sol}} \) the ordering constraints in \( \prec_{\text{sol}} \) ensure that no plan step \( te_k \) with an effect that implies \( \neg \varphi \)
3.2 Formal Framework

...can be consistently placed between plan steps $te_i$ and $te_j$.

5. The ordering and variable constraints in a plan $p_{sol}$ are consistent, i.e., there is no plan step $te \in TE_{sol}$, such that $te \in^*_{CL} te$ ($\prec$ does not induce cycles on plan steps $TE$) and no $v \in V$ for $z \in Z$, such that $VC \models v \neq v$.

6. All tasks in a plan $p_{sol}$ are grounded. That is, all variables are co-designated to some constant.

In our approach, any solution must be a decomposition of the initial plan $p_{init}$. Since abstract tasks are regarded as non-executable, criterion 2 ensures that only executable tasks, *i.e.*, primitive tasks, are part of a solution plan. Criterion 3 ensures the applicability of tasks in a plan: in order for a task to be applicable in a state $s$, all its literals of its pre-condition must hold in $s$. This can be ensured by establishing appropriate causal links. Criterion 4 guarantees that every plan step in all linearizations of a plan is applicable in the sense of criterion 3: causal threats can cause a literal of the pre-condition of a plan step to be false in some linearizations although it is supported by a causal link. Since we require every linearization of a plan to be a valid solution, causal threats have to be eliminated. Criterion 5 is obviously necessary for constituting meaningful plans since a task can neither be ordered before itself nor can a constant be different from itself. Criterion 6 maps the variables used by tasks onto the objects available in the modeled world. In the next section we will discuss how to generate a solution plan $p_{sol}$ from the initial plan $p_{init}$.

3.2.6 Plan Generation

The process of generating a plan is adapted from the work of Schattenberg [99]. Plan generation means a stepwise refinement of the initial plan $p_{init}$ into a partial plan $p$ until the solution plan $p_{sol}$ is reached. Before presenting our refinement algorithm in more detail, we will illustrate the main requirements of our refinement algorithm: flaws and plan modifications.
Chapter 3 Planning

Flaws

A flaw is a data structure that refers to all plan elements which violate the solution criteria.

**Definition 6 (Flaw).** For a given planning problem specification $\Pi$ and a partial plan $P = \langle TE, \prec, VC, CL \rangle$ that is not a solution to $\Pi$, a flaw $f = \{f_1, f_2, \cdots, f_n\}$ consists of a set of critical plan components $f_i$ of $P$. It represents a defect in which these components are involved.

Let $F$ be the set of all flaws. Its subsets $F_x$ represent classes\(^{18}\) of flaws for a partial plan $P$. Then, a flaw class is the set of all possible flaws of a specific type in the current plan such as flaw class of abstract tasks $f_{AbstractTask}$ which refers to the abstract tasks in the current plan like Load task in $\Pi_{load}$ and $f_{OpenPreCondition}$ which points to those tasks that are not fully supported i.e., tasks that still have pre-condition not achieved by another task. The set of all flaw classes is computed through a detection module $f^{det}_x$. A detection module $f^{det}_x$ is a function that, given a partial plan $P$, a domain model $D$, and a planning problem specification $\Pi$, returns all flaws of type $x$ that are present in the current plan.

Plan Modification

Each refinement in our approach focuses on a single plan defect (flaw) and generates a new plan for each way of resolving that flaw. However, refinement steps include the decomposition of abstract tasks by selecting the appropriate methods, the insertion of causal links to support open pre-conditions of plan steps as well as the insertion of ordering and variable constraints. We call such a refinement step a plan modification.

**Definition 7 (Plan Modifications).** For a given partial plan $P = \langle TE, \prec, VC, CL \rangle$ and a domain model $D$ a plan modification is defined as the structure $\omega = (E^{\oplus}, E^{\ominus})$, where $E^{\oplus}$ and $E^{\ominus}$ are

---

\(^{18}\)The complete definitions of various flaw classes can be found in [100].
disjoint sets of elementary additions and deletions of plan elements, respectively, over the partial plan $P$ and domain model $D$.

All elements in $E^\oplus$ are elements in plan $P$ ($E^\oplus \subseteq P$) such as plan steps in $TE$, or constraints elements in constraints $\prec, VC$ and $CL$, while $E^\oplus$ consists of new elements such as new plan steps or new constraints ($E^\oplus \cap P = \emptyset$). This generic definition makes all changes a modification imposes on a plan explicit. Applying a modification $\omega = (E^\oplus, E^\ominus)$ to a current plan $P$ returns a new plan $P'$ that is obtained from $P$ by adding all elements in $E^\oplus$ and removing those of $E^\ominus$.

The set of all modifications is denoted by $\Omega$. Its subsets $\Omega_y$ represent modification classes. The set of all modification classes is computed through a modification module $f^{ModGen}$. Each plan modification class $\Omega_y$ can be used to handle a specific flaw class $F_x$. Therefore, the plan modification class $\Omega_y \in \Omega$ is suitable to solve the flaw class $F_x \in F$ iff there exists a partial plan $p$ which contains a flaw $f \in F_x$ and a plan modification $\omega \in \Omega_y$, such that the new plan $P'$, which is produced by applying a plan modification $\omega$ on a plan $P$, does not contain flaws $f$ anymore. A plan modification might introduce new flaws. Therefore, in order to separate the computation of flaws from the computation of modifications, a modification triggering function $\alpha$ is introduced to relate each flaw class to those modification classes that are suitable for generating refinements that solve the respective flaw such as $\alpha(F_{AbstractTask}) = \Omega_{ExpandTask}$ (See Table 3.1). Note that the inconsistent ordering flaw can not be resolved by adding constraints. Therefore, it does not have a modification function to solve it.

### 3.2.7 Search Strategy

As we mentioned later, the detection module is used to detect the set of flaws in the current plan, while the refinement or modification module is used to generate refinement alternatives. Then a search strategy used to decide which paths in the refinement space to pursue. In general, our refinement planning divides the search strategy into two processes: the process of selecting the
Table 3.1 This table lists the different kinds of flaw classes and the corresponding ways to solve those flaws

<table>
<thead>
<tr>
<th>Flaw Class $F_x$</th>
<th>Modification class $\Omega_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{AbstractTask}$</td>
<td>$\Omega_{ExpandTask}$</td>
</tr>
<tr>
<td>$F_{OpenPrecondition}$</td>
<td>$\Omega_{AddCausalLink}, \Omega_{ExpandTask}, \Omega_{InsertTask}$</td>
</tr>
<tr>
<td>$F_{CausalThreat}$</td>
<td>$\Omega_{ExpandTask}, \Omega_{AddOrdering}, \Omega_{BindVariable}$</td>
</tr>
<tr>
<td>$F_{InconsistentOrdering}$</td>
<td>$-$</td>
</tr>
<tr>
<td>$F_{UnboundedVariable}$</td>
<td>$\Omega_{BindVariable}$</td>
</tr>
</tbody>
</table>

refinement options, the so-called modification selection strategy and the process of selecting the path that is to be followed in the Induced search space (Definition 8) which is generated by refinement options, the so-called plan selection strategy. The modification and plan selection strategies impose a partial order on the respective input refinement options and these strategies are therefore suited for a sequenced arrangement.

A basic form of a modification selection strategy is either to prefer specific classes of plan modifications, e.g., we prefer the expansion of tasks or we try to delay an assignment of variables to constants as long as possible.

Refinement Algorithm

Before we present our refinement planning algorithm in more detail, we define the search space induced by the HTN planning problem $\Pi$.

Definition 8 (Induced Search Space). Let $\mathcal{P}_\Pi = (\mathcal{V}, \mathcal{E})$ be the directed acyclic graph which represents the (possibly infinite) search space induced by a planning problem $\Pi$. Then, the set of
vertices $\mathcal{V}$ is the set of plans in the search space and the set of edges $\mathcal{E}$ corresponds to the set of used plan modifications. By abuse of notation, we write $P \in \mathcal{P}_\Pi$ to state $P \in \mathcal{V}$. The root of $\mathcal{P}_\Pi$ is the initial plan of $\Pi$. The direct successors of a plan $P \in \mathcal{P}_\Pi$ are all plans $P'$, such that $P'$ resulted from $P$ by applying a plan modification $\omega$ to $P$. Then, $\omega \in \mathcal{E}$.

With the defined flaw and modification detection modules as well as a search strategy, we can now illustrate our refinement planning algorithm (Algorithm 2) which takes the initial plan $p_{init}$ of the planning problem $\Pi$ as an input and refines it stepwise until a solution plan $p_{sol}$ is found. Our algorithm performs an informed search, guided by search strategies in the search space induced by the HTN planning problem $\Pi$.

**Algorithm 2: Refinement Planning Algorithm**

<table>
<thead>
<tr>
<th>Input</th>
<th>The sequence Fringe $= \langle p_{init} \rangle$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Solution Plan $(p_{sol})$ or fail.</td>
</tr>
</tbody>
</table>

1. while Fringe $= \langle p_1 \ldots p_n \rangle \neq \varepsilon$ do
   2. $F \leftarrow f^{FlawDet}(p_1)$
   3. if $F = \emptyset$ then return $p_1$
   4. $\langle \omega_1 \ldots \omega_n' \rangle \leftarrow f^{ModOrd}(\bigcup_{f \in F} f^{ModGen}(f))$
   5. $\text{succ} \leftarrow \langle \text{apply}(\omega_1, p_1) \ldots \text{apply}(\omega_n', p_1) \rangle$
   6. Fringe $\leftarrow f^{PlanOrd}(\text{succ} \circ \langle p_2 \ldots p_n \rangle)$
   7. return fail

The fringe in the algorithm is a sequence of plans $\langle p_1 \ldots p_n \rangle$ ordered by the deployed search strategy. It contains all unexplored plans that are direct successors of visited non-solution plans in $\mathcal{P}$. According to the used search strategy, a plan $p_i$ leads more quickly to a solution than plans $p_j$ for $j > i$. The current plan under consideration is always the first plan of the fringe. The planning algorithm loops as long as no solution is found and there are still plans to refine.
(line 1). Hence, the flaw detection function \( f^{\text{FlawDet}} \) in line 2 calculates all flaws of the current plan. A flaw is a plan component that is involved in the violation of a solution criterion. In HTN planning, the presence of an abstract task raises a flaw that includes that task, a causal threat consists of the causal link and the threatening plan step, and so on. If no flaws can be found, the plan is a solution and returned (line 3). In line 4, all plan modifications are calculated by the modification generating function \( f^{\text{ModGen}} \), which addresses all published flaws. Afterwards, the modification ordering function \( f^{\text{ModOrd}} \) orders these modifications according to a given strategy. The fringe is finally updated in two steps: First, the plans resulting from applying the modifications are calculated (line 5) and are put in front of the fringe in line 6. Second, the plan ordering function \( f^{\text{PlanOrd}} \) orders the updated fringe according to its strategy. This step can also be used in order to discard plans (i.e., to delete plans permanently from the fringe). This is useful for plans that contain unresolvable flaws like an inconsistent ordering of tasks. If the fringe becomes empty, no solution exists and \text{fail} is returned.
Chapter 4

System Architecture

Emotional intelligence is considered as an important factor when it comes to improving our performance. Psychology and neuroscience research indicates that emotions have a powerful influence on human behavior. Moreover, people with different personality cope with a problem in different ways. In this context personalized education systems aim to give the student an individual course lesson. The process of arranging personalized adaptations is usually complex, and the current platforms usually do not provide more than a relatively simple way of personalization and adaptation. This chapter describes the general architecture of PAND-TUTOR, a model of an emotional intelligent tutoring system enhancing the emotional intelligence of the student as well as the intellectual intelligence. In addition, we will illustrate how the planning system and the ontology can cooperate to support the modeling of the system.

4.1 Affective Computing

Human emotions are essential for thinking and the learning process, for virtual platforms and learning environments. Therefore, it is important to improve the quality of communication and transaction intelligence between human and computer. Thus, emotions must be taken into account
when implementing information systems in order to enable a more social and humanistic computing for our knowledge society.

So, a novel and an important topic in the field of human computer interaction is introduced by Picard [104], which was described in the monograph ”Affective Computing”. The term affective computing involves the intention of Artificial Intelligence researches to model and incorporate emotions in intelligent systems. The aim of affective computing is to build an intelligent computing system which can recognize and understand human emotions, and has the ability to respond intelligently, sensitively, and friendly, like a human. This concept combines the perspectives of many disciplines, such as neuroscience, psychology, and philosophy.

Modeling emotion plays an important role for emotional computing because of the necessity of making human computer interaction more natural. Thus, human prefers communicating with the computer by a natural way [105]. Software applications which include a model of the emotional process need to model the emotion and communication theory of human machine interaction. The AI directions in recent years aim to simulate human emotion.

This approach has a broad and multifaceted subject in the field of artificial intelligence. Thus, Affective Computing is a new artificial intelligence area that deals with the possibility of making computers able to recognize emotions in different ways.

On the other side, teaching and learning are emotional processes: the student can learn more from a teacher who communicates with the student in an emotional way. A human tutor or teacher is in fact an emotional practitioner in the sense that they can influence the students’ emotions with the objectives of improving their learning efficiency. It is a fact that, combining and dealing emotionally with the student can enhance the student’s involvement in the learning environment and increase the efficiency of the cognitive process. In the learning process, the emotional state of the student is the responsibility of parents or a teacher rather than computers. It is important to consider the emotional side in computer interaction as well. Otherwise, it will be a cold technology without
any emotions and will never be a successful learning process. Accordingly, the new generation of ITS should be able to influence the student’s emotions in the same way as a human tutor. Thus, ITSs need to regard the concept while selecting the way of teaching, the lessons they deliver, the relationships they develop with students, and the results that students ultimately achieve.

The affective educational systems should consider the student state and the tutorial situation to establish the affective action. The affective action helps the tutor establishing the emotional realization based on the student module. This new paradigm works by detecting and responding to the student’s emotions. The affective tutoring systems would be significantly enhanced if the system could be adapted according to the student’s emotions. In other words, the student can receive the tutorial actions with an affective and pedagogical component, which is our main contribution in this research. Some educational systems have given attention to the generation of emotion during the learning process [106] or the recognition of the student’s emotion during the learning process [107]. However, different aspects of the student’s emotions have not received adequate attention and the exciting systems are not interested in improving the emotional intelligence of the student. In this way our proposed system improves learning process by considering a more personalized environment through recognizing the student’s affective state with the aim of reacting appropriately from pedagogical and affective point of view.

It is important to mention the relation between affective computing and emotional intelligence. According to Picard,” if we want computers to be more genuinely intelligent and to interact naturally with us, we must give computers the ability to recognize, understand, and even to have an expresses emotion”. Thus, it is not enough to consider the human emotion during human computer interaction as the affective computer approach, but it is also important to consider the emotional intelligence. We will explain the importance of considering emotional intelligence in intelligent tutoring system in the next section.
4.2 Emotional Intelligence

The term ‘intelligence’ is defined as ”the ability to learn or understand from experience or to respond successfully to new experiences” or ”the ability to acquire and retain knowledge” (Webster’s New World Dictionary). That means, to be intelligent, to have the ability to solve problems and direct conduct.

Cognitive Intelligence means intellectual abilities such as logic, reason, ability to read, to writ, to analyze and prioritize. These go on in your own head and utilize only the neocortex, not the emotional centers of the brain which also provide crucial information. These abilities do not require any social skills, i.e., you can solve a math equation by yourself, or write an essay, or balance your business books by yourself.

The mind, according to Minsky [108], is a collection of structures that can both cooperate with and contradict each other to find ways to deal with conflicting goals. He investigates various causes of the human’s ability: representation of knowledge as objects with various descriptions and interconnection, emotional state as an important factor for thinking, and the cognitive state of human. Thus, during the learning process the student should accomplish a variety of tasks such as understanding, memorizing, analyzing, reasoning, or applying. Students need to develop the way of solving problems and to reduce the redundant factors to be able to solve them, as well as, changing the approach in case of failure. The student’s performance in these learning tasks depends on his/her emotional state. In addition, Emotions play an important role in the learning process. Positive emotions can enhance learning enthusiasm, but negative emotions can suppress learning interest. For example, the student who feels generally happy performs better than students who feel sad, angry or scared.

Emotional intelligence (EI) is a different type of intelligence. It can be defined as the ability to think constructively and act wisely, as well as the ability to recognize, interpret, and influence someone’s emotions. It’s about being ”heart smart,” not just ”book smart”. Thus, the integration of
4.2 Emotional Intelligence

Figure 4.1 The basics of emotional intelligence

emotions with cognitive processes facilitate and enhance reasoning [109]. Because of the involvement of emotional (right brain) with the cognitive (left brain) can dramatically improve learning. It is worth to mention that emotional intelligence and cognitive intelligence should be positively associated because they are both subsumed under general intelligence. The human behavior is a result of the integration of the cognitive mind and the emotional mind. For that reason, it is natural to include the emotional intelligence into the learning process. Emotional intelligence is considered as an important factor in improving performance at work, at school, or at home. It is considered as an important factor for the student’s performance. On the other hand most students, no matter how intelligent a tutor or a companion is, will eventually become annoyed if the tutor does not have emotional intelligence. The student’s performance depends on the external states that exist when learning occurs. Most of the researches’ results indicate that the cognitive process is widely related to emotions and it plays an important role in cognitive process (such as attention, long term memorizing, problem solving, decision-making, etc.). In addition, performance, creativity and flexibility regarding learning and solving the problem is improved [110].

The student’s learning depends upon cognitive and emotional intelligent. However, intellectual
intelligence (IQ) is usually less important in determining how successful we are than emotional intelligence (EQ). Some people who are academically ingenious are socially bungling and unsuccessful. What they are missing is emotional intelligence. In particular, being emotionally intelligent means to act in a way that takes into account someone’s emotions in order to improve his cognitive abilities. The research shows that IQ contributes only about 20% to success in life. The rest of 80% depends on our EQ. However, we train IQ but rarely, if ever, train EQ in a structured way. So, we can develop our EI by upgrading our emotional skills. Some researchers suggest that emotional intelligence can be learned and strengthened. Thus, intelligent tutoring systems should integrate emotional intelligence for improving the student’s performance.

A new approach in the field of emotion and instruction which tried to overcome the mentioned problems of affective education is "emotional education" based on the concept of emotional intelligence [1]. Within this approach, affective education was criticized because it used emotions for achieving certain socially accepted learning objectives. Emotional education used instructions to achieve positive emotions, the ability to monitor emotions in relation to oneself and others, the ability to label and interpret emotions, or the ability to express emotions accurately. Within this approach emotions were the Content of instruction. Emotional education considers different emotions and emotional problems of learners and also has preventive bad effects. So, students need additional courses that reduced the traditional course offers.

The learning environments need to consider all of these factors and the emotional state of the student. Although several approaches have been constructed for ITS, enhancing the students emotional intelligence has not been considered so far. In this thesis we will introduce a new approach, the PANDA.TUTOR to achieve this goal. It is an emotional intelligent tutoring system hat has the ability to enhance the student’s emotional intelligence as well as intellectual intelligence. It considers the students personality type during the learning process. Our approach is actually realized with the help of AI Planning techniques.
4.3 System Architecture

In this thesis, we will introduce PANDA.TUTOR, a new approach for authoring/course generation using ontologies and an HTN planning technique called PANDA.TUTOR. This system allows the author to prepare the course structure and content, enriched with classification information without the need to define the adaptation rules or to specify configurations for each student, aiming to keep the amount of time and effort that has to be expanded during the authoring process low. The course generation system, which is basically an HTN planning system, introduces a personality based planning approach, in which the course structure and the course content are generated and adapted according to the student’s personality and state, as well as helping the student to adjust or maintain his/her emotion. PANDA.TUTOR includes different learning strategies, learning styles, and regulation strategies which help the planner to construct highly individualized courses and help students to build up their knowledge. The student’s personality type is taken as a guide to select an appropriate scenario, teaching strategy, and learning style in order to regulate the current emotional and motivational state of student. In addition, a dialog in PANDA.TUTOR is presented to improve the emotional intelligence of the student and grasp the students appraisal to build up an individual interpretation of how external events relate to the student’s goal and desires as well as overcoming has emotional reasoning.

As depicted graphically in Figure 4.2, our architecture consists of four modules: The course module, the student module, the pedagogical module and the interface. An authoring phase is defined through the course module and the course generation phase is defined through the pedagogical module. We have developed general educational ontologies for the student module and course module that can be used for different domains. Ontologies help us to share, reuse and reason about information. During the pedagogical module two main processes are defined: The dialog process and the course generation process based on the planning technique that generates the course, regarding the course module and student module. The generated plan will be a set of
querying statements that query the course ontology in order to retrieve learning objects to generate the course for students. PANDA.TUTOR includes four types of scenarios: A lesson scenario, revision scenario, companion games scenario, and a test scenario. The term scenario means a discrete sequence of steps inside the learning process. In these scenarios the emotional and motivational state are into account for generating the course for the concept goal.

Figure 4.2 The system architecture

So, we developed general educational ontologies for the student module and course module that can be used for different domains. Ontologies help us to share, reuse, and reason about information. The pedagogical module based on the planning technique generates the course regarding the course module and student module.
4.3 System Architecture

4.3.1 The Goal of the Proposed Architecture

Generally Speaking, the proposed architecture has the following features:

- **Generic and flexible course model** The course model offers a general framework for representing course knowledge. Using the authoring part, teachers should be able to organize the course knowledge and configure the course structure. Besides, the course module should support course ware for different domains in different fields.

- **Flexible student model** The student model typically stores information about the student’s learning state, goals, preferences, and misconceptions. This information helps the planning system to personalize the teaching strategy. The initial state of the students needs to be provided, so that the system gets a primary overview of the student. In contrast to other authoring systems which expect the teachers to define various configurations with different aspects of the student model, our architecture does not require to specify a configuration for each student, the system does that automatically.

- **Modification of the pedagogical module** An important part is the pedagogical module or the teaching model, where the actual process of teaching is carried out. Methods of teaching might vary in different environments and for different domains. If the strategy of the teaching process is modified, the versatility of the system increases.

- **Specifying course content and context** Different students have different knowledge state and therefore require different learning materials. The architecture should be able to provide the most appropriate learning materials to the student.

In the next subsection a brief description of the main system modules will be provided.
4.3.2 Course Module

The new generation of the web, the so-called Semantic web, appears to be a promising technology for implementing e-Learning. The semantic web has the capability to describe elements in a way that computers can understand. Such a representation provides the process of analyzing, extracting, and integrating information, and reusing the existing ontology and data. The shared understanding which is considered to be one of the Semantic Web is primary characteristics is based on ontologies as its backbone. Ontologies provide an obvious conceptualization that describe domain elements in a hierarchal way. So, the course module in our approach is represented by an ontology as a hierarchical representation of concepts, instructional objects, and teaching materials as well as different relationships between them. The idea of this module is to employ the concepts as a core building block for the conceptual domain and to represent them as a hierarchal architecture relying on the concept-based approach [111]. The selected concept -by student- will be considered as the goal in the planning problem. Teaching materials are defined as learning objects such as digital resources that can be reused during the learning process [112]. Generally, each course has a series of concepts. Each concept is available at a different cognitive level (quantity level and quality level), and the most appropriate level is presented to the students based on the past performance. Each concept can formulate in multiple versions targeted for various scenarios and various learning styles. The authoring part is a layer on top of our architecture which is described above. Its purpose is to help the teachers to configure different authorable courses with minimum effort. The author as a teacher is responsible for constructing and defining the course structure with different relations between them, as well as defining the learning objects, taking the ontology of our approach into account. The planning system helps the domain expert in defining and driving the appropriate scenario, teaching strategy, and style according to the student’s personality. The author in our authoring environment should define the course concepts with the definition of different relation between them, the instructional object for each concept considering different types
of teaching instructions, and the learning objects for each teaching instruction. Thus, the author does not need to define the adaptation rules or specify a teaching strategy or a concept level for each student.

In our ontology we modeled our ontology based on the task center instruction strategy that was introduced by Merrill [11]. Accordingly, we define six main classes the ”scenario”, the ”Concept”, the ”knowledge objects”, ”instructional strategies”, ”Instruction objects” and the ”learning objects”. The details of our ontology will be considered in the next chapter, but generally, we define the course module as $C_m = \{<D,S,C,KO,ISc,IO,LO>,<R>,<P>\}$, where D is a course domain, S the system’s scenario such that four types of scenarios are defined $S= (Lesson, revision, game, test)$. C stands for the basic concepts used in the course. KO is the knowledge object that defines the concept, such that $KO= (information\_about, Part\_of, kind\_of, How\_to, What\_Happen)$. ISc instructional strategy components that used to define the KO. These components are defined as $ISc=((Presentation (Tell), Demonstration (Show), Activation (Recall) or Application (Do)))$. IO (Instruction objects) are objects that are used through IS to define KO, different IO are used (example, exercise, definition, ...). Finally, LO (Learning objects) is the teaching material that is connect to the IO. Different types of LO or media types can be specified (video, audio, text,... etc). R is the set of relations that could exist between the members in the same class and between the classes (relation between course concepts, relation between course concept and its knowledge objects) and P is the set of properties of each class.

### 4.3.3 Student Module

The student model is considered to be a guide to the system to generate an adequate plan for the student state. It stores information about the student which is later used by the pedagogical module. Personalized education systems aims to give the individual student an individual course lesson. Individualization is the activity in where the system is changed w.r.t. the user. Thus, to individualize
the student’s requirements, different parameters should be considered such as the student’s goal, preferences, performance history, current emotions, and motivation. This is typically performed by explicit user actions. The data in the user model is used to personalize the presented information. This information is used by the planning system to generate a personal course for the student.

The student has different personality types. These personalities have different concerns which allows us to predict the student’s responds in different situations, foresee the degree of stress, specify their preferences, and define the best coping strategy. Personality is defined as a permanent pattern of characteristics that discriminates between people in their feeling thought and behavior. Different Types of personality are used in our system, and research that makes a connection between personality type and student attitude was considered. All the details about this module will be explained later in the next chapters. The ontology is used to describe the student module. This module provides a basis of the adaptation policy employed by the ITS. Different from these parameters are dynamic parameters. They are changed according to the performance of the students. Where the other parts of these parameters are static, as overall nature of the student, student personality or Learning styles. In PANDA.TUTOR we will consider the personality type as evidence for these parameters as the convenience style, scenario, and so on. The ontology of our approach contains the following classes: 

\[ SM = (SI; SP; Sg; Sc; Sph, Se, Seh, Sm; SmhSls; Sh) \]

where, personal information Sp: includes all permanent identification (ID; name; birth-dates; gender; mail). Sp Personality type = (spectator, insecure, sceptic, brooder, hedonist, impulsive, entrepreneur, complicated). The selected concept is considered to be the student’s goal Sg, cognitive state Sc: contains quantity Level=(beginner, intermediate, advanced) and quality level (low, good, very-good, excellent) for the hall course as general and for each part(concept, teaching material) as specific. Sph is student performance history that register the performance state for each course part and for each teaching material. Emotional state includes the current emotional state and the reason behind this emotion Se = (emotion, reason), where Seh is student emotional history. Motivation state. Sm
includes the current motivational state, while Smh is student motivational history.

### 4.3.4 Pedagogical Module

This module is considered to be the backbone of the system. It communicates with other modules and provides adaptive instructions to the students by executing the actual teaching process. Thus, this module refers to the student module and the course module for its decision. Two main processes are defined on this module: The dialog based planning and the course generation based planning. The aim of the dialog based planning system is to improve the emotional intelligence of a student, grasp the information necessary for the apprising process and regulate the emotional state of the student. Appraisal process is the process of extraction of the emotions according to the student’s appraisal of events that cause specific reactions in different people based on appraisal theory [113]. The appraisal theory assumes that the event has no meaning in itself, but has a significance for the evaluation of the person’s interpretation considering the person’s beliefs, desires, intentions, and ability of control. Our work depends on the student’s emotion. Therefore, we need to achieve the following conditions: To know the current emotional state of the learner, to determine the impact of an action on the learner emotional state, and to select the most advantageous plan that improves the student performance. The details of this process will be explained later, in the next chapters. The course generation planning system considers the appraisal parameters and coping way for selecting the adequate learning scenario and tailoring the lesson for that scenario, aiming to increase the student’s performance. Each scenario contains different teaching strategies and learning styles that are suitable for the student’s personality and the current emotional and motivational state. The hybrid planning is used to model this module. Teachers do not have any direct control over the produced plan, they need only define the course structure and teaching materials. Four types of scenarios are defined in PANDA.TUTOR. The term scenario means a discrete sequence of steps inside the learning process. In these scenarios the emotional and motivational states are used for
generating the lesson for the concept goal. Different emotional and motivational goals are used, for instance Increase-Confidence, Increase-Effort, Decrease-Fear, ... etc. In all these scenarios the generated course is constructed based on the Task-Centered instruction strategy that was introduced by Merrill. Before we explore this scenario, we will review the Task-Centered instruction strategy approach.

### 4.3.5 Task-Centered Instruction Strategy

During the past few years there have been various approaches of instructional design strategies including guided discovery learning, model centered instruction, problem-based learning, situated learning, case-based learning, and exploratory learning [114]. These approaches have in common to give students the opportunity to involve them selves in whole tasks or problems. In contrast the traditional curriculum approaches or topic-by-topic approaches teach topic-centered instructional strategies typically task components in a hierarchical fashion by teaching all the related skills of one type and then the related skills of the another type, chapter by chapter, until all of the component skills have been taught. Learners are then given a task to which they can apply their skills as a final project in a course. A topic-centered approach is often characterized as the approach of skill development “you won’t understand this now, but later it will be very important to you”. It is a fact that getting students involved using realistic situations will help them form appropriate schemas and mental models which helps them complete internal representations are believed in student’s mind to facilitate their later application when student tries to solve a new acquired knowledge and skill. Problem-based approaches take the form of a scaffolding, a learner’s guide, or coaching. A typical problem-based instructional strategy gives a small group of learners a complex problem to solve, identifies resources that can be used to solve this problem, and expects learners to acquire the necessary skills by searching the resources and developing a problem solution. Learners are expected to learn from each other and to seek other sources when the identified resources are
insufficient to solve the problem. But the disadvantage of this approach is that the strategies are more structured and provide the learners already familiar with a given Content area unstructured exploratory learning approaches may be appropriate. However for learners who are novice in a content area, learner guidance is essential. Related to this approach is [115] where strategies are more structured and provide more guidance to the student. A task-centered instructional strategy is not the same as problem-based learning [11]. In this strategy each topic in a given area is taught in turn. Figure 4.3 illustrates a task-centered instructional strategy. This approach incorporates students in the whole task early in the instructional sequence.

**Task-Centered Instructional Strategy**

1. Show a new whole task.
2. Present topic components specific to the task.
3. Demonstrate the topic components for the task.
4. Show another new whole task.
5. Have learners apply previously learned topic components to the task.
6. Present additional topic components specific to this task.
7. Demonstrate the application of these additional topic components.
8. Repeat apply, present, demonstrate cycle (steps 4-7) for subsequent tasks.

![Task-Centered Instructional Strategy Diagram]

**Figure 4.3** Task-Centered instruction strategy

The instruction starts by demonstrating the first whole task in the progression. The first demonstration should be a complete task but it should be the least complex version of the whole task in the progression. This demonstration forms the objective for the task and provides the context for
the students. The second task provides questions about the first task and introduces more information, and so on. That helps the students to easily grasp a demonstration of the whole task. Each task in the diagram indicates an entire complex task. The increase in the size of the task indicates the increase in the task complexity with each subsequent task in the progression. Based on

![Diagram of the first principles of instruction](image)

**Figure 4.4** First principles of instruction Merril.

task-centered instruction strategy, Merrill defines the first principles of instructions in Figure 4.4. These principles describe a cycle of instructional phases consisting of activation, demonstration, application, and integration all in the context of real world problems or tasks. This approach offers a different aspect to the content analysis as compared with the more traditional instructional systems development (ISD). Traditional ISDs specify objectives of the phase in the design process but do not specify the actual content to be taught during the development phase.
4.3.6 PANDA.TUTOR Scenarios

According to the student’s emotional, motivational and cognitive state the system selects the appropriate scenario. In these scenarios we regard a number of different learning strategies considering the competency based learning.

Competency is defined as ”a combination of skills, abilities, and knowledge needed to perform a specific task” [116]. The hierarchical relationships among terms commonly used in the learning process and competency are depicted in figure 4.5. Each level in the pyramid affects the level above. The first level of this pyramid consists of traits and characteristics which depend on the student personality type. The differences in students’ personality help us explain the difference in student skills, abilities, and learning styles. The second level consists of skills, abilities, and knowledge which are developed during the learning process. Competencies, then, are the result of the integration of skill and knowledge. And finally, demonstration is the result of applying the competencies.

![Figure 4.5 A conceptual learning model](image-url)
Competency based models mainly depend on measurable assessment. The learning process employs assessment strategies. The competency approach can be considered to be a way to support the presentation and assessment of concepts from different prospective by giving varying tasks to students. These tasks differ in the required skills, abilities, and competencies.

There are clear advantages in using competency based learning models because measuring competencies permit the student to return to one or more course parts that have not been mastered during the learning process. In addition, competencies also provide students with a clear map and the navigational tools needed to move towards their goals.

These scenarios are the following:

- **Lesson scenario:**

  In this scenario the student can learn or study not only the selected concept, but also the prerequisite and related concepts, considering first the principles of instruction [117] and task-centered instruction strategy which helps in constructing the student’s knowledge. The scenario is started by representing presentation/demonstration information about the concept. Periodically there is practice or test activity to assess the degree to which learners are acquiring the knowledge or skill component being taught. After completing the instruction of the first topic, the next topic is taught in similar manner. Such that if a student is unsure of how a given piece of content will be eventually used it lacks the necessary relevance. Acquiring knowledge and skill components out of context makes it very difficult for students to form mental models over how this information applies in the real world. Besides, acquiring this skill in the context of whole tasks makes it more likely that learners will form mental models regarding how these individual skills are integrated into the overall performance. Toward the end of the lesson a culmination of experiences are taught where students are required to apply their new knowledge and skills. In the traditional instruction it is not always clear to the learners how this component knowledge and skill should eventually be applied.
We want to avoid a situation like “You won’t understand this now but later it will be really important to you”.

**Revision Scenario:**

The student in this scenario can construct and review his/her knowledge. Therefore, a complete simple task is introduced to the student with a guide for solving this task. Thus, students are then given instructions - presentation, demonstration of the skills required to do this task. Afterwards, a more complex task is introduced, asking the student to solve the presented problem and introducing a new part, they are taught additional skills or more detail for the initial skills that are required for this new task. Students are asked to recognize how the previous and new skills are used to complete the task. Finally at the last level, the student can solve a complete task.

**Companion Game Scenarios**

Game scenario is an important factor in the learning process: It helps the student adapt his/her emotion, decrease the student’s boredom, and increase motivation and attention. The development of intelligent tutoring systems has long been focusing on applying artificial intelligence and cognitive science in education. A new branch of intelligent learning environments called learning systems was developed by Frasson [118]. In contrast to an intelligent tutoring system, in which a computer mimics an intelligent tutor, the learning companion system assumes two roles, one as an intelligent tutoring system and another as a learner’s companion. Using a learning companion, not a human, to train the student gives the system the ability to control the competency and behavior of the learning companion for a particular pedagogical strategy.

Four types of companion strategies are used as training scenarios: Companion with assistance friend, companion with competitor friend, companion with disturber friend and companion as teacher to your friend. Within these strategies we developed different scenarios.
The system will select an appropriate strategy type for the current cognitive, emotional and motivational student’s state, and then adjust the learning material level and interaction type for each scenario for that state. The companion game strategies are as follows:

- **Training with an assistant friend**
  In this strategy the student can co-operate with the system as his/her assistant friend. Based on the assumption that the student is more willing to learn from an assistant friend than a teacher. Three scenarios are defined according to the student’s cognitive level, emotion and motivation state. The companion and the learner perform the same task and exchange ideas for solving the problem, they work together and ask for help only if they cannot find a solution. This assistant has a quite similar level of knowledge. In the second one, the knowledge level of the assistant is higher slightly than that of the learner, supposing that the assistant passed the problem stage and s/he helped the student solving the problem. In the third scenario, the student should solve the exercise alone, without help from the assistant if s/he does that, the score is decreased.

- **Training with a competitor friend**
  In this strategy the competitor has a lower, comparable or higher level of knowledge. Thus, the system defines the level based on emotional or motivational state. The competitor and the student compete to get a higher score. They compete to solve the same exercise, or a different exercise. Hence, the teacher will ask the student to solve a problem. If the answer was right, s/he will get the score, otherwise the competitor is asked to solve it and vice versa (note that, the system enforces the competitor to solve (or not) based on the student emotional and motivational state). Likewise, the student can collaborate with another companion to compete against another learning companion.

- **Training with a disturbing friend**
  In this approach the teacher asks the student a question and the disturber or the trouble-
maker interrupts the student to give him/her his solution. This solution could be right or wrong and the student should decide to accept or refuse the disturber’s solution. The disturber (or the teacher) could wait until the student answer’s the question then asks about the confidence on his/her answer. This strategy helps the student to have confidence in his/her answer and it enhances knowledge acquisition.

– Learning by Teaching

Here the computer takes on the role of a student who is taught by a human student [119]. The idea of this strategy is to encourage the student to act as a teacher, which helps in increasing the student’s confidence and efficiency. The student (as a teacher) can help the companion solving the exercise, justify the answer, or explain the right answer. The tutor takes on the role of a teacher and provides knowledge and support when the companion faces difficulties. The student can teach one student or more. Learning by teaching approaches do not attempts to introduce new knowledge, but instead force students to brainstorm to their knowledge.

• Test Scenario

Student evaluation or assessment is a basic issue in our system. One of the standard ways of achieving that is through testing. The test in our approach consists of a number of exercises with specific levels of difficulty corresponding to the student level. The student uses this module to take a complete test about the course. The system delivers adequate test for the student’s cognitive level. From the testing results, the tutor is able to specify each student’s progress.

4.3.7 Generated Plan

One of the main goals of our approach is to improve the usability and maintainability of information in the course environment. That depends on retrieving relevant information and visualizing it...
in an appropriate way. The generated plan automatically assembles learning objects retrieved from one or several repositories to generate the lesson for the student according to the generated plan. The generated plan will be a set of querying statements that are used to query the course ontology to retrieve the learning objects according to the generated plan.

4.3.8 Student Interface

The student interface is the system’s getaway for communicating with the student. Through this interface, all the learning materials are presented to the student for interaction. Through the student interface, the student can define for the first time the required personal settings and take the personality type test. After subscription the student can, at any time, enter the system using her ID. In addition, it collects the student’s states during the interaction with the student. Then, it sends this information to the dialog phase or Lesson Generation phase to analyze the cause and expect the coping way. If the emotional appraisal module needs any more information from the student to complete the appraisal process, it creates a new dialog and sends it to the interface module to get new information from the student.

4.3.9 Author interface

The teacher’s responsibility as an author is to define the domain concepts and the interrelation between them, the example and the exercises, the answers, hints, and far more. This process includes the insertion of available materials into the repository.
Chapter 5

Course Module

Personalized e-learning is a new direction for intelligent tutoring system (ITS) and adaptive hypermedia systems AHMS. The task of building a course module is a difficult process and requires time and effort. This chapter introduces an ontology to describe the course module of our PAN-DA.TUTOR system. Thus, developing the course ontology is a step toward creating a shared and reusable adaptive educational system. Also, ontologies permits the retrieval of learning material when the pedagogical module has generated the course. The proposed ontology model is a general educational ontology based on various learning strategies and instruction design theories. Furthermore, we have realized a separation between the course Content reusability and the learning objects reusability. In addition, this ontology can be used by the author to construct a course with minimum effort. The authored course later is consulted by the pedagogical module to construct a personalized lesson for the student.

5.1 Ontology and Semantic Web

The term "Semantic Web" embraces efforts to build a new WWW architecture that enhances content with formal semantics. The semantic web is an extension of the current web in which infor-
Information is given a meaning that can be understood by a computer. Likewise, it realizes automatic fetching and processing of web data. The core of the semantic web is ontology, which is used to explicitly represent conceptualizations.

The Semantic Web stack in Figure 5.1 has been proposed and gradually refined by Berners-Lee [120]. It is supposed to guide us through the process of increasing the level of semantics, as well as always to be updated with the new corresponding web technologies. The first three layers contain technologies that are well known from hypertext web and provide a common syntax. Middle layers contain technologies standardized by W3C to enable building semantic web applications and add semantics to the Web. The top layers contain technologies that allow new knowledge to be inferred from the explicitly provided information and to check the validity of the statements made in the Semantic Web.

![Semantic web stack](image)

**Figure 5.1** Semantic web stack

The very basic layers of the semantic web are resources. They provide a standard way to ex-
change symbols identified by their unique resource identifier (URI) or internationalized resource identifier (IRI) based on ASCII Unicode. The next semantic layer is XML which represents the structure of data. Extensible XML is general purpose markup language based on IRI, URI and Unicode. XML allows users to add arbitrary structure to their documents but does not encode the meaning of the structure.

On top of XML the resource description framework, RDF is defined. Simply put, it as language to describe whole resources and represent the meaning of data. Thus, the semantic annotation of data is done by means of RDF\(^1\). RDF is a data model language developed by the World Wide Web consortium (W3C). It is an infrastructure that enables encoding, exchanging, and the reuse of structured metadata. It is based on the idea that things being described have properties. These properties have values and resources can be described by making statements. Thus, RDF describes how to make statements about resources, in particular, web resources, in the form of subject-predicates-object expressions which are known as triple in RDF terminology, for example ([subject] ‘Java’ [predicate] ‘has website’ [object] ‘www.java.com’). The semantic annotation data gives the possibility to perform some kind of reasoning. RDF Schema\(^2\) adds a new layer of functionality by allowing the representation of ontologies. RDF Schema (RDFS) is a vocabulary description language for describing properties and classes of RDF resources, with semantics for generalization hierarchies of such properties and classes. RDFS is based on an object-oriented approach and is used by introducing the notion of the class of similar resources, i.e. objects showing a set of the same characteristics. Resources are then viewed as individuals of some class: Every instance of class is also an instance of it’s super class, from which it inherits the properties.

To summarize, resources on the web are considered to be a set of available web resources, each identified by a URI. Such resources are mainly represented by plain XML descriptions. RDF allows a web document to be written in a structured way, using a user-defined vocabulary. However,

\(^1\)(http://www.xml.com/xml/pub/98/06/rdf.html)  
\(^2\)(http://www.w3.org/TP/PR-rdf-schema/)
both RDF and RDFS suffer from the lack of formal semantics for its resource modelling. The semantic web resolves this problem by adopting URI for the pieces of information it can retrieve. Besides it supports relations between these pieces. These identifiers are similar to web page URLs but not necessarily point to a web pages. In addition, these identifiers solves the problem of retrieving data. Moreover; the semantic web can link the individual pieces of data to their conceptual categories, in order to support the reusability of data. This can be achieved using ontologies by considering hierarchal structures of concepts with different relations between them and by binding the individual or instances of data to those concepts. The ontology provides an explicit conceptualization that describes data semantics. The ontology also provides a common vocabulary, interpretable by machines, to researches that need to share information about a domain, including definitions about a basic concept of a domain and its relationships.

This way, the ontology is the backbone of the Semantic Web, a new form of Web content that is meaningful to computers [121]. Ontologies are part of the knowledge representation field of research. They can assist developers to define objects that exist in an application domain [122]. Ontologies define specifications for the conceptualization and the corresponding vocabularies that are used to describe a domain. Besides, ontologies allow the definition of an infrastructure for integrating intelligent systems at the knowledge level, independent of particular implementations, which enables knowledge sharing. Together with various reasoning modules and common knowledge representation techniques, ontologies can be used as the basis for the development of libraries of sharable and reusable knowledge modules [123].

Ontologies are appropriate for describing heterogeneous, distributed, and semistructured information sources that can be found on the web. Typically, ontologies consist of definitions of concepts relevant to the domain, their relations, and axioms about these concepts and relationships. Several representation languages and systems have been defined.

OWL (Ontology Web Language) provides a full grown vocabulary for defining ontologies and a
more powerful ontology language. In addition, OWL is the proposed standard for web ontologies representation, and is built upon RDF and RDF Schema. However, OWL provides additional vocabulary and allows a representation of more complex relations such as transitivity, symmetry, and cardinality constrains. In other words, OWL supports more powerful expressive capabilities which are required by real applications. In particular, ontologies make knowledge reusable by describing the properties of classes and relations between classes.

OWL provides the three sub-languages owl Lite, owl DL, and owl Full, which offer on the one hand increasing expressiveness, yet on the other hand increasing computational complexity:

- Owl Lite provides the means of defining classification hierarchies, together with simple cardinality constraints.

- Owl DL offers maximum expressiveness while retaining computational completeness and decidability. The ”DL” illustrates the correspondence to the field of Description Logics.

- Owl Full provides the full, unconstrained expressiveness of the owl vocabulary, yet without any computational guarantees.

Moreover, it is possible to reason about ontologies by means of techniques that are typical for description logics. Reasoning in the semantic web is mostly reasoning about knowledge expressed in some ontology and the ontology layer is the highest layer of the semantic web tower that can be considered as quite well assessed. Top layers contain RIF or SWRL that bring support of rules. With rules, more reasoning can be done over ontologies in knowledge base and they allow verification of knowledge consistency. This is important for example to allow describing relations that cannot be directly described using description logic used in OWL. SWRL has been used to improve the expressivity of OWL language to allow for rules in domain ontologies and to provide a way of automatic mapping between Owl ontologies. Cryptography is important to ensure and verify that semantic web statements are coming from from a trusted source. This can be achieved
by an appropriate digital signature over RDF statements. The user interface is the final layer that will enable humans to use semantic web applications.

### 5.1.1 Ontology and E-Learning

Ontologies are useful in different fields and in the educational environment, as in the case of our research. Ontologies are a research domain that help us to overcome the most common problems in intelligent tutoring systems [124, 125]. It enables the ontology designer and the teacher as an author to share a common course structure, and the educational material to be reused. Besides, it allows us to formally specify the concepts that appear in a concrete domain, their property, and their relationships. Devedzic [125] analyzed several systems described in the literature. Such analysis has been important to verify how the systems use ontologies and what are the advantages and disadvantages are. However, the problem of how to build a “good” system still remains.

Concerning the instructional aspects, the ontology covers the instructional theories by considering the description of learning materials independently of a specific learning context. The ontology also provides a standardization of the instructional role of a resource (for example, definition, example, exercise, etc.). Related to using ontologies in instructional design is the work of Mizoguchi [124] that describes general directions of research regarding general ontology for the educational environment. Aroyo et al. [126] describe how an assistant layer uses an ontology to support complete authoring, for instance by providing hints on the course structure. Recent work by Hayashi et al. [127] develops “an ontology of learning, instruction and instructional design” in which the ontology formalizes the learning process by describing task events (called “educational events”) which can be decomposed in various ways (methods), where each method can be decomposed into smaller events. The author can use ontologies for supporting and providing guidelines for the learning theory. However, drawbacks of this approach are that the ontologies misses some requirements necessary for this task. First, the event as a task decomposition does not contain pre-
conditions and the selection of an event depends only on the instructional theory used in the current authoring process. Additionally, the connection to the resource layer is missing: Basic events are not coupled with partial descriptions of educational resources that can be used to locate resources that support the achievement of the educational event. Furthermore, different scenarios are not represented. There are systems that have a pedagogical model but do not show a clear representation of this model nor the relationship among the components. This is the case of AIP - Adaptive Instructional Planning using Ontologies [128]. Silva [129] describes ontology-based meta-data to achieve personalization and reuse of content in the adapt web project where DAML+OIL is used to represent the ontology.

For course structure Henze et al. [130] propose an approach based on domain ontologies. This approach describes the features of the domain ontology and the learner ontology, as well as observing the learner’s interactions with the e-learning system, and of the presentation for generating hypertext structures. Hence, the authors propose the use of several RDF ontologies for building adaptive educational hypermedia systems. For the authoring aspects the ontology of instructional objects assists authors to search for the teaching material or describe the conceptual model of the content structure [131].

Karina et al. [132] enable dynamical building of the learning courses according to the user preferences. It is based on the conceptual description of the learning material using conceptual graphs and uses some (prerequisite) strategies to achieve the users objectives in the search/navigation process. However, this approach does not describe the explicit structure of the course. The Collaborative Courseware Generating System uses modern web technologies (XML, XSLT, WebDAV) for describing course structures, but without explicit ontology support. It also does not define the context and structure of the learning materials explicitly [133]. Ullrich [51] used the ontology to assemble learning resources to generate a curriculum that takes into account the knowledge state of the student, the preferences and learning goals.
Different systems use ontologies in their design, for example AIMS and OntoAIMS [126] (the difference between the two systems is that the first one used a concept map to implement the domain model, while the second one used OWL and DAML-S). Both of which include an ontology driven subject domain, a repository of learning resources, and adaptive and personalized modules based on the conceptual domain knowledge.

The course structure in these systems considered the course as a set of concepts with the usual relation between them as is_a, part_of, prerequisite, or following relations. And the delivered concept to the user as goal is usually a single concept without considering the related concepts of the goal concept.

The contribution of our model is to build a general cognitive and constructive conceptual construction of an ontology that can be used for different types of courses, and help the planner to construct the student’s knowledge. We consider a new module to construct our ontology based on instruction design theory of Merrill, as we will explain in the next sections. The ontology of the course content module has been developed under the OWL language, the last standard language W3C to represent the ontologies on the web. The ontology provides a set of terms which should be shared among the authors, and hence could be used as well-structured shared vocabulary. Of course, the use of ontologies is a way of describing the semantics of information on the Web. It is important to make the information machine-readable: Especially the amount of data is growing continuously.

The Protege version 4.1 framework\(^3\) has been selected to edit and construct the contents. Protege is an ontology development that has been used to construct the domain ontology.

\(^3\)(http://protege.stanford.edu)
5.2 Concept Map Based on Adaptive Educational System

Concept mapping has become popular in both the educational and business fields. It is considered to be a specific kind of mental model that plays an important role in modeling educational courses. Concept mapping is a technique of representing concepts and their hierarchical interrelationship as a graph, which nodes and arcs represent concepts and relationships between them. The hierarchical fashion of the concept map represents the most general concept at the top of the map and the more specific then less general concepts or sub concepts arranged hierarchically below [134]. Another property of concept maps is the inclusion of cross-links which represent relationships between concepts. The final feature that could exist in concept maps is the ability to add examples of events or objects that can help clarify the meaning of a given concept.

At the beginning of the course, concept maps can be useful as a means of conveying the course contents, and providing students with a big-picture as an overview over the course. They can also help the instructor to divide the course into several levels, to relate the material to the overall course concepts, and to assess what students understand and how. Also, it helps the student to review the course.

There is a number of concept-based adaptive web based educational systems [16, 123], which include a concept based subject domain a repository of learning resources, learning tasks, and adaptation rules. Concept maps are quite similar to ontologies [135]. Both have classes or concepts and relations between them. But the ontology has attribute for the classes as a data type property. In addition we can define instance(s) of a class. Therefore, ontologies are a technique for representing concepts, objects, or things which are defined by classes. So, representing the concept map as an ontology helps us represent knowledge structure. Ontologies typically consist of a hierarchal arrangement of classes and subclasses. Mapping between main elements of an OWL ontology and elements of the concept map are introduced in [136] (see Figure 5.2).

For each class a set of properties can be defined describing the class which is known as data
properties. There exist relationships between those concepts (classes) which are defined by object properties [137]. In addition a set of individuals or instances can be defined for each class and provide axioms as rules and constraints for manipulating the terms and their relations within the domain. Ontologies also provide knowledge sharing and reasoning.

Semantic web technologies form a new trend in modeling and developing adaptive web-based educational systems (WBES). The current direction of research into the design of web-based educational systems indicates that concept-based systems provide adaptation and personalized learning [100].

In our work we considerer the concept map approach during the development of an adaptive WBES. WBES explores how concept map standards can be used to represent the knowledge of the system concerning the educational content, the course knowledge, student model, and pedagogical aspects. In addition, we add other relations for defining the concept map.


5.3 Learning Objects

Learning objects are an important factor for e-learning. Learning objects are the use units of content of digital resources that can be shared and reused to support the teaching and learning process [112].

Learning objects are defined as learning resources or an entity of learning of the smallest term of learning material that has the ability to be reused from one course to another. Thus, it can be used in different learning scenarios or instructional situations, which reduces the cost and time of developing the course content. The instructional planning component of an ITS is a component responsible for generating individualized sequences of learning material. The instruction planning approach is adapted to the e-learning fashion and it uses the learning objects as a course content to provide a high level of individualization. The learning objects are designed to assist the system to deliver different strategies and styles for the course. Such that the learning object can be used in contexts that differ from what it originally was designed for.

The LO is one of the main research topics in was e-learning. Especially, the direction of the researches recently aims to increase the reusability and granularity issues of LOs. In the e-Learning community three metadata standards are emerging that describe e-Learning resources. Meta-data is considered to be helpful in such a process. This provides an adaptive learning environment and reusable educational resources.

Metadata is information that provides a common set of tags that can be applied to any resource, regardless of creator, the used tool, or the place of storage. Tags are data describing data, that is used to enable us to describe, index, and search resources, which are essential for reusing learning objects. In the e-Learning community different metadata standards are emerging to describe e-Learning resources, for instance IEEE LOM⁴, SCORM⁵ and IMS ⁶. Those meta-data models

define how learning materials can be described in an interoperable way. All the metadata elements are necessary to describe the resources and they can be classified into several categories, each offering a distinct view on a resource.

Different systems have developed their own standardized metadata vocabularies to meet their specific needs. Although these standards are defined to offer the base for educational information sharing, there are still some shortcomings in the e-learning domain. These standards do not include any domain ontologies which can be specified building on formalisms. Note that, most of those metadata standards lack formal semantics. Thus, they introduce the problem of incompatibility between disparate and heterogeneous metadata descriptions. Also, the metadata is not enough, it does not consider the pedagogical aspects and does not allow adequate searching for information. In these models the student or the authors interact only with the sharable content object. The metadata just provides some descriptions of LO’s properties, it lacks the reasoning capabilities. That means, different learning systems cannot understand the meaning of each other. In addition, these standards did not cover the instructional function of the learning resource, and, in addition, were not designed for the Semantic Web. For instance, LOMs educational category allows the description of resources from an instructional perspective. Possible types of learning resources are, among others, Diagrams, Figures, Tables, Exercises, Narrative Text, Exams. In LOM, these values are provided as a list, without taking into account the inherent structure which is represented in the ontology. More critical, the LOM types combine instructional and technical information. The first three values of the above example describe the format of a resource, whereas the last three cover the instructional type. Furthermore, several instructional objects are not covered by LOM (e.g., definition, example). IMS LD describes ordered activities during learning and the roles of the involved parties. It does not represent single learning resources and their instructional functions. On the other hand, many non-SCORM systems such as DCG [46], [18] support the pedagogical approaches but have a number of problems, such as lack of interoperability or flexibility, and
they do not support the reusability of educational materials. The educational resources are entities that can be presented in the web-based learning environment and thus should be identifiable and addressable. These contents are stored as learning objects, which are described and accessed by means of metadata. The learning objects repository holds the contents that are to be presented to the students. With these standardized descriptions of learning materials, it is possible to build large learning repositories in which teachers and students will be able to store or retrieve valuable materials for learning for preparing courses. Current approaches are ARIADNE\(^7\) or EDUTELLA\(^8\). Some researches make it possible to use the ontology to describe learning objects. Thus, the lack of a shared understanding between terms in one vocabulary as well as between terms in various metadata vocabularies might be avoided by using ontologies as a conceptual backbone in an e-Learning. The role of an ontology is to formally describe shared meaning of the used vocabulary. According to our ontology we consider the terms of LOM standards during ontology, as we will explain later in next sections. To make web content machine-understandable, web pages and documents themselves must contain semantic markup, i.e. annotations which use the terminology that one or more ontologies define and contain pointers to the network of ontologies. Semantic markup persists with the document or the page published on the Web and is saved as part of the file representing the document/page. The LO has a URL of each learning object because its responsibility is to provide information to the learning objects which are designed by the author.

In our work, learning resources are files with different formats (HTML) queried from web repositories. In our prototype, we use the Protege OWLDoc plug-in function to define different course resources.

\(^7\)(http://ariadne.unil.ch/Metadata/)
\(^8\)(http://edutella.jxta.org)
5.3.1 Knowledge Objects and Instruction Strategies

Merrill [138] defines a set of knowledge objects to describe the subject matter content or knowledge to be taught. These knowledge objects as a framework to organize the knowledge base of content resources. So, different instructional algorithms can use the same knowledge objects to teach the same subject matter. In other words, the components of knowledge objects are not specific to a particular subject matter domain. The same knowledge object components can be used to represent a variety of domains (e.g. mathematics, science, humanities, technical skills, etc.). To organize the knowledge base five types of knowledge objects are defined: information about, parts of, kinds of, how to, and what happens. Knowledge objects are based on the instruction transaction theory (ITT). ITT is an extension of CDT which is described in chapter one in a more detail. Merrill [139] describes the contents in a way that they can be manipulated by a computer system to automatically create instructions from only the content. In addition, he develops a more detailed description of instructional strategies. It is considered to be a new manipulation of the instruction planning. As in instruction planning the ITT has two sides. The first one what to teach to describe knowledge objects considering three types of knowledge: entities, activities, and processes, as well as identifying properties and define each of these components in terms of properties. For each knowledge object, two levels of information levels are defined; information_level (generality) and a portrayal_level (instance) for each of the five categories of learning outcomes, defining four instructional strategies for a primary presentation form; presentation (TELL), demonstration (ASK), recall (SHOW), apply (DO).

The ontology described in this chapter provides a vocabulary that captures the instructional strategy for the component skills, the first principle of instruction and task center instructional strategy. Table 5.1 summarizes each of these four instructional strategy components for each of the five types of instructional outcomes [139].
Table 5.1 Instructional strategies for component skills (Merrill)

<table>
<thead>
<tr>
<th>Description</th>
<th>Information</th>
<th>Portrayal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRESENT(TELL)</td>
<td>RECALL(ASK)</td>
</tr>
<tr>
<td>Information-about</td>
<td>Tell the name and information</td>
<td>Recall the name and information</td>
</tr>
<tr>
<td></td>
<td>Parts-of description and location</td>
<td></td>
</tr>
<tr>
<td>Information-about</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kinds-of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>How-To</td>
<td></td>
</tr>
<tr>
<td></td>
<td>What-happens</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tell the name and information</td>
<td>Recall the name and information</td>
</tr>
<tr>
<td></td>
<td>Tell the name description and location</td>
<td>Recall the name description and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>description and description</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tell the steps and their sequence</td>
<td>Recall the steps and their</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sequence</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tell the conditions and consequence involved in</td>
<td>Recall the conditions and</td>
</tr>
<tr>
<td></td>
<td>the process</td>
<td>consequence involved in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>process</td>
</tr>
</tbody>
</table>
5.4 Course Module Ontology In PANDA.TUTOR

The course space in our approach has two main sub-spaces. The course structure as the course domain space and course content as the media space. The domain space consists of most important concepts and the learning goals of the course domain with various cognitive levels and relationships between them. While, the media space is a semantic network that is named the resource network which is used to represent teaching materials of course Content. The separation between representation of course models and the repository makes it possible to use different kinds of materials for the same domain subject (concept). \( [LO\ repository\ has\ URL\ (uniform\ resource\ location)\ to\ each\ LO\ because\ the\ repository\ is\ exactly\ to\ provide\ this\ information\ to\ presentation\ model] \). The proposed domain is a framework for the instructional design of knowledge objects from the viewpoint of the ontology and semantic web. The domain ontology has been developed under OWL language. We have developed a general educational ontology that can be used for different knowledge domains and for different adaptations of personalized e-learning. The Protege 4.1 framework has been selected to edit and construct the contents. In the proposed ontology not only class taxonomy but also properties of classes and relationships of the main class in its own subclass have been built. In addition, relationships between main classes that are used to connect a semantic network of domain model have been built. For that reason, object properties and data type properties have been defined for each class of the ontology and its subclasses. The object property is a link that connects two individuals of some classes. Also, the object property is a connection between two nodes in the semantic network.

According to our approach the course domain is defined in six layers: The course space, the course concepts, the knowledge objects, the instructional strategy components, instructional objects, and learning objects. The first five layers represent the course domain ontology and contain knowledge about the domain and the curriculum structure. The last layer represents the media space. Consequently, in our ontology six main classes are defined as a hierarchical fashion, the course class,
concept class, knowledge object class, instructional strategy components "as short instructional components" class, instructional objects class, and the Learning Objects class.

Consequently, in our ontology six main classes are defined as a hierarchical fashion. The course space, the course concepts, the knowledge objects, the instructional strategy components, instructional objects and learning objects. For each course, a set of concepts can be defined. For each concept different types of knowledge objects could be defined, for each knowledge object (KO) different instruction components (IC) can be defined, different instruction objects (IO) can be specified for each instruction object according to the course and the preferences of the author. Finally for each instruction object different media or learning objects (LO) can be determined. In our ontology we will consider different kinds of relations: Relations among Concepts, relations between the concept and its knowledge objects, relations between knowledge objects and its instructional components, relations between the instructional component and instructional objects and relations between the instructional object and its teaching material types.

These layers as depicted in figure 5.3 are as follows:

- **The Course layer** $Course \equiv Concept_1 \sqcup Concept_2 \sqcup \cdots \sqcup Concept_n$.

  The course class represents any course of the learning environment. This class consists of a set of concepts that define the course. It considers the main class which represents the subjects being taught in an educational application. For example, the artificial intelligence could be individuals of this class (see Figure 5.4). The course class contain several data properties that describe the names and a brief description of the course as; CourseName, CourseDescription and the objective. Different object properties (relations) are defined. For instance HasObjective points to the objectives and links between the course and the class objective, whereas hasConcept (isConceptIn is its inverse) connects the course and its concepts, and the hasResource relation points towards the set of learning objects that compose the course.
• **The concepts layer:**  $\text{Concept} \subseteq \text{Course}$, and $\text{Concept} \subseteq \exists \text{has KO.KO}$. The course in our ontology considers the concept as a main building block. Course concepts refer to basic concepts concerning the course. A concept holds one unit of knowledge and explains different aspects of it with different types of teaching material. Thus, the concept can represent a bigger or smaller course structure (as a composed or primitive concept). The ontology for this model preserves the relationship between the concepts like the concept map, such that the author can specify various interconnections or relationships between concepts. In addition, the concept structure is built to represent a domain ontology that provides a structure for domain knowledge representation. Thus the ontology specifies the concepts to be included and how they are interrelated. The course concepts consist of a hierarchal representation of the main concepts and relations between them. Thus, various concepts can be linked to other concepts according to their relation. In this way, one or more concept networks could be constructed representing the course structure of the course to be taught.
5.4 Course Module Ontology In PANDA.TUTOR

- Introduction to Artificial Intelligent
- Problem Solving
- Knowledge & Reasoning
- Planning
- Uncertain Knowledge & Reasoning
- Learning
- Communication, Perceiving and Acting

Figure 5.4 Artificial intelligence course

The concept is represented by a class. Considering different data type properties and object properties for the concept class. The data properties of class concept are as follows;

\[ \text{Concept\_Name, Author\_Name, Date\_of\_Authoring or Educational\_level.} \]

Regards Object properties, first, we will take a look at the relations(object properties) between the concepts:

- \( \text{Concept} \subseteq \exists \text{has\_NextConcept\_Concept} \): Indicate the next concepts in order.

- \( \text{Concept} \subseteq \exists \text{has\_Prerequisite\_Concept} \) and

\( \text{Concept} \subseteq \exists \text{is\_Prerequisite\_for\_Concept} \).

Such that; \( \text{has\_Prerequisite\_Concept} \equiv \text{is\_Prerequisite\_for\_Concept} \): Point to concepts that must be known before stating the corresponding concept.

- \( \text{Concept} \subseteq \exists \text{SimilarTo\_Concept} \) and
Chapter 5  Course Module

Concept \subseteq \exists \text{OppositeOf}.\text{Concept}.

Such that; SimilarTo \equiv \text{OppositeOf}^{-}: connects a concept to other concepts that have the same or different semantic meaning, respectively.

- Concept \subseteq \exists \text{RelatedTo}.\text{Concept}: this relation relates concepts that have a kind of relation between them.

- knowledge objects layer: KO \equiv \{\text{Information\_About} \sqcup \text{Part\_of} \sqcup \text{Kind\_of} \sqcup \text{How\_to} \sqcup \text{What\_happen}\}, and KO \subseteq \exists \text{has IC}.\text{IC}.

Thus, each concept can be connected to five types knowledge objects information\_about, Part, subclasses, How-to (process), what-happen (principle or condition) (see Figure 5.5). We represent the Knowledge\_objects class as an abstract class of subclasses; Information\_about, Parts\_of, Kind\_of, How\_to and What\_happen.

Regarding the relation between a concept and its knowledge objects considers the following:

These knowledge objects can be connected to the concept according to the following relations Concept \subseteq \exists \text{hasK}.\text{KO} and KO \subseteq \exists \text{isK\_for}.\text{Concept}

Such that; hasK \equiv \text{isK\_for}^{-}: indicates the main knowledge objects for the concept, or the concept of these knowledge objects, respectively. These properties are abstract properties which define the following sub properties respectively;

has.KO \sqsubseteq has.P\text{art} \sqcup has.k\text{ind} \sqcup has.H\text{ow\_to} \sqcup has.W\text{hat\_happen}

is.KO\_for \sqsubseteq is.P\text{art\_of} \sqcup is.k\text{ind\_of} \sqcup is.H\text{ow\_to\_for} \sqcup is.W\text{hat\_happen}

Such that; has.Information\_about \equiv \text{is\_Information\_about\_for}^{-}

has.P\text{art} \equiv \text{is\_Part\_of}^{-}

has.k\text{ind\_of} \equiv \text{is\_kind\_for}^{-}

has.H\text{ow\_to\_o} \equiv \text{is\_how\_for}^{-}\text{and}

has.W\text{hat\_happen} \equiv \text{is\_what\_happen\_for}^{-}
Figure 5.5 Knowledge object of planning concept in artificial intelligence course

- **instructional Component strategies Layer**: \( IC \equiv \{\text{Presentation(Tell)}, \text{Demonstration(Show)}, \text{Activation(Recall)}\text{or Application(Do)}\} \), and \( IC \subseteq \exists \text{has IO.IO} \).

For each IC a set of IO can be defined. In instructional component strategy four components can be defined as sub classes of the the abstract class Instruction component strategy as follows; Presentation, Demonstration, Activation, and Application (see Figure 5.6). The relations between the knowledge objects and Instructional Components are defined by \( KO \subseteq \exists \text{has IC.IC} \) and \( IC \subseteq \exists \text{is IC.for.KO} \) Such that;

\[
\text{has Instructional_Component} \equiv \text{is Instructional_Component for} \quad \text{for} \\
\text{is IC.for} \equiv \text{is Activation for} \quad \text{for} \\
\text{has kind of} \equiv \text{is kind for} \\
\text{has kind} \equiv \text{is kind for} \\
\]

Such that; \( \text{has Activation} \equiv \text{is Activation for} \)

\( \text{has Application} \equiv \text{is Application for} \)

\( \text{has Demonstration} \equiv \text{is Demonstration for} \)

\( \text{has Presentation} \equiv \text{is Presentation for} \)
has Demonstration ≡ is Demonstration for and

has Presentation ≡ is Presentation for

- Artificial Intelligent
- Problem Solving
- Knowledge & Reasoning
- Planning
- Uncertain Knowledge & Reasoning
- Learning
- Communication, Perceiving and Acting

Figure 5.6 Instruction component strategies of planning concept

- Instruction objects layer:

  \( IO \equiv \{Definition, Example, Exercise, Theory, \ldots\} \), and \( IO \subseteq \exists has LO.LO \).

  For each IC we should define a set of instructional objects (IO). Different IO can be assigned (by the author) for each type of instructional components. For instance, Definition, Example, Exercise, Cased Study, Assignment, Test, Experimental, List, Condition, Action, Theory, Fact, Principle, Proof, Analysis, Description, History, Menimonic, Cased Study, Supporting Information etc. For example the author can assign Definition, or Theory or both for the Presentation component, and Problem, Case Study or Exercise for the Application part. The instructional objects class, defines different instructional objects types that play different instructional roles. Each type considers as a subclass of the abstract class Instructional_objects.
5.4 Course Module Ontology In PANDA.TUTOR

For the Meniomonic class three types of sub classes are defined.

\[ \text{Meniomonic} \equiv \text{Analogy} \sqcup \text{Music Tune} \sqcup \text{First Letter Memimonic}. \]

For Supporting Information class two main subclasses are defined, \text{At Beginning Support Info} and \text{At End Support Info}. These sub classes are defined as abstract class. The first one contains Introduction, Overview and State of Art. For the second class the Conclusion References and Summary can be defined.

Two types of relations exist the instructional component and their Instructional Objects
\[ \text{has Instructional object is Instructional object for respectively. } IC \subseteq \exists \text{has IO IO} \text{ and } IO \subseteq \exists \text{is IO for IC} \]

These object properties also are abstract properties that define the following sub properties

\[ \text{has IO} \equiv \text{has IO for Activation} \sqcup \text{has IO for Application} \sqcup \text{has IO for Demonstration} \sqcup \text{has IO for Presentation} \]

\[ \text{is IO for} \equiv \exists \text{is IO for Activation} \sqcup \exists \text{is IO for Application} \sqcup \exists \text{is IO for Demonstration} \sqcup \exists \text{is IO for Presentation} \]

\text{Such that; } \text{has IO for Activation} \equiv \exists \text{is IO for Activation}^-

\text{has IO for Application} \equiv \exists \text{is IO for Application}^-

\text{has IO for Demonstration} \equiv \exists \text{is IO for Demonstration}^-

\text{has IO for Presentation} \equiv \exists \text{is IO for Presentation}^-

Additinally, according to Task-center-instructional strategy and the first principle of instructional there is a direct relation between the concept and its homework. and Test defined by \text{has Homework} / \text{is Homework for} and \text{has Test} / \text{is Test for} as a relation and its inverse.

For each class of instructional components we define a closure axiom on the property (has IO) for each sub classes (Presentation, Activation, Demonstration and Application). The closure axiom consists of a universal restriction that acts along the property to restrict the type of instruction object type. For example, the closure axiom on the has IO for Application prop-
property is a universal restriction that acts along has_IO_for_Application property, with a set of instruction objects (Problem, Exercise and Case Study).

- **Learning Objects layer**: LO=(LOᵢ, LO is teaching material with different media type)
  
  where media type ∈ text, audio, video, ...

For each type of instructional object, different forms of teaching material or learning object can be assigned. The type of teaching material or learning object types as media type is defined by a class which represents the presentation format. The teaching material class is an abstract class which defines five types of teaching material: Audio, Image, Figure, Text, and Video. For example, text, video or audio can be used to define the concept. The relation has_Teaching_Material_Type and its inverse is_Teaching_material_Type_for is defined to connect between the Instructional_Objects and Teaching_Material_Type.

The separation between representation of domain and learning objects makes it possible to use different kinds of materials for the same domain subject.

Each learning goal is related to one or more domain concepts in the ontology.

Note that, in our educational system we considered different cognitive levels for the course parts, the quantity_level and Quality_Level. The quantity_Level is defined by the author for the course part, while quality_Level is determined according to the student achievement level and will be explained in details during Student Module chapter. Quantity_Level defines four levels of course Content, Advanced_Level, Beginner_Level, Expert_Level, and Intermediate_Level. For the Example and Exercise, different types and levels are defined for each. The Example class different types of Examples are defined: Analogical_Example, Counter_Example, Direct_Example, and Misconception_Example. For the Exercise class, also different Exercise types and Forms are defined: Analogic_Exercise, Description_Exercise, Design_Exercise, Direct_Exercise, Indirect_Exercise
5.4 Course Module Ontology In PANDA.TUTOR


Algorithm 3: Authoring Algorithm

1 Step 1 : Define the course objectives of the course.
2 Step 2 : According to the course learning objectives, add concepts accordingly, and finally get:
   \[ \text{course} = \{ \text{Concept } i \mid i = 1, 2, \cdots m \} \]
3 Step 3 : define different relations between concepts
4 Step 4 : Define the data property Concept of the concept
5 Step 5 : For each knowledge object of \( \text{concept } i = \{ \text{Knowledge.object } j \mid j = 1, 2, \cdots \} \)
6 such that Knowledge.object= (information_about, Part_of, kind_of, how_to, what_happen)
7 if the concept not further has KO then
8   Go to step 3 to define the next concept.
9 else
10   Define the different instruction strategies(IS) for its KO. IS= Presentation, Demonstration,
11      Activation, and Application
12 foreach IS do
13   Define a set of IO. IO= Definition, Example, Exercise, ..
14 foreach IO do
15   Connect it with its learning object
5.5 Authoring Part

Authoring of adaptive Content is one of the most important activities of the course-based adaptive tutoring system. Our goal is to help the author(a non-programmer) to build highly reusable software components that can be employed in a large number of scenarios. Thus, we introduce an ITS authoring system aim to help the teachers to configure different authorable courses with minimum effort. The authoring model allows the author to include different contents in the same concept.

The complexity of authoring intelligent content for a specific domain is the ability of the tutoring system to understand the course content which is defined by the author. The problem here is that the course content to be created for the system should have intelligent content to facilitate dealing with the course generation part which uses the hybrid planning for modeling. The ontology of knowledge objects assists authors in describing a conceptual model of the content structure.

By using our proposed architecture, the author can modify the course structure and incorporate new learning materials without the need for intervention from the system developers. Moreover, there is no need to create or modify the teaching method or strategies for each student. The planning system plays the role of the teacher in our architecture. In other systems the author uses the authoring tool to define the Domain Model, Student Model, and some features of the Control Engine as a rule for the Teaching Model. The author with the course module has two major missions. The first is to define the course structure as knowledge base of the course. The other is to define the Repository by storing all the learning and test materials. The planning system in the pedagogical module helps the domain expert defining and driving the appropriate scenario, teaching strategy, and style according to the student’s personality. The pedagogical module based on the user model selects and orders the course units presented to the user. The existence of a verity of relations between concepts enables the system to generate different plans for teaching of the same educational content. Our course model uses a concept structure as a domain knowledge representation. The au-
5.5 Authoring Part

The author should separate the course into concepts as derived from the concept-mapping. The concepts are represented in a hierarchical fashion, such that the collection of concepts can be constructed from composite concepts, generating a concept hierarchy. Concepts can relate to each other at any level of the hierarchy. The division of the course into concepts gives us the main building block of the course, such that the planner can later put these building blocks together with various sequences generating various presentation. This concept structure is represented by the ontology. The concept structure is performed by the author. The author can add new concepts, delete existing concepts, and change the relationships between the concepts according to his/her design. The author will define his/her course as instances of our structured ontology. That means s/he should consider the architecture of the classes and the different defined relationships (object properties) between classes. The author should also define the learning object for each course part. Moreover, the author can enrich the concept with different learning materials: After defining the concepts structure, different teaching materials (introduction, summary, example, exercise, definition, etc) can be specified by input forms. Additionally graphics, video, and predefined presentations can be uploaded to the LO repository.
Chapter 6

Student Module

The learning process is a cognitive and emotional process that differs from one student to other. One of the main problem in ITS is how the system can understand the student mental behavior and the emotional reaction to manipulate them. Thus, personalization of the educational process needs a certain understanding of the student and the tasks that are important for learning. Therefore, the student module should represent the student’s characteristics which are relevant to the learning process and it should represent the individual information about the student.

Generally, constructing a student model is a complex task for a number of reasons. In principle, it should simulate what a teacher does when she/he evaluates the behavior of the student in order to modify or enhance the student’s effort. We developed a student module that is modeled by an ontology technique, order to personalize the educational system to guide the student’s learning process. The student module in our approach is an adaptive model which monitors the student’s cognitive progress and emotional states. Thus, it can update the learning scenario or concepts knowledge content and decides which should learn or review. The HTN planning system as a pedagogical module responds differently according to the student characteristics and performance.
6.1 The Human Brain

To improve the learning process, it is important to understand the human brain. The human brain is a very complex and still cryptic kind of matter. Virtually, MacLean [140] has modeled the human brain as three regions, neocortex, limbic system and reptilian brain see figure 6.1.

![Diagram of the human brain with labeled regions: Neocortex, Limbic System, and Reptilian Complex.]

**Figure 6.1** The layers of human Brain

- **Neocortex**: It is the place of the main perceptual processes. It is involved in higher functions such as sensory perception, generation of motor commands, spatial reasoning, conscious thought, and in human’s language.

- **Limbic system**: This is the set of brain structures that forms the inner border of the cortex. It appears to be primarily responsible for our emotional life and has a lot to do with the formation of memories.

- **Reptilian brain**: It is located in the brain stem. It was the first part of the modern brain to develop in human evolution. It operates behind the scenes, regulating our survival needs and controlling the body’s vital functions: Processing of food, oxygen, breathing, body temperature and balance, heart rate, blood pressure and reproduction, among many others.
It is worthy to mention that the involvement of emotional brain with the cognitive brain can dramatically improve learning. During the learning process the student should accomplish a variety of tasks such as understanding, memorizing, analyzing, reasoning, and applying. The student’s performance in these learning tasks depend on his/her emotional state.

In the first place, memory plays an important role in the learning process. Memory is defined as a process by which the human brain retains and draws on past experiences to use information in the present. Encoding, storing, and retrieving are the main operational components of memory.

The human memory works on the basis of three different processes. First, information from the external and internal sensory system of the environment is encoded. Second, this information is stored either in the short-term-memory (STM) or long-term-memory (LTM). Finally, the information can be remembered or retrieved. On the whole, there are three basic types of memory: **Short-Term Memory, Long-Term Memory, and Working Memory** [141] as follows:

- **Short Term Memory**: Provides an interface between long term memory and events or objects. Short term memory has a limited capacity. For that, it temporarily stores the received information.

- **Long Term Memory**: Long term memory keeps a large quantity of information for potentially a very long time. Different types of information could be stored in long term memory. The ability to store new information may be infinite in long term memory. There are at least three kinds of long-term memories: procedural, episodic, and semantic as follows:
  
  - **Procedural memory** contains knowledge about what the available operators are, when to perform them, and how to perform them, as well as simple entailments about the situation. It is fast, requires an exact matching and allows for parallel retrievals.
  
  - **Episodic memory** records the previous situations that an agent has been in. It is probably slower, allows for partial matches, and only allows for serial retrievals.
– **Semantic memory** contains facts that the agent knows, but which are not tied to a particular previous situation. It is also slow, allows for partial matches, and only allows for serial retrievals.

During episodic and semantic retrievals, what gets retrieved is determined by the ”goodness” of the match, which may be biased by things like activation (of both cue and memory), noise, emotion and thresholds.

All of these long-term memories are *sources of knowledge* that the agent can use to help choosing and applying operators.

- **Working memory:** The term working memory refers to a brain system that provides temporary storage and manages the required information to carry out complex cognitive tasks as learning, language comprehension, and reasoning. Working memory is involved in the selection, initiation, and termination of information-processing functions such as encoding, storing, and retrieving data. So, working memory has been found to require the simultaneous storage and processing of information. It can be divided into the following three subcomponents:

  – The central executive; it is important in the execution of skills. It is assumed to be an attention-controlling system, and it is particularly susceptible to the effects of Alzheimer’s disease.

  – The visuospatial sketch pad, which manipulates visual images, and

  – The phonological loop, which stores and rehearses speech-based information and is necessary for the acquisition of both native and second-language vocabulary.

The main three activities that are related to memory modeling are: encoding, remembering and forgetting [142]. Information from short term memory is encoded in long term memory through repeated exposure and generalization. Remembering or retrieving information involves recall and
recognition. Forgetting is important and useful for preventing information overload. It is important to note that information is stored through iconic (visual) or echoic (verbal) means. Research shows that verbal information stores itself in the brain much better than iconic (visual) formation. Iconic memory is erasable.

Accordingly, the aim of our system is to improve the cognitive, constructive, motivational, and emotional abilities of the student, regarding the function of each part of the human brain and considering the relations between them.

6.2 The Cognitive Process

Cognitive processes concern the acquisition and representation of knowledge and the relations among objects and facts. Cognition can be considered as a modeling process which creates a model from which deductions can be drawn. The student’s cognitive model should observe the student knowledge in relation to the knowledge domain. The cognitive theory is a learning theory that attempts to explain human behavior psychologically by understanding the thought processes. Besides, its aim is to formalize how students can understand educational material. Social cognitive theory is a subset of the cognitive theory which focuses on the ways in which people can learn to model and appraise the behavior of themselves and others.

In Intelligent Tutoring Systems, a good cognitive model should have the ability to select the appropriate knowledge components, a tailored feedback and hints, the difficulty level of problems with and learning pace that match to individual students, and eventually, improve student learning. A further development of the cognitive model is the constructive model. The constructive model emphasizes the role of the teacher in constructing the student’s view or understanding. It suggests much more activities that the learner can be involved in to help the student creating ("constructing") new meanings.
In the applied research of education and psychology, it is well known that cognitive, motivational, and emotional processes are related to each other and affect on the human world in different ways. In addition, it was found that the cognitive intelligence alone does not guarantee a successful life. Emotions are considered as an important component of human mental states, experiences, and behavioral expressions. Emotions may initiate, terminate, or disrupt information processing. Consequently, Emotional Intelligence (EI) [143] is an important factor in the learning process. Emotional intelligence is localized in the center of future plans concerning school development. EI concerns the ability to perceive, appraise, and express emotion, the ability to access and/or generate feelings when they facilitate thought, the ability to understand emotion and emotional knowledge, and the ability to regulate emotions in order to promote emotional and intellectual growth.

In the next sections we will explain the relation between student personality type and cognition, motivation and emotion in more detail.

6.3 Personality Mood and Emotion

Personality, mood, and emotion are three factors affecting on human interaction. The layered approach which is proposed by Kshirsagar [144], builds a relation between them; such that, personality influence mood and mood influences emotion. The mutual influence on each other is introduced as follows:

- **Personality**, on the highest layer, is a characteristic of a virtual human that distinguishes him from others. Practically, it does not change over time and has an effect on a person’s mood and emotion.

- **Mood** is the second highest layer. It is considered a protracted state of mind resulting from a cumulative effect of emotions, and is influenced by personality and emotional state.
• **Emotion** is analogous to the state of mind that is only momentary. It is influenced by the personality type and mood.

![Appraising Process Diagram](image)

**Figure 6.2** Relations between personality mood and emotion

Appraising process appraises the emotional effects such that these effects act on mood space and emotion space under the influence of personality see figure 6.2. In the next sections we will describe this model in more details and explain the relation between these parameters and learning process.

### 6.4 Student Personality

The term Personality is defined as a permanent pattern of characteristics that discriminates between people in their feeling thought and behavior [145]. A person’s Personality is a constant but reflects the way that s/he can pursue during the appraisal and coping with the incoming events.

A Student’s personality is an important issue in the educational process. It has effects on a student’s cognitive ability, behavior, motivation, emotion, mood, and preferred learning style. Besides, the student interacts with the incoming event according to his/her personality. Thus, students with
different personality types differ in their appraisal and their way of coping. To summarize, personalizing the learning process is necessary for a more efficient process. Although there is progress in the recent years regarding intelligent tutoring systems and the course generation field in general, there is very little attention on the consideration of the student personality, mood, and emotion during the learning process.

Accordingly, the student’s personality is considered as a key factor in our approach. It has different concerns in our approach. Thus, it helps us to expect a student’s response in different situations, foresee the degree of stress, specify their preferring learning style, deduce the motivational goal and define the best coping strategy. Furthermore, the personality of a student will be evidence to the appropriate learning scenario, appraisal, and coping way. In the next subsection we will define the personality types which we considered.

### 6.4.1 Types of personality

There are different models of personality, but the big five factors of personality, or OCEAN model is the most usable one [146]. The five factors are considered to be the dimensions of the personality space. This model is proposed not only for a general understanding of human behavior but also for psychologists treating with personality disorders. This model defines the following factors; Extroversion (E), Neuroticism (N), Conscientiousness (C), Agreeableness (A) and openness (O).

The model states that these five factors form the basis of the personality space; one should be able to represent any personality as a combination of these factors. The test of the big five factors of personality is used to evaluate these factors [146]. These types can be defined as follows;

- **Extraversion (opposed to introversion):** Extroverts are sociable, talkative, assertive, and energetic. They prefer to behave in social situations.

- **Neuroticism (opposed to emotional stability):** Neurotic people anxious, tension, self-pity,
impulsivity, insecurity, emotional, distress, self-consciousness, illogical thinking, depression, low self-esteem, nervousness, and are inclined repetition and anger. Generally, they have the tendency to experience negative thoughts.

- **Conscientiousness (opposed to negligent):** These people are organized, dutiful, methodical, efficient, reliable, persistent, and persevere pursuing their goals. They also think well before making a decision, studying all possible outcome.

- **Openness (oppressed to close):** These people are opened minded, creative, and imaginative.

- **Agreeableness (opposed to antagonistic):** These persons are trusting, cooperative, behave and interact in a kind and friendly way with others and usually place a higher priority on the goals of other’s than on their own.

Note that, in our system we considered the first three factors. Meanwhile, it is worth mentioning the relationship between the Five Factor Model of personality traits and learning. This relation is as follows:

- **Conscientiousness:** this traits is related to a high work discipline, interest in the subject matter, and concentration. Persons with this trait care about their working conditions and have clear goals regarding their studies. Also, they use a strategic approach, are good at organizing and managing their work and time, and work hard in their studies. Usually, they have an intrinsic motivation and positive study attitude [147]. Conscientiousness is related to preference of thought-provoking documents instead of documents which confirm previous ideas and use their effort in information seeking. One central feature of conscientiousness is self-control, with a capacity to carry out tasks. Conscientious people are strong-willed and determined to achieve and academically. The conscientious students are willing to meet this challenge with their usual willingness to work hard. This persistence may prove usefulness in the academic context. The conscientious students were indeed shown to obtain good study
results. Easygoing students with low levels of conscientiousness often chose their information sources on the basis of easy access with the use of minimum effort and thoroughness. These students can be described as easily distracted, careless, impulsive, and hasty.

- **Neuroticism**: This type of personality is linked to the lack of concentration, stress, fear of failure. Moreover, neuroticism is linked to lack of the ability to understand how things relate to each other. This can be linked to the surface learning style. The student with a surface approach concentrates on memorization without any concern of finding a deeper meaning of understanding. They are most concerned on getting through exams and are not really interested in the material itself. Their motivation is extrinsic and they take on a strategic, or syllabus-bound approach to studying. Negative emotions consume energy, distract concentration, and could be an initial obstacle to success. Feelings of anxiety tend to enforce an escaping reaction to a threatening situation where there is a history of failure. The lack of time can be a reality but people may also vary according to how strongly they perceive time pressure and how they act upon it. Research has shown that persons with high levels of neuroticism drop in stressful situations. However, it should be pointed out that temporary states of worry and insecurities can in some circumstances enhance performance. Although strong negative emotions can distract the attention off the actual task, it can also be a means of concentration on familiar tasks and, in this way, in fact enhance performance. Whether there is a basic insecurity about what is relevant or not, new information can appear too challenging. Students who faced relevance problems were shown to prefer documents which confirmed old knowledge to new thought-provoking documents. Students with high levels of neuroticism are more vulnerable to the strain of many conflicting messages and, accordingly, prefer less confusing information. This is a way of increasing the feeling of control and the confidence of sufficient topical knowledge which is particularly relevant to insecure persons. Previous research has shown that the more secure people are, the more
active in information seeking and the more able to accept new information.

- **Extroversion**: Extroversion was related to informal information retrieval as well as preference for thought-provoking documents over documents which confirmed previous ideas. Extroverted students have an enthusiastic, active, and confident character, which is reflected in their information seeking. These energetic and outgoing students want to find much information without being very systematic in their quest for it. Their information strategies were characterized by quick solutions and the use of social abilities. Outgoing students often consulted teachers, supervisors, and friends as information sources. Supervisors and teachers are good sources for direct guidelines and literature suggestions, while fellow students provide the opportunity for informal feedback and the exchange of ideas. Extroversion was in the present study shown to have a negative impact on marks. Extroverted, outgoing students are likely to prefer to devote their time to social activities instead of studies.

Moreover, different relations between personality factors, motivation, mood, and learning style are considered in our approach as we will see in the next subsections. All these dimensions of personality are closely related to the expressional, logical, and emotional personification.

### 6.5 Student Mood

In general, we define mood as a conscious and protracted state of mind that directly controls the emotions and hence facial expressions. It is related to the cognitive appraisal of the person towards events and has a longer effect than emotion.

Mehrabian [148] describes the PAD model of mood, that we considered it in our system. He defines the mood with three factors: Pleasure-displeasure, arousal no arousal, and dominance-submissiveness (or control/control-lack). From these factors he defines eight types of moods: Exuberant (+P+A+D), Bored (-P-A-D), Dependent (+P+A-D), Disdainful (-P-A+D), Relaxed
(+P-A+D), Anxious (-P+A-D), Docile (+P-A-D) and Hostile (-P+A+D). Early, we clearly distinguished between emotion, mood, and personality. Personality causes thoughtful reactions, which in turn causes the mood to change. Some researchers differentiate between emotion and mood as emotion modulates action, while mood modulates cognition and stays longer than emotion. The mood is an extended state of mind as a cumulative state of emotional effects. Mood is also affected by momentary emotions as a cumulative effect. On the contrary, it has effects on the current emotional state of the person. In other words, the reaction between mood and emotion is as follows; first, mood affects the appraisal of events and decides which emotion will be triggered with intensity of emotion. For example, if the person is in an anxious mood, he/she will get more easily disappointed by bad events and with higher intensity of the triggered emotion. Second, emotion can also cause a particular mood to occur. For example a person in a bored mood can become in a more positive mood after some emotional appraisal from the environment.

6.6 Student Emotion

Emotions are much more important to human behavior. Emotions are the reaction to events and objects, and play a powerful and central role in our life. They affect our perception, our attitude, and our relationship to others. Besides, they affect our decisions and have an impact on our actions. As such they are controlling our behavior. Thus, emotions have a strong effect on a person’s behavior, beliefs, and motivations. Izard [149] And Goleman [1] investigate that the emotions effect on a person’s performance. Positive emotions increase the performance and negative emotions decrease it. In addition, they control our reactions.

Emotions are expressed through changes in speech, facial expression, expression and the physiological process. These change the effects on a person’s beliefs, desires, and intentions. Thus, our emotions always influence our beliefs about the cause behind the current emotion. Although
there are many common properties, emotions are not universal processes [150–152]. Some define it as the physiological change, while other treat it as purely mental, for the reason that emotion is found in different areas, such as anthropology, psychologies, and biology. For the learning process, emotions play an important role, and it is an important aspect for student’s intelligence. Students’ emotions have affects on their attention, motivation, performance, personality evaluation [10], and have influence on a human’s cognitive abilities, such as action selection, learning, memory, problem solving [153], decision making process [110], and human communications. According to Goldman, ”the extent to which emotional upsets can interfere with mental life is no news to teachers. Students who are anxious, angry, or depressed don’t learn; people who are taught in these states do not take in information efficiently or deal with it well”. In the learning environment, the teacher tries to maintain a supportive relationship with the learner aiming to serve the development of positive emotions [154]. According to Coles [155], the emotional state can accelerate or perturbing the learning achievements. For these reasons, the tutor’s actions in intelligent tutoring systems should be carefully selected in order to regulate a student’s negative emotional state.

Emotions also influence the efficiency the memory storage and retrieval process. Thus, the learning process involves integration of the cognitive and emotional process. In addition, the learning process involves in particular three cognitive processes, the attention process, memorization process and the reasoning process [156]. These three processes are:

- **Attention**: Is considered a necessary condition for the success of the learning process. It means focus or concentration during the learning process. The emotional state plays a main role in attention. Emotions effect attention as well as the acquisition process. Such that, the student’s attention or emotion is related to the student’s emotions, that can enhance or prevent student’s concentration. Thus, If the emotion is strongly negative (or sometimes positive), it can disturb any kind of attention and the concentration and prevent the student from focusing on a subject. For that reason, negative emotions lead to difficulties in switching the attention
to a new focus [1]. Emotions can be used in the learning content to, in particular, increase the attention of the learner and improve his/her memory capacity. So, the content of the material is learned better if we add some emotions, and we produce the emotion according to the current emotional state of the student. In the next chapter we will discuss how emotional charges can contribute to learning content and how it influences the learner’s performance.

- **Memorization process:** Memory is one of the most important concepts in learning knowledge and is necessary for the major complex cognitive tasks of the learning process. It is a fact that after the learning process, information must be encoded or transformed into a semantic form to facilitate the transformation of information from short-term memory to long-term memory. The memorization process involves the acquisition retrieval, and storage of information. The retrieving process requires remembering and fetching the information from the long term-term memory. Emotions have effects on memorization process. Positive emotions enhance the memory performance in general, and support the transformation, and retrieval processes.

- **Reasoning process:** Reasoning is the development task the memory process. This process enables a student to perform complex tasks for instance; comprehension, classification, application analysis, and problem solving. The positive emotions improve any kind of reasoning as well as relations are built easily between objects or ideas [153]. In other words, the positive emotions improve any kind of reasoning and lead to a more creative, flexible, and divergent thinking process, whereas negative emotions cause more linear, convergent, and sequential thinking [157].

Consequently, the representation of the emotional state is an important factor in ITS architecture in order to define how students apprise the situational states and how they cope with emotion. In other words, the ITS should be able, like any successful human teacher, to emotionalize the
learning process. That can be done by changing the emotional state of the student. This subject will be explained in more detail in the next chapter.

6.6.1 Emotion and Personality

Our personality type has effects on our way to react. While many emotional properties are individual, there is rarely a one-size-fits-all solution for the growing variety of computer users and interactions. It must be emphasized that there is a broad terminology related to the affective states in human beings. Thus, to recognize the student’s emotional reactions to a specific action we take the student’s personality type as a guide to the degree of the student’s reaction in different emotional states. We also ask the student about his/her current emotion and consider the appraisal of this emotion from the student’s point of view.

Romano.et.al [158] consider in their system the relation between the five factors of personality and different types of emotion. They investigate the strong of emotion in the five types of personality. Because we will consider three types of personality (E,C,N) in our system, we will only consider the result of these three types. They investigate that the Anger emotion is positive in both Extraversion and Neuroticism and negative at Conscientiousness. Where, as Disappointment is negative at extraversion. The Relief is negative at Neuroticism, but positive at Extraversion. Distress is positive at Neuroticism and negative at Extraversion. Joy is positive at Conscientiousness and negative at Neuroticism. Fear is negative at Extraversion. Hope is positive at Conscientiousness. Pride is negative at both Conscientiousness and Neuroticism. Love is positive at Conscientiousness. Finally, Hate is negative at Conscientiousness and positive at both Extraversion and Neuroticism.

Our work depends on the student’s emotion. It focuses on relationships between student’s emotional state, mood and personality type, and the tutor’s actions. We adapt system performance according to student state and manipulate the influence of the tutor actions on the student’s emotional state with the aim of improving the quality of the learning process in intelligent tutoring.
6.7 Motivation in Educational Environment

There is an obvious link between motivation and emotion, as emotion is often a precursor of motivation. Also, emotion and motivation influence the way we act towards our environment, while they have different concepts. Parkinson and Colman [159] differentiate between them as, “Emotion and motivation both depend on their relationship between the organism and its environment. The emphasis is on the evaluative aspect of this relationship: In the case of emotion, how the situation makes the person feel; in the case of motivation, it is how the individual acts with respect to the situation that is of interest.”

The term motivation can be defined as a person’s obligation, perseverance, and insistence in doing an action or task. It can also be defined in terms of the individual’s commitment and persistence in choices, plans, and actions. Petri [160] defines motivation as "the concept we use when we describe the forces acting on or within an organism to initiate and direct behavior". He also notices that motivation is used "to explain differences in the intensity of behavior" and "to indicate the direction of behavior". Weiner [9] defines it as ”Motivation is the study of the determinants of thought and action. It addresses why behavior is initiated, persists, and stops, as well as what choices are made.”.

A student’s motivation is an important factor in the education process, and it is a basic concern in the classroom. It is considered as a key component of success in any pedagogical activities. Motivation is responsible in keeping the student interested during the learning process. In other words, motivation can be considered as a hidden hand behind the student effort and defines how an individual’s internal state such as goals, beliefs, and emotions affects behaviors. In fact, students who are motivated to learn will have greater success than those who are not. Additionally, students
who learn well will be more motivated to do so in the future. Emmanuel.et.al [161] define achievement motivation of student as ”Achievement Motivation (AM) is the part of psychology dedicated to the study of motivation to succeed that a learner has and how this motivation can affect learner’s results and behaviors.”.

Different researches showed the importance of considering the student’s motivation during the learning process. However, many e-learning systems don’t care about learner’s motivation. Also, the motivation as an aspect of an instruction is ignored in many ITSs and authoring systems. Computer diagnosis of motivation is usually considered only by few researches.

### 6.7.1 Personality and motivation

There are two types of motivation; intrinsic motivation and extrinsic motivation. People with an intrinsic motivation do the activity for their own aim. Where, as the people with an extrinsic motivation have an external interest in doing the activity, like receiving a prize. If the student has intrinsic motivation for an activity, that increases the learning ability. In that time the regulation style will be interstice, his locus of control will be internal which concludes satisfaction, happiness and concentration. The relation between personality types and learning motivation types is concerned. Komarraju, et [162] define three types of academic achievement: Avoidance, Engagement, and Achievement.

- **Avoidant students** feel discouraged to learn, they worry about failure, and take courses for extrinsic reason. Avoidance was positively with neuroticism and extroversion, and negatively with conscientiousness.

- **Engaged students** enjoy the process of learning, seek knowledge for self-improvement, and enjoy sharing ideas. The extroverted engages in learning more than other types.

- **Achievement students** put in effort to excel and enjoy outperforming others. Achievement
was best with conscientiousness and neuroticism.

### 6.8 Learning Styles

Honey and Mumford [163] define learning styles as the term learning styles is used to describe the attitudes and behaviors that determine an individual’s preferred way of learning. They itemize four styles depending on the Kolb learning styles. The Kolb learning style [164] the so called “Kolb Model” has two bipolar dimensions the active/reflective dimension and the abstract/concrete dimension. These dimensions describe a four-stage cycle of learning. This cycle starts with a Concrete Experience (CE) of an event. The next cycle is Reflective Observation (RO), followed by an Abstract Conceptualization (AC) and then Active Experimentation (AE). Depending on these two axes Kolb defines four learning styles; divergent, as simulators, convergent as accommodators. Based on the learning styles of Kolb, Honey and Mumford [163] defines the following learning styles are: Activist, Reflector, Theorist and pragmatist. These types are as follows;

- **Active learning styles** The active student likes to act first and consider the involvement afterward, likes new ideas and becomes bored in implementation or repetition, but motivated by experimentation and challenging tasks, and she/he prefers activity-oriented learning materials with a high interactivity level.

- **Reflective** Students like to deliberate their experiences and study the situation from different perspectives. Thus, they tend to collect and analyze data before taking action. Those students prefer example-oriented learning material.

- **Theorists** Try to formulate their experiences in theoretical or logical form, motivated by trying ideas, theories, and experiments. They prefer exercise-oriented learning material.

- **Pragmatists** Prefer to explore and discover concepts on a more abstract level. They are eager
6.9 Student Control

to try out, but concentrate on the concept that can help them achieve their task. Therefore, the theory-oriented learning materials are convenient to this type.

6.8.1 Personality and Learning Styles

The primary principle for an individualized learning process is that no single instruction strategies are best for all students. Assessing students learning styles provides awareness of their particular preferences. Many studies investigated the relation between personality and learning and teaching styles. Furnham [165] links between the big five of personality and learning styles. He found that extroverts have a learning style comparable to that of the activists and pragmatists. On the other hand introverts are like reflectors regarding their learning style. The Neuroticism is more probably theorists and reflector, and Conscientiousness interacted positively with Activists style and negatively with theorist.

6.9 Student Control

Schnackenberg, et. al. [166] investigated the importance of considering student control during the learning process and differentiated between a student’s control and system control. The student’s control can help students to become independent students and make choices according to their preferences. On the other side, some critics advocate that student control could disturb students, because it forces them to interrupt their learning and concentrate on the sequencing of material [166]. Others [30] find that the novice students are unable to make the right sequence choices, because they lack the background knowledge necessarily to make educated decisions. Therefore, it is important for the novices to benefit from system control [167]. Some researchers find that the degree of learner-control should vary depending on the students’ familiarity with the topic as well as the students’ motivation, aptitude, and attitude [168]. It also depends on the personality type of
the student, as we explained before.

On the whole, we can conclude that the two approaches, student control and system control can be considered according to student state, but at this point, the research regarding a balance between these two types, such that, the system should adapt itself according to the student cognitive level considering both types [169].

6.10 Student Module Ontology in PANDA.TUTOR

The personalization feature of e-Learning systems requires the system to be adaptive regarding needs of the student. In order to establish an e-learning system with personalization capability, we need to define the student module, which is considered as one of the most important concepts for personalization in e-learning system. The student module provides individual information about the student, analyzes the student’s behavior and updates the student information according to the student’s performance. The student module is considered to be a leader for the system to generate adequate plan for the student’s state in PANDA.TUTOR. Likewise, the individualization of the student module is essential for adapting the learning process, which helps the planning system to behave differently with different users and draw a decision and generate a convincing lesson. Thus, the pedagogical module defines the ideal behavior that fits the behavior and the characteristic of the student. The semantic web technology enables systems to be more adaptive and intelligent with the capability of reasoning any related data. In each application of the semantic web, the ontology holds an indispensable role for its successful use. The student module in PADA.TUTOR is represented by an ontology. The ontology offers the opportunity to model all information about the student. As in the Course Module, the Protege version 4.1 framework has been selected to edit our ontology.

The Main class in our model is the student class and two main classes are defined and connected
to student class; **Static Information**, which contains the student’s information that is not altered during the interaction with the system, and **Dynamic Information**; it includes information that is updated during the learning process according to the progress achieved by the student, as we will see in the next sub sections.

### 6.10.1 Static Information

The static model consists of two subclasses: student identification and personality type.

- **Student Identification**: $Student \subseteq \exists has_{St.ID}$
  
  contains the basic information about the student. For instance: Student ID, student Name, Address, Gender, Birth dates, E-mail, etc.

- The **Student Personality Type**: $Student \subseteq \exists has_{Personality.type}$ defines the traits and the individual characteristics of student’s. The personality type is defined by the test of big five personalities. are defined as a subclasses based on three types: Extroversion, Neuroticism; Conscientiousness )these types are: Spectator type (low $E$, low $N$, low $C$), Insecure type (low $E$, high $N$, low $C$), Sceptic type (low $E$, low $N$, high $C$), Brooder type (low $E$, high $N$, high $C$), Hedonist type (high $E$, low $N$, low $C$), Impulsive type (high $E$, high $N$, low $C$), Entrepreneur type (high $E$, low $N$, high $C$), Complicated type (high $E$, high $N$, high $C$)

### 6.10.2 Dynamic Information

Many aspects of the learner are constantly changing. Hence, the system builds an individual model for each student and consequently updates it, in order to keep an updated state according to the student’s progress. The Dynamic class has the following subclasses;
• **Student Cognitive Level:** \( \text{Student} \subseteq \exists \text{has\_cognitive\_level} \)

The cognitive level is an important factor. In general, it defines the student level in the course contents. In our system we consider two types of cognitive level as subclasses; These properties are abstract properties which define the following sub properties respectively:

\[ \text{has\_cognitive\_level} \sqsupseteq \text{has\_quantity\_level} \sqcup \text{has\_quality\_level} \]

- **Student Quantity Level:** \( \text{Student} \subseteq \exists \text{has\_quantity\_level} \)

The quantity levels determines the depth of student’s knowledge. We defin three quantity levels (Beginner, Intermediate, Advanced).

- **Student Quality Level:** \( \text{Student} \subseteq \exists \text{has\_quality\_level} \)

The quality level defines the goodness of student in the course concepts. Five levels are defined for quality level (Low, Good, Very Good, and Excellent).

When a new user logs into the system, the system asks the student if s/he prefers to define his/her quantity and quality level or to take a general test, to determine approximately the level of the student.

• **Student Performance History:** \( \text{Student} \subseteq \exists \text{has\_personality\_history} \)

This module stores the history of the student’s interaction with the system and defines the student state and the cognitive level for each concept and for each teaching material. Thus, the student’s performance state connects between the student identification and course ontology to define the student state in each course part. We define the following subclasses; concept state (selected, not selected), learning object state (visited, not visited), answer state(right, wrong, help request, I don’t know, give up, partly answer), homework(no answer, low, good, v.good, excellent), help state(requested, accepted, rejected). In addition, different levels of Concept\_performance levels are defined as axioms (ExcellentPerformance, VeryGoodPerformance, GoodPerfromance, LoePerformance), such that these axioms justify their decision
according to the performance of the homework that is related to the concept.

- **Student Emotion**: $Student \subseteq \exists has_{\text{Emotional\_state}}$
  
  In this class we use the FEASP- approach [170] which introduces five major dimensions of emotions that are relevant to the learning process. The emotions are as follows (Fear, Envy, Anger, Sympathy, Pleasure).

- **Student Emotional History**: $Student \subseteq \exists has_{\text{Emotional\_history}}$
  
  It stores the last emotional state to ask student about.

- **Student Motivation**: $Student \subseteq \exists has_{\text{Motivational\_state}}$
  
  In this class we consider the following subclasses:

  - **Confidence**: The detection of the student’s confidence level relies mostly on the students’ beliefs on their efficacy to perform the instructional task. Accomplishing the task raises the learner’s expectancies of future success. Whereas, failures affect one’s sense of self efficiency and decrease the level of confidence. Less confident learners are likely to avoid tasks perceived as difficult, or give up before attempting to perform a task. So, as will be explained in the next chapter, we will consider the student self appraisal or belief. Hence, three levels of confidence are defined (high, average, low).

  - **Effort**: is measured by the student’s actions, such as the number of attempts to solve a problem, help requests, etc. In other words, it refers to how this result was achieved (did it require much or little work). Three levels of effort are defined (high, average, low).

  - **Independence**: relates to the perceived feeling of needing or not needing the tutor’s help to accomplish the instructor task. Personal control over the learning process should be perceived by the learner as actual control over his/her own success in accomplishing
desired goals. In this sense, the students "independence model" relates to the perceived feeling of needing or not needing the tutor’s help to accomplish the instructional task.

- **Student Goal:** \( \text{Student} \sqsubseteq \exists \text{has\_student\_goal} \)

  defines the student’s purpose: It is the answer to the question of what students want to achieve by learning the course. There are two kinds of goals; the direct goal which is defined directly by student (to learn a specific concept with a specific scenario); the motivational goal which is defined by the planner during the planning according to student state (increase confidence, decrease sadness), as we will explain in the next chapters.

Note that in addition to this information there is other information that is defined and modeled in the planning system according to the personality type of the student as we explained in this chapter. that we did not consider these in our student’s ontology; This Information is:

- **Learning Style:** \( \text{Student} \sqsubseteq \exists \text{has\_student\_style} \)

  we explained in this chapter we consider four types of learning styles; (Active, Reflective, Theorists, Pragmatist)

- **Strong of emotion:** \( \text{Student} \sqsubseteq \exists \text{has\_Emotional\_state} \)

  strength of the emotion is determined according to the personality type as we explained before.

- **Mood:** \( \text{Student} \sqsubseteq \exists \text{has\_Mood} \)

  we will consider PAD mood model, which is based on three factors, Pleasure-displeasure, arousal-no arousal, and dominance-submissiveness (or control/control-lack), eight types of moods are defined; Exuberant (+P+A+D), Bored (-P-A-D), Dependent (+P+A-D), Disdainful (-P-A+D), Relaxed (+P-A+D), Anxious (-P+A-D), Docile (+P-A-D) and Hostile (-P+A+D).

  Thus, according to the appraisal process in PANDA.TUTOR the mood state will be induced.
Chapter 7

Emotional Intelligence and Apprising Process

Emotional intelligence (EI) is considered to be an important factor for increasing student performance. Student performance is increased according to student’s emotion and motivation. The idea of emotional intelligence is that some individuals are better at having the ability to reason about emotions and interacting emotionally to enhance thoughts more effectively than others. Giving emotional intelligence to ITS means enabling the system to define a student’s emotional reaction and react with the student according to his/her appraisal. In this chapter we will clarify the concept of emotional reasoning and how we can improve the emotional intelligence of a student to overcome this problem and teach the student how to be emotionally intelligent. Moreover, we will explain the appraisal process.
7.1 Why we focus on Emotional Intelligence in PANDA TUTOR?

The intelligent tutoring system is a computer based educational system that provides individualized instructions. The traditional ITS which is based on pedagogical state decides how and what to learn. It is important for ITS to consider the emotional aspects of student as a human tutor, to manage and motivate the emotional state besides the pedagogical state for improving the learning process. Emotional Intelligence as it was defined earlier, is the ability to recognize, interpret, and influence someone’s emotions.

Emotional intelligence (EI) refers to an ability to recognize the meanings of emotion with its relationships, and the capacity to reason about emotions to enhance thinking; to reflectively regulate emotions, and promote emotional and intellectual growth. In other words, EI concerns the ability to perceive control and evaluate emotions. It has the ability to identify, use, understand, and manage a person’s emotion in positive and constructive ways. The concept of EI is based on the relation between “thinking“ and “feeling“. Hence emotions are linked to the cognitive process. It is a fact that our emotions help us to make the decisions.

Since 1990, Peter Salovey and John D. Mayer [171] have been the leading researchers of emotional intelligence. In their influential article ”Emotional Intelligence,” they defined emotional intelligence as ”the subset of social intelligence that involves the ability to monitor one’s own and others’ feelings and emotions, to discriminate among them and to use this information to guide one’s thinking and actions”. The concept of emotional intelligence was popularized through the publication of psychologist Daniel Coleman [1], an American psychologist. His work has been to study the efficiency of EI in the world of business for increasing personal effectiveness; it also has been substantial in the educational field, because it is offering ways of improving students’ achievement and providing them with skills for their personal and working lives.
The persons who are reconsidered ‘intellectually’ intelligent are not necessarily emotionally intelligent. Thus, having a good memory, or good problem-solving abilities, does not mean you are capable of dealing with emotions or motivating yourself. The highly intelligent people may lack the social skills that are associated with high emotional intelligence. Scientists, who show beyond belief intellectual abilities in narrow fields, are an extreme example of this: a mathematical genius may be unable to relate to people socially. However, high intellectual intelligence, combined with low emotional intelligence, is relatively rare and a person can be both intellectually and emotionally intelligent. On the other side, a person with a high emotional intelligence is capable of understanding the feelings of others. Culturally, they are better at handling relationships of every kind.

EI is not only concerned with understanding your own emotional state, but also the emotional states of others, and being able to use this information to induce the best outcome for all concerned. Thus, it is important for EI to know where emotions come from and know what emotions mean and what information they provide. For the purpose of being able to manage the emotional state for the person itself or others, being able to work well with others as well as alone, and being able to combine cognitive knowledge with emotional knowledge. All in all, emotional intelligence helps us to build strong relationships, have success at work, and achieve our goals. The skills of emotional intelligence can be developed throughout life.

The research shows that intellectual intelligence contributes only about 20% to success in life. The remaining 80% of success depends on our emotional intelligence [1]. It is important to know also that we can develop our EI by upgrading our emotional skills. In contrast, we usually train our intellectual abilities but rarely, if ever, train our emotional intelligence in a structured way. In fact, educating the heart is as importance as educating the mind. Most educational systems considered improving our IQ factor but not the EQ factor. It is important for the tutoring system not only to understand the cognitive level of the student but also the emotional status. In most of our educa-
tional systems, we are not taught how to handle frustration anxiety, stress, failure, or depression problems during learning process. Also, we are not told to learn how to manage emotions through e.g., interaction, coordination, adjustment, or communication. On the other hand, at a later stages of our lives we are told to master emotional competencies to be successful. But unfortunately, it is a fact that not all people have the ability to regulate their own emotional reactions. For that reason the proposed system has the ability to teach students how to define their emotion and how to adapt their emotional state.

Thus, intelligent tutoring systems should integrate emotional intelligence for improving the student’s performance. Such that, giving this ability to the computing system means, enabling it to determine a person’s emotional reactions capabilities. When the effect of feeling or thinking is known, this can be used in order to improve someone’s cognitive abilities. Accordingly, the system should understand the student’s emotion to be able to deal with it. On the other side, the system should interact with the student in an emotional way, which will be explained in the next sections. Whereas, the existing course generation systems lack consideration of the emotional conditions of learning that should exist in order to motivate the student and facilitate the achieving of the learning process. In this thesis we propose a way to detect and manage the student’s emotions in order to improve the learning process and the emotional conditions of the student according to the cognitive state of student.

### 7.1.1 The Emotional Intelligence Branches

In 1990, after various psychologists, psychiatrists, and evolutionary biologists had identified a number of human capacities involved in identifying and understand emotions, during the late 1980’s, Drs. Mayer and Salovey proposed that these abilities make up the collective concept of emotional intelligence. Salovey and Mayer proposed a model that identified four different factors of emotional intelligence [172]: the perception of emotion, the ability to reason using emotion-
7.1 Why we focus on Emotional Intelligence in PANDA TUTOR?

The four branch model of emotional intelligence describes four areas of capabilities, or skills that define emotional intelligence. These branches are as follows:

1. **Perceiving Emotions**
   It provides the starting point for a more advanced understanding of emotions. It concerns the ability to accurately identify and express feelings. In many cases, this might involve understanding nonverbal signals such as body language and facial expressions or asking the user about his/her emotional state.

2. **Using Emotion to Facilitate Thought**
   It concerns the ability to use our feelings constructively, the ability to employ emotion in guiding cognitive processes and promoting thinking, the ability to use our feelings for helping our decisions, or the ability to let our feelings guiding us to what is important to think about. A good system of emotional input helps in the consideration of these important matters.

3. **Understanding Emotions**
It concerns the ability to understand the meanings of emotions and how they can be changed. The emotions that we perceive can carry a wide variety of meanings. Emotions impart information. Therefore understanding emotional messages and the actions associated with them is central to this skill. Emotional understanding also includes the ability to understand the reason of emotions; the relationship between thoughts and feelings, the student’s emotional effectiveness, the relationships between emotions, and how and why they can change from one feeling to another.

4. Managing Emotions
The ability to manage emotions effectively is a key part of emotional intelligence. It concerns the ability to turn negative emotions to a positive one, the ability to help others identifying and benefit from their emotions, and the ability to regulate emotions. Responding appropriately and responding to the emotions of others are all important aspects of emotional management. For instance, we can enhance our EI by learning how to rapidly reduce stress, connect to our emotions, communicate nonverbally, use humor plan to deal with challenges, and defuse conflicts with confidence and self-assurance.

7.2 How to Improve the Emotional Intelligence

It is worth pointing out, that emotional intelligence can be learned and developed [1]. Meanwhile, the intellectual understanding of emotional intelligence is very important, but substantial; the development of it depends on sensory, non-verbal learning, and much practice. But what exactly might such skills be? Daniel Goleman introduced The Emotional Intelligence Competencies of High Achievers [1]. He gives this list:

- **Self-awareness**
  One of the basic emotional skills involves being able to recognize feelings and to put a
name on them. It is important to be aware of the relationship between thoughts and feelings that was behind that action, and knowing one’s internal states, preferences, resources, and intuitions. Also, it is important to recognize our emotions and their effects, the way of reacting with our environment, and the affect of our emotion on our behavior.

- **Managing emotions and Self-Regulation**
  
  Being able to change our emotions to a positive end is a key ability. We should practice controlling impulsive feelings and behaviors, dealing with our emotions in healthy ways; being reliable and responsible; learn to adapt to change our situations. It is important to realize what is behind feelings. Beliefs have a fundamental effect on the ability to act and on how things are done. For instance, it is important to find ways to deal with anger, fear, anxiety, and sadness

- **Self-motivation**
  
  Self-motivation means how emotional dispositions can guide or facilitate reaching goals. Thus, when we have a goal, the control of emotions will assist greatly in achieving it. Thus, it
is important to learn how can we motivate ourselves in general. That includes endeavoring to improve or meet a standard of excellence, and persistence in pursuing goals despite obstacles and setbacks.

- **Empathy**

  The ability to recognize feelings of others is important if we are intended to establish good relationships with them. Getting the measure of the situation and being able to act appropriately requires understanding the feelings of the others and being able to take their perspective. It is important to be able to listen to others, sensing others’ feelings and perspectives, and taking an active interest in their concerns.

- **Social Skills**

  Developing quality relationships has a very positive effect on all involved. This includes the ability to listen openly and to send convincing messages, to negotiate and resolve disagreements, initiate or manage change, to work with others toward shared goals. A good understanding of emotions can help us to manage the emotions of others. Recognizing the value of the contribution of others and encouraging their participation can often do more good than giving orders or complaining.

According to Salovey and Mayer, there is a set of conceptual related mental processes involving emotional information [172]. The mental processes include;

1. Appraising and expressing emotion in one self and others.
2. Regulating emotion in one self and other and,

In the following sections we will explain these processes in detail and will show how we modeled these processes in our pedagogical module, with the explanation of the emotional intelligence
7.2 How to Improve the Emotional Intelligence

The first step towards improving emotional intelligence is preserving or electing emotion. Understanding or recognizing emotions of other is related to perception factors that can occur by visual, auditory, and tactical. Intuitively, the lack of this data causes misunderstanding of a person’s emotional state. Understanding emotion and its intensity is also related to getting to know a person’s personality type. Note that we should differentiate between hot and cold emotions to justify the process of the election process. Hot emotions are that they influence behavior and that they are not under a complete selection control of the person as a result of the primary emotion. The primary emotion defines what is felt first, as a first response to the situation. If we are threatened, we may feel fear. When we learn of a death, we may feel sad. They are unthinking, instinctive responses that we have. There are several ways to evaluate the emotional state, Lang [173] proposed analyzing emotions according to three systems involved in the expressions of emotions: Subjective or verbal information (i.e. reports about perceived emotions described by users), Behavioral (i.e. some are based on the detection of facial, speech paralinguistic parameters, postural expressions and biolog-

Figure 7.3 Conceptualization of emotional intelligence

7.2.1 Emotion Election

The first step towards improving emotional intelligence is preserving or electing emotion. Understanding or recognizing emotions of other is related to perception factors that can occur by visual, auditory, and tactical. Intuitively, the lack of this data causes misunderstanding of a person’s emotional state. Understanding emotion and its intensity is also related to getting to know a person’s personality type. Note that we should differentiate between hot and cold emotions to justify the process of the election process. Hot emotions are that they influence behavior and that they are not under a complete selection control of the person as a result of the primary emotion. The primary emotion defines what is felt first, as a first response to the situation. If we are threatened, we may feel fear. When we learn of a death, we may feel sad. They are unthinking, instinctive responses that we have. There are several ways to evaluate the emotional state, Lang [173] proposed analyzing emotions according to three systems involved in the expressions of emotions: Subjective or verbal information (i.e. reports about perceived emotions described by users), Behavioral (i.e. some are based on the detection of facial, speech paralinguistic parameters, postural expressions and biolog-
Chapter 7 Emotional Intelligence and Appraising Process

...inal signs [174]), psycho-physiological answers (such as heart rate, galvanic skin response -GSR-, and electroencephalographic response), the others are based on student interaction [175], or the use of the cognitive model of emotion [8]. Cold emotions describe a dispassionate cognitive process that may be appropriate for reasoning about emotion but do not actually inform on the external behavior. The cold theory of emotion or cognitive model of emotion [8] allows us to reason about the emotional responses or the appraisal of the current emotion towards the incoming events as a result of the secondary emotion. Thus, secondary emotions appear after primary emotions. They may also come from more complex chains of thinking. The secondary emotions build on the primary emotions and involve more thought and interpretation and they give a picture of the person’s mental processing of the primary emotion. The proposed model is inconsistent with the hot theory of emotion but not with the cold type of emotion.

7.3 Emotional Reasoning

The term Emotional Reasoning refers to the thinking error that occurs when a person believes that his/her feeling is true regardless of the evidence.

The thinking error basically involves an individual wrongly believing that what they are feeling must also be a fact. However, this is often recognized as being unhelpful in terms of one’s mental health and well-being. Thus, it prevents one looking at alternative, more balanced informations or evidence supporting the contrary. For example a depressed person can feel that she/he is worthless and bad. If they are able to understand that these feelings are merely symptoms of depression then everything is fine. But, if they use emotional reasoning to misinterpret their feelings of being bad, that will be evidence that they must have done something awful. Another example, an anxious from test might feel that he does not understand the material. He might feel as though he does not understand at all, but in fact is perfectly capable of answering the questions, and is merely insecure.
about it. By acting on the basis of his insecurity, he might assume he does not know the answers and guess randomly. Thus he creates a self-fulfilling prophecy of failure. In this way, emotional reasoning amplifies the effects of other cognitive distortions.

Appraisal to an optimistic one for improving the emotional intelligence of the student. Learning to identify thinking errors such as the emotional reasoning is half of the battle. In our approach, we will use the appraising processes to identify the student’s appraisal and determine is that error thinking or not. Note that according to this appraisal and personality type, the student will close a reaction following his/her coping way. In this time the role of the system is to regulate the emotional state of student and try to understand student’s new reappraisal, for the aim of changing the student’s appraisal to an optimistic one and to improve the emotional intelligence of student.

7.4 Apprising and Coping Theory

7.4.1 Appraising Process

The cognitive theory organizes our behavior around two basic processes: appraisal and coping. The appraisal process considers the relationship between a person and their environment. It assumes that the event has no meaning in itself, but has a significance corresponding to the person’s evaluation or interpretation [176]. Likewise, Marsella and Gratch [177] assume a relation between the event and goal achievement, such that the emotional effects of the event can facilitate or inhibit a person’s goal. The appraisal process generates emotions by assessing a person environment relationship (did the event facilitate or inhibit the student’s goals). There are two levels of appraisal process; the primary appraisal and the secondary appraisal [113]. The primary appraisal concerns the evaluation of the person’s impression and motivation towards events presuming four kinds of effects; harmful, threatening, challenging, and benign. The secondary appraisal depends on the evaluation of the event’s controllability and ability to alter the stressful situation. Thus, the people
are motivated to respond to events differently depending on how they are appraised [178]. Thus, if the event is appraised as undesirable but as controllable, that motivates people to develop and execute plans to reverse these conditions. On the other hand, if the event is appraised as uncontrollable, it leads people towards escapism or avoidance.

On the other side, the coping process concerns how can we deal with the current emotion, either externally (by forming intentions to act in the world) or internally (by changing the person’s interpretation of the situation).

So, when the effect of the appraising process is known, the system should interact with the student to improve the cognitive abilities. Therefore, the system needs to achieve the following conditions: knowing the current emotional state of the learner, and determining the impact of an action on a learner’s emotional state and the educational goal to improve his/her performance by selecting the most advantageous emotional state and learning scenario for the student.

The work on cognitive emotion is based on the psychological theory of [8, 113, 174]. Smith and Lazarus’ introduced the cognitive-motivational-emotive approach, as illustrated in figure 7.3.

**Figure 7.4** The cognitive motivational emotive system
Note that the appraisal theory considers how events influence the other and use this appraisal to guide the person action (for example if a person believes that his action has harmed another, he may move to undo his actions). For this reason, we will consider the appraisal of the students as their beliefs, and use the student’s appraisal to regulate their emotion considering their coping ways according to a student’s personality type. In the next section we will explain the theories that we considered during appraisal and coping process in PANDA.TUTOR.

**OCC model**

The OCC model describes a person’s emotions as a result of a person’s appraisal. The OCC model is one of the most known emotional models. It was developed by Ortony et al [8]. The OCC model defines the hierarchy classification of 22 emotions. These emotions concern the appraisal of events in the environment and their affects on the internal emotional state. In the OCC appraisal model, emotional states represent the relations between possible causes and emotional states. In this theory, emotion arises from valence (positive or negative) reactions to the concerns of the user in an environment. The situations are divided into events (desired states of the world), actors (ideas about how people should act) and objects (likes and dislikes). The valence of one’s emotional relation depends upon the desirability of the situation for oneself, which is in turn defined by one’s goals and preferences.

In this work we used the OCC cognitive model of emotion which manipulates the affective state as a cognitive appraisal between cause and situation. Thus, we induce a student’s emotional states by assessing the student’s appraisal before giving the course, in the light of a student’s personality and goal. In addition we considered three factors from the five factor models of personality traits to define the strong of this emotion as we explained in the previous chapter. Thus, the use of the emotions specified by the OCC model will facilitate the integration of a dialog module that can generate emotions depending upon the semantics and context, as we will see in the next chapter.
Table 7.1 The emotion type specifications of the OCC model.

<table>
<thead>
<tr>
<th>Emotion Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joy</td>
<td>(pleased about) a desirable event.</td>
</tr>
<tr>
<td>Distress</td>
<td>(displeased about) an undesirable event.</td>
</tr>
<tr>
<td>Happy-for</td>
<td>(pleased about) an event presumed to be desirable for someone else.</td>
</tr>
<tr>
<td>Pity</td>
<td>(displeased about) an event presumed to be undesirable for someone else.</td>
</tr>
<tr>
<td>Gloating</td>
<td>(pleased about) an event presumed to be undesirable for someone else.</td>
</tr>
<tr>
<td>Resentment</td>
<td>(displeased about) an event presumed to be desirable for someone else.</td>
</tr>
<tr>
<td>Hope</td>
<td>(pleased about) the prospect of a desirable event.</td>
</tr>
<tr>
<td>Fear</td>
<td>(displeased about) the prospect of an undesirable event.</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>(pleased about) the confirmation of the prospect of a desirable event.</td>
</tr>
<tr>
<td>Fears-confirmed</td>
<td>(displeased about) the confirmation of the prospect of an undesirable event.</td>
</tr>
<tr>
<td>Relief</td>
<td>(pleased about) the dis-confirmation of the prospect of an undesirable event.</td>
</tr>
<tr>
<td>Disappointment</td>
<td>(displeased about) the dis-confirmation of the prospect of a desirable event.</td>
</tr>
<tr>
<td>Pride</td>
<td>(approving of) one’s own praiseworthy action.</td>
</tr>
<tr>
<td>Shame</td>
<td>(disapproving of) one’s own blameworthy action.</td>
</tr>
<tr>
<td>Admiration</td>
<td>(approving of) someone else’s praiseworthy action.</td>
</tr>
<tr>
<td>Reproach</td>
<td>(disapproving of) someone else’s blameworthy action.</td>
</tr>
<tr>
<td>Gratification</td>
<td>(approving of) one’s own praiseworthy action and (being pleased about)</td>
</tr>
<tr>
<td></td>
<td>the related desirable event.</td>
</tr>
<tr>
<td>Remorse</td>
<td>(disapproving of) one’s own blameworthy action and (being displeased about)</td>
</tr>
<tr>
<td></td>
<td>the related undesirable event.</td>
</tr>
<tr>
<td>Gratitude</td>
<td>(approving of) someone else’s praiseworthy action and (being pleased about)</td>
</tr>
<tr>
<td></td>
<td>the related desirable event.</td>
</tr>
<tr>
<td>Anger</td>
<td>(disapproving of) someone else’s blameworthy action and (being displeased about)</td>
</tr>
<tr>
<td></td>
<td>the related undesirable event.</td>
</tr>
<tr>
<td>Love</td>
<td>(liking) an appealing object.</td>
</tr>
<tr>
<td>Hate</td>
<td>(disliking) an unappealing object.</td>
</tr>
</tbody>
</table>
7.4 Apprising and Coping Theory

**Figure 7.5** The original structure of emotions of the OCC model

**Attribution Theory**

The attribution theory analyzes the student’s beliefs about the causes of success or failure and conjectures the effect of different emotions. Weiner [9] classified attribution using three dimensions: locus of the cause (internal/external), stability of the cause (stable/unstable) and controllability of the cause (controllable/uncontrollable). The locus of control dimension can be external or internal depending on how the cause of success (or failure) is attributed by the student. The stability dimension determines if the cause is changed over time or not. The controllability dimension makes a distinction between controllable attribution causes like skills/effort and uncontrollable ones like luck. In other words, attribution theory is concerned how a student explains success and failure.
Students explain their success or failure based on the assignment of the reason as internal or external to them. This reason could be stable and controllable. For example, the student may attribute a low grade in a test to his/her low ability (see Figure 7.6).

**Figure 7.6 Attribution theory**

The student’s ability is considered to be an internal attribute, and it could be interpreted by a student as a stable and uncontrollable factor (however it can be interpreted as an unstable and controllable factor by other students). But if the student backs the reason to his/her low effort, the case will be changed because effort is considered as internal, unstable, and controllable factor. The emotional case will be changed if the student attributes the failure to the difficulty of exam, which is considered as external, unstable and uncontrollable factor.

Attribution theory proposes that the achievement outcomes immediately produce emotions such as happiness after success and frustration after failure, “attribution-independent”. These emotions do not need any more cognitive elaboration. In contrast, ”attribution-dependent” are emotions which are shaped by the causal attribution of the outcome; such that, success or failure cause the emotional outcome to be aroused. For example, success is assumed to induce joy and failure induces frustration or sadness. Non-occurrence of the expected success is thought to arouse disappointment, in the other side non-occurrence of the expected failure arouses relief emotion.
Pride attributes the success to internal factors (such as ability). Shame and guilt which are induced by failure are described as internal factors that are subjectively controllable factor(such as lack of effort or ability). Gratitude is expected to arouse the attribution of success to external factors that are under control of others (e.g., help), and anger attributes the failure to such factors.

In line with Lazarus’s and Weiner’s theories, empirical research has corroborated the subjective indicators about the lack of controllability. For instance, low self-confidence in ability and failure expectancies, are related to students’ anxiety. Thus, students could translate the feeling of success or failure to emotions such as pride, shame, guilt, or anger.

All in all, attribution theory is concerned with appraising the achievement outcomes and how it can determine the future of achievement strivings. It gives us at least one assumption regarding motivation. The instructor should make an effort to help students attribute their learning outcomes to determine the problem. The instructor should try to alter the student’s beliefs. For instance the change of beliefs from uncontrollable to controllable and from stable to unstable could help students build their confidence in them selves. Without that, the student will have no motivation to participate in the learning process without belief that the change is possible. For instance, when a student believes in his abilities, it increases the student’s confidence. However, if the student does not believe in his abilities, it indicates that he/she unconfident in his / her ability. But if the student believes that the problem is related to his or her effort that could encourage the student to exert more effort. Likewise in this theory, Weiner explains when the teacher should blame the students and when he should have sympathy with them, which is considered during the dialog and course generation plan process in our system.

**Control theory**

Control theory investigates emotions regarding achievements. It focuses on student’s emotions during the learning process, as well as after (success or failure). It uses three parameters; posi-
Chapter 7 Emotional Intelligence and Apprising Process

tive/negative emotion, active/deactivate motivation, and student’s control or the ability of student to overcome on the learning difficulty [179].

The control-value theory addresses emotions related to activity such as enjoyment and boredom experienced during the learning process, and relates among student’s appraisal, control motivation and emotion. Thus, if the student categorizes his appraisal as a controlled factor, that plays an important role in determining a student’s emotion and helps to arise his motivation or activation to achieve the learning goal. We can observe that these parameters are identical with the parameters of PDA mood model that we described in the previous chapter. Taking all together, emotion identifies the current emotional state, but when we reason about the cause of this emotion and investigate that this reason has long effect on student’s emotional state, it will be considered as the student mood. Also we can consider mood as a combination of emotion and motivation state. For example, anxiety is seen to be induced when the outcome of an exam is perceived as not being sufficiently controllable, but subjectively important. Conversely, if the student feels in control and does not expect failure or does not care about an exam, there is no need to be anxious. Similarly, enjoyment of learning is seen to be instigated if a student feels competent to master the material and values the material. If the student feels incompetent or is disinterested, negative activity emotions such as boredom are induced rather than enjoyment.

For the relation between emotion and motivation, they investigate that anxiety reduces students’ interest and intrinsic motivation, but it can motivate students to do extra effort to avoid failure. Also, anger and shame, have a negative effect on learning, but this is also dependent on the student’s beliefs. Thus, if the students believe in their ability, failure can motivate students to increase their effort to overcome this situation. Boredom and hopelessness are two emotions that have negative effect on student motivation and behavior, because these emotions usually propagate from the student experiencing a lack of self-assurance regarding his/her ability. Positive emotions like enjoyment of learning, have a positive effect on learning achievement.
7.4.2 Coping process

Coping determines how one responds to the appraised significance of events. Thus, coping process operates on the same representation of a person’s appraisals, beliefs, goals, and plans, but in the reverse direction, seeking to make a change, directly or indirectly, that would have the desired impact on appraisal. Furthermore, the agent might form intentions to change external factors. In other words, coping process substantially is the inversed process of appraisal. This process tries to alter the event or the reason directly or indirectly.

There are two types of coping strategies; problem coping strategies and emotional coping strategies. The coping process which endeavors to change the environment of the reason behind negative emotion follows the problem coping strategy, but when the coping process aims to change the interpretation or beliefs of this relationships, it pursues the emotion coping strategy. The controllability of the reasoning also determines the type of coping strategy, such that if the event is evaluated as undesired but controllable then problem focused coping is selected. But if the event is evaluated as undesired and uncontrollable then emotion focused coping is adequate. Personality influences this process through stable biases in how an individual appraises and copes with events. Folkman et al. [177, 180] define eight types of coping strategies in an example of coping strategies is as follows:

1. **Problem focused Coping:**

   - **Active coping or Confronted coping:** taking active steps to trying to remove or circumvent the stressor.
   - **Planful problem-solving:** Thinking about how to cope with the coming action.
   - **Seeking social support:** Talking to someone who could do something, seeking advice or assistance.
2. Emotion focused Coping:

- **Suppression of competing activities**: Put other projects aside or let them slide.
- **Restraint coping**: Waiting until the appropriate opportunity arrives.
- **Seeking social support for emotional reasons**: getting moral support, sympathy, or understanding.
- **Positive reappraisal**: Look for silver lining or try to find a new faith;
- **Acceptance**: accept stress or as real. Learn to live with it
- **Turning to religion**: Pray, put trust in god (assume God has a plan)
- **Denial**: Denying the reality of event
- **Distancing**: The person went on as if nothing had happened and/or reduce the effort.
- **Escape or avoidance**: Wish that the situation would go away or somehow be over with. Or Use other activities to take mind off problem: daydreaming, sleeping
- **Self-control**: The person tries to keep feelings to himself);
- **Accepting responsibility or self blame**: Criticize or lecture myself);

However, there is a relation between the personality type and coping as we will explain in the next section.

### 7.5 Personality and Coping Process

The big five tests are used to evaluate these factors. Three types are respected in our approach Extroversion (E), Neuroticism (N), and Conscientiousness(C). From these three types of personality as we defined before, eight combinations of high and low of the big five factors (26) are defined.
These eight types focus on the degree of stress and their coping way (problem or emotion strategy). These types of personality are defined as follows [181].

- **Spectator** ($E_-, N_-, C_-$): This type has low responsiveness to other people or to situational cues, is emotionally flat, not very interested in social norms, and low in ambition in his or her work.

- **Insecure** ($E_-, N_+, C_-$): These persons are self-conscious, dependent on other people’s opinions, overly sensitive to his or her own mental and physical experiences, and experience frequent negative emotions.

- **Sceptic type** ($E_-, N_-, C_+$): This type of personality is relatively closed in relations to others, self-secure, emotionally stable, and effective in managing his or her life, but sometimes somewhat rigid.

- **Brooder type** ($E_-, N_+, C_+$): These persons are shy and withdrawn, ambivalent, insecure, and scrupulous, brooding over every decision and tending to give up easily when meeting difficulties.

- **The Hedonist type** ($E_+, N_-, C_-$): is a socially skilled, pleasure-oriented, emotionally and physically robust, but also undependable person.

- **The Impulsive type** ($E_+, N_+, C_-$): is pleasure-oriented, attention seeking, and in need of social confirmation, has little control over his/her emotional reactions, and appears chaotic and changing.

- **Entrepreneur type** ($E_+, N_-, C_+$): is a socially secure, independently thinking, domineering, cool-headed, goal-oriented and effective person.

- **Complicated type** ($E_+, N_+, C_+$): is emotionally intense with occasional emotional outbursts and subsequent guilt-feelings, sensitive, dependent on others, conscientious and orderly.
Neuroticism and Extroversion are related to stress, and conscientiousness is related to the ability of coping. They investigate that the highly neurotic person becomes more stressful and changes easy to a negative emotion, distress and copes in inactive and maladaptive way. Whereas the high Extroversion person becomes also more stressful in events but changes easy to positive emotion, and coping in active way and seek in social support. Finally, the persons with high conscientiousness cope in an active and planned way. The connection between personality and coping makes understanding the coping of the students easier, and helps teachers to teach [182–184]. We must meet the needs of each student in our framework according to his/her coping style. Note that, some student need adaptation in their coping style as we will see in the following:

- **Neuroticism (N)** Those with a high N are characterized by a tendency towards negative emotions, such as anxiety, depression or sadness, hostility, and self-consciousness, as well as a tendency to be impulsive. Also, they have been found to report greater distress when faced with either work or home overload, or when faced with interpersonal stress, than those with low N. For the coping way, the findings of several studies suggest that those who are high on N are less likely to engage in problem-focused coping. Instead, they tend to rely on emotion-focused forms of coping, particularly ones that involve escape-avoidance and accepting responsibility (self-blame). They are less planful problem solving than those lower on N. Also, those higher on N exhibit a greater tendency to engage in confrontation in coping with stressors in both close interpersonal and work contexts.

In stressful interpersonal situations involving someone close, those higher on N employed less empathic response than those lower on N. But those higher on N are not lacking the ability to be empathic, but they find it more difficult to engage in these processes when a stressful situation involves someone close.

However, those higher on N tend to cope in ways that may be inappropriate for the demands of the situation. Both escape-avoidance and confrontation have been found to be associated
with a variety of negative outcomes across a number of studies.

- **Extraversion (E).** Those high on E tend to experience positive emotions and to be warm, gregarious, fun-loving, and assertive. They are positively connected with happiness and pride, and negatively with stress, fear, and self-disgust. Studies examining associations between E and coping have found that those high on E engage in higher levels of problem-focused coping than those low on E. Those high on E also engage in less avoidance and other maladaptive forms of emotion-focused coping. Thus, they tend to engage in more support seeking, positive thinking, restraint, planful problem solving, less self-blame, wishful thinking, and avoidance and positive reappraisal than those lower on E.

- **Conscientiousness (C).** Those high on C have been characterized as having a tendency to be habitually careful, reliable, hard-working, well-organized, thorough, diligent, self-disciplined, dependable, and achievement-oriented and purposeful. Those high on C reported high use of problem-focused coping and low use of emotion-focused coping. Compared to those lower on C, those higher on C reported relatively more empathic responding and relatively less self-blaming and escape-avoidance. The higher on C displayed a greater tendency to engage in planful problem solving to cope with a genetic stressors than those lower on C. Taken together, these results may help explain previous research indicating that work performance is predicted by the trait of conscientiousness.

For the combination between these factors, the person with low extroversion and high neuroticism (the insecure and broody types) has a high level of job stress. Conversely, the person with high extroversion and low Neuroticism (the hedonist and entrepreneur types) has low level of job stress. The person with complicated typefaces also has a high level of job stress, and has an ability to control (or high level of independency), but the person with low Extraversion, low neuroticism and
high conscientiousness has a low level of control (or high level of dependency). In general the person with high level of N and low C reported more stress, whereas types combining low N and high E or high C reported the least stress. For coping, the combination between high E with high C is related to higher use of problem-focused coping, whereas the combination of high N and low C is related to the reduction of the use of the problem-focused-coping strategies. On the other side, the combination between high E with high C showed the most use of the emotion-focused coping, whereas the types that combine low E and low C showed the least problem-focused coping.

### 7.6 Emotion Regulation

In the past two decades, psychological research has started to focus more on emotion regulation. Regulating the learner’s emotional status is one of the main requirements in an e-learning system. Emotion regulation strives to increase positive emotions and to decrease negative emotions. Basic components of regulation are recognition and understanding of one’s own emotions, managing these emotions by inducing, modulating, or preventing them, and using emotions for action and goal attainment [185].

Gross [186, 187] describes the model of emotional regulation using the following definition: “Emotion regulation includes all of the conscious and non-conscious strategies we use to increase, maintain, or decrease one or more components of an emotional response”. Thus, humans use different strategies to affect their level of emotional response for a given type of emotion, to prevent a person from having too high or too low emotional response level.

Accordingly, there are two main types of emotional regulation strategies; antecedent-focused strategies and response-focused strategies. Antecedent-focused strategies are applied to the process preparing for response tendencies before they are fully activated. In his model, Gross distinguishes among four different types of antecedent-focused emotion regulation strategies, which can be
applied at different points in the process of emotion generation: situation selection, situation modification, attentional deployment, and cognitive change. In the other hand, the Response-focused strategies are applied during the activation of the actual emotional response, when an emotion is already underway. Response modulation is a response-focused strategy. 7.7 shows an overview of these strategies.

The first antecedent-focused emotion regulation strategy in the model is situation selection: a person selects a situation that matches the emotional response level the person wants to have for a certain emotion. The second strategy in the model is situation modification. When this strategy is applied, a person modifies an existing situation so as to obtain a different level of emotion.

The third strategy is attentional deployment. This strategy refers to shifting your attention to a certain aspect. The fourth strategy is cognitive change for selecting a cognitive meaning to an event. A specific type of cognitive change is aimed at down-regulating emotion, is reappraisal: 'Reappraisal means that the individual reappraises or cognitively re-evaluates a potentially emotion-eliciting situation in terms that decrease its emotional impact'. However, note that cognitive change could also be aimed at up-regulating emotion.

![Figure 7.7 Emotion regulation model by Gross](image-url)
Table 7.2 The emotion type specifications of the OCC model.

<table>
<thead>
<tr>
<th>Strategy Corresponding Element</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>situation selection</td>
<td>Situation</td>
</tr>
<tr>
<td>situation modification</td>
<td>Sub-situation</td>
</tr>
<tr>
<td>Attentional deployment</td>
<td>aspect</td>
</tr>
<tr>
<td>cognitive change</td>
<td>meaning</td>
</tr>
</tbody>
</table>

The second type of emotion regulation strategy, *response modulation*, a response-focused strategy, is applied after the emotion response tendencies have been generated: a person tries to affect the process of response tendencies becoming a behavioral response. A specific type of response modulation, again aimed at down-regulating, is *suppression*: 'Suppression means that an individual inhibits ongoing expressive behavior.'

To sum up, the situation selection strategy affects the situation in general by trying to select the appropriate one. The situation modification strategy affects sub-situations by trying to adapt the conditions of the current situation. Attention modification deployment tries to change the attention and thinking of the person from the current event to the next by changing the aspect of the aspect of the situation. Finally, by using the cognitive change strategy, we try to reappraise the student appraisal of the situation by changing the student’s beliefs. Our system tries to find the best regulation for the current emotional state that can enhance the learning process.

For emotional regulation we will consider different emotional and motivational strategies as we will explain in the next sections.
7.7 Emotional Regulation Strategies

Most of the instruction design strategies have neglected emotions because they could interfere with the achievement of important cognitive or motivational objectives. Rarely do we find such strategies in the literature on instructional design and tutoring systems. Astleinter and Leuter [188] introduce instruction strategies for making instructional technology more emotionally sound. These strategies can help instructional designers to develop emotionally sound web-based instructions. These instructional strategies increase positive emotions and decrease negative ones. The positive emotions relate to sympathy and pleasure. The negative emotions relate to fear, envy, and anger. The five major dimensions of emotions are identified as (FEASP) approach; Fear which triggers in response to judged a situation to be treating; Envy, which comes from the desire to either get or not lose something; Anger which comes in response to the encounter of obstacles while trying to reach a goal; Sympathy, which is experienced in response to people in need of help; and Pleasure which is experienced when mastering a situation and achieving a goal. According to these emotions five types of emotional instruction strategies are defined; Fear Reduction, Envy Reduction, Anger Reduction, Sympathy Increase and Pleasure Increase. By understanding these five basic emotional conditions instructional designers will be able to use relevant strategies to produce emotionally sound web-based instruction. The prospective proposition points out how instructional technology should be designed to get the desired emotional outcomes from the students. In our system these strategies are considered during the course generation process; such that, they are modeled within the planning domain model.

7.8 Appraisal Coping and Regulation in PANDA.TUROR

The appraisal theory considers how events influence each other and uses this appraisal to guide the person action. The appraisal process allows us to assess the performance of the learner in his
current emotional state. For this reason, we will consider the appraisal of the students and their beliefs, and use the student’s appraisal to regulate their emotion considering their coping ways according to the student’s personality type.

![Diagram of the appraisal and coping theory](image)

**Figure 7.8** Appraisal and coping theory in our approach

The learning appraiser module asks the student about the current emotion and the reason behind this emotion as student beliefs, to measure the knowledge retention in the student’s mind. Then, the system uses the personality types to induce the strength of emotion and the coping way. Note that we are not usually coping with the situation or event by the right way. Everyone copes with an incoming event according to his/her appraising way and depending on the personality type of the person. Accordingly we will enhance the good coping way and adapt or regulate the bad way, to teach the student how to be emotionally intelligent. In other words, in our system we analyze the impact of positive and negative emotions on student performance, check whether the reason behind this emotion is internally or externally, and try to increase/decrease the internal emotion by increase/decrease the external one. As in traditional learning environments in which, the system attempts to select the best conditions for learning and especially the emotional conditions. For
example, the teacher could tell a joke in the classroom in order to change the current emotional states of the students, if they are bored or frustrated. Because, the student is bored, anxious, angry or depressed, she could not learn or think efficiently.

7.9 Emotional Ontology

To address this requirement, we are developing an Emotion Ontology (EMO). Requirements in computing and artificial intelligence have led to the development of ontology-like resources for emotions. Affective computing aims to integrate emotional responses into computer interfaces in order to produce more realistic systems which are able to respond to the emotional communication of their users. To facilitate affective computing, Lopez et al. propose a slim ontology schema for describing emotions in human-computer interfaces [189]. Also motivated by affective computing requirements, the W3C emotion markup language (EML, http://www.w3.org/TR/emotionml/) is an XML-based standard for markup of emotions in text or databases. The proposal presented in this work is based on a generic ontology for describing emotions and their detection and expression systems taking contextual and multimodal elements into account. The ontology is proposed as a way to develop a formal model that can be easily computerized. Moreover, it is based on a standard, the Web Ontology Language (OWL), which also makes ontologies easily shareable and extensible. Once formalized as an ontology, the knowledge about emotions is used in order to make computers more accessible, personalized and adapted to user needs.

Our model is based on the appraisal and coping theory. The produced ontology serves as a basis for the representation of emotions and it is stored in the student affective model. First, each affected event, has encent which could change the person emotion. Four types of effects are considered as a primary appraisal (harmful, threatening, challenging, and benign). Second, according to the personality type of the person and the type of effect of the incoming event the
person appraises the event and triggers his/her emotion. Finally, the triggered emotion affects on a person’s behavior. However, the person behavior is affected by the person personality. Thus, the persons cope differently with the incoming events according to a person’s personality type.

The ovals represented classes in the ontology, while the arcs represent the relationships between these classes. In our system we consider this ontology to register an emotional state history for each student, which is used during the interaction with the student.
Chapter 8

Affective Planning

Personalization and adaptation in the educational process are important factors for providing an effective educational service on the Internet. We emphasize the fact that students perceive and process information in a very different way. The primary principle of individualized learning is that no single instructional strategy is best for all students. Therefore, it is necessary to define the model taking the whole context that surrounds the user into account. This chapter describes PANDA.TUTOR, a model of Affective Intelligent Tutoring system that illustrates how a planning system supports a better treatment of emotional reasoning. In this model we introduce two types of planning modeling; dialog based planning and lesson generation.

8.1 PANDA.TUTOR

The process of arranging personalized adaptations is usually complex because people with different personalities cope with the problem by different ways. The current platforms usually do not provide more than a relatively simple way of personalization and adaptation. Also, many adaptive hypermedia systems have been created without considering personality types and different learning objects. For instance, [104, 190] have developed an adaptive system to the student that adjusts the
difficulty of the delivered exercises. Furthermore, most of the previous approaches are based on a model of generic student’s emotional reactions. Also, coping and regulation have been ignored in the course generation field.

It is a fact when the system appears to care about the student’s state, this may make the learning process more fun and help the student enjoy their learning process. In order to respond to the student, the system should consider the student’s emotion and personality correctly. Thus, if the student has a specific negative emotion and if we not consider the student’s state, this can have side effects on his/her learning state. We should try to change the negative emotion and motivation to a positive one, or maintain the positive one. The foundation of emotional plan has been inspired by [191]. He represents emotional plan as a plan-based appraisal based on assessing the incoming event to goal dedication. Afterward, plan-based appraisal is ameliorated by considering appraisal and coping theory [192]. Different systems consider the emotional planning during human-computer interaction and conversation [177], [193], and some consideration for appraisal and coping theory during the learning process [4, 5]. In this chapter we introduce a personalized planning system called PANDA.TUTOR, in which the system deals with students in different ways, according to their personality type, emotional and motivational state of student, aiming to improve the student’s emotional intelligence as well as intellectual intelligence. Moreover, the system tries to change the student’s emotional reasoning to increase the student’s internal motivation. Thus, the system attempts to put the student in a good and motivated particular mood that could be more receptive to the material being taught. For instance, the system could treat mistakes as less important if the student is going through a particular bad emotion. As consequence, students will be able to achieve learning goals more efficiently when pedagogical procedures are adapted to their individual differences.

Indeed, the system is able to predict the emotional reaction according to a student’s appraisal for each specific personality types in various situations. That could help us to predict the student be-
havior, which help us to correct the weakness in each personality type and strong the positive side. For example, suppose we have two personality type, the first one being a complicated student (nervous, conscientiousness and extroversion) and the second one being an impulsive student(nervous, conscientiousness, and introversion). The extroversion side in the first type could help him/her to be less depressed than the introversion side in the second one. Thus, if the student fails or gets a low score in the exam, the second type will blame his/her self, and lose his/her confidence easier than the first one, while the first type could be more nervous than the second one at the first impression. Therefore, the system tries to reduce the level of nervousness in the personality character, or increase the conscientiousness or extroversion (as social factor)to increase the confidence level in student character. In contrast, the system could also try to help the high level of extroversion student to be more concentrated. That can be done by the emotional regulation. For Instance, by changing the student’s appraisal or understanding of the situation, (as ”consider the white side ”or ”see the glass as half full and think you’ll come through this difficult time”), being empathetic with the very sensitive person, or blaming the careless student and teaching him how to understand or value his/her responsibility.

The system could focus on different aspects of instruction depending on personality characteristics of the student. Thus, PANDA.TUTOR has the ability to adapt the learning environment taking into consideration the improvement of the emotional state through adaptive educational systems in the semantic web environment. We integrate different theories of instruction , which are guided by the student model. That means our system can generate different presentation for the generated lessons from different scenarios according to the student’s personality and states.

Hybrid planning is used for modeling the the pedagogical module. The hybrid planning provides the generated plans with a hierarchical, temporal relationships and causal dependencies between tasks on both abstract and primitive levels. Hybrid planning can be used for synthesizing flexible and adapted plans. In this chapter we will explain how we used the hybrid planning approach in
modeling the dialog and course generation.

8.2 Pedagogical Module

The Pedagogical module makes decisions about the student model. The decisions are related to the selection of an appropriate topic for the student, selection of an appropriate learning method, curriculum sequencing, etc. through the course module.

It is a fact that learning environments that don’t consider motivational and emotional factors are not adequate. So, we should simulate a teacher’s teaching way and analyze the students’ emotional state to give a proper regulation for adjusting their negative emotions. In our system we aim to enhance the intellectual intelligence of student as well as the emotional intelligence of student. Therefore, the pedagogical module in PANDA.TUTOR has two phases; the Dialog phase, and the Lesson generation phase.

The goal of the Dialog phase is to enhance the emotional intelligence of the student by teaching the student how to appraise his or her emotional state, by defining the current emotion and the reason behind this emotion. Note that, we ask the student about his/her emotion and the reason

**Figure 8.1 Phases of pedagogical module**
behind this emotion. We ask about long term emotion or the mood of the student. In other words, it is not like emotions that can be recognize from facial expression, which could be the momentary feeling or reaction. We are interested in emotions that could have effects on the physiological state of the student and that could have an effect on the student’s behavior.

During this phase the system will establish a dialog with the student to collect the student’s appraisal. The dialog actions come in different forms, based on personality type. Note that, the emotional state of the tutor (as a teacher) is modeled as well, including values for emotions and parameters such as satisfaction, disappointment, sympathy, and surprise.

While the Lesson Generation phase is considered as the core of the whole system, in which the system generates a lesson for the selected concept and regulate the emotional and motivational state of the student considering the appraisal result of the Dialog phase. Two processes are defined over the selected concept in this phase; scenario selection and lesson generation. The emotional and motivational goals that are defined during the Emotional Appraisal phase as well as the personality type of the student guide the planning system to select the convenience scenario and structured lesson.

## 8.3 Emotional Intelligence and Appraisal process in PANDA.TUTOR

During this process, we teach the student how to be emotionally intelligent. As we explained in the previous chapter, the first process is to teach the student to identify his or her emotional state. The second process is to understand the student’s emotion by defining the reason behind this emotion. The third process is a managing the emotion. The system will try to enhance the positive one and reduce the negative one. This process is modeled as a Dialog based planning in PANDA.TUTOR. The dialog based planning is used as a diagnostic model to define the emotional state of the student and determine the motivational and emotional goal and to grasp the information necessary for the
appraisal process. The emotional and motivational goals are used afterwards in the learning phase in the planning problem, to generate the appropriate plan for the student and to archive the fourth process of IE, "Using Emotion to Facilitate Thought”.

In PANDA.TUTOR we considered the Goleman’s competencies as follows;

- **Self-awareness:** The students can learn how to recognize their feelings and name them as well as recognize the relationship between thoughts or beliefs and feelings, and the effect of these emotions on the student’s behavior.

- **Managing emotions and Self-Regulation:** In which the student can learn how to change their emotions to a positive end. We should practice controlling impulsive feelings and behavior, dealing with our emotions in healthy ways; being reliable and responsible; learn to adapt situations. For instance, changing our beliefs has a fundamental effect on our ability to deal with anger, fear, anxiety, and sadness.

- **Self-motivation:** The system uses emotional regulation strategies to guide students in achieving the goal. Also, the system teaches students how to be self motivational in general. That includes endeavoring to improve a standard of excellence, and persistence in pursuing goals, despite obstacles.

- **Empathy:** The system should have the ability to interact with the students’ feelings if we are intending to establish good relationships with them. Getting the measure of a situation and being able to act appropriately requires understanding the feelings of others involved and being able to take their perspective or beliefs.

- **Social Skills:** Good understanding of emotions can help us manage the emotions of students. Developing quality relationships has a very positive effect on all involved. This includes the ability of sending convincing messages, negotiating and resolving disagreements, initiating or managing change, working with students toward their goals.
The planning system models the appraising process with the emotional significance of events that relate to the learning process considering different reasons (figure 8.2) and learning goals, and predicts the emotional state for each personality type. It is worth mentioning that the system considers the personality types during the dialog. For instance, if the student is nervous we should avoid blaming him or avoid interacting with him in an aggressive way. Instead, we should try to down him. The system also generates different dialogs for each personality type considering the appraisal of the student and his/her emotional state.

By using the hybrid planning mechanism we can benefit from the hierarchical of plan in considering different emotions and different interpretations, and benefit from POCL in considering the causality between event and emotion development and also between the student’s appraisal and the appropriate way to regulate the situation and student personality.

Note that recognizing the appraisal of the student helps us to change the appraisal of the student and consequently the emotion of student. Hence, our emotional state influences our beliefs about information available in our environment. Thus, we could change the student’s beliefs regarding the reason of his/her failure from low ability to low effort or low concentration. For instance, if the event is appraised as bad but controllable by the student, it could motive student to develop and execute plans that reverse these circumstances. While, if the event is appraised as uncontrollable, it could lead to denial, ignoring or depressed condition. With each case the system will generate a different plan to overcome the obstacles that could be executed in student mind. Accordingly we can define the motivation and emotional goals for each case that will be considered by the planning system during course generation. Note that the performance of the system as a tutoring system should change from a student to student and according to the current situation and personality type of the student. In the next section we will define the appraisal variables that we considered in the appraisal process.
Figure 8.2 Reason of emotion

8.3.1 Appraisal variables

The appraisal theory stresses the cognitive view of emotions: emotions are elicited by appraising events and situations. We should teach the student to appraise his or her emotion, knowing the name of emotion and knowing the reason behind this emotion. Accordingly, the system can also appraise the real emotion of the student accordingly to personality type. As we explained before in the previous chapter, during the appraisal process we considered the OCC cognitive model of emotion, attribution theory and control theory. Based on these theories we considered the following appraisal ingredients:

- **Personality types:** Retrieved from student module.
• **Student Emotion:** Ask about the current emotion as good or bad only.

• **Emotion Reason:** Ask about emotional reason (*we considered the reason in figure 8.2*)

• **Occurrence:** The event is checked if it happened or will happen in the future. Accordingly, the following emotions can be induced: Hope or fear concern a desired/undesired event will happen. While, Satisfaction or Fears-confirmed concern a desired/undesired event happened.

• **Self consideration:** Check if the event occurred to the student or someone else. Accordingly, the following emotion can be induced joy or distress concern (if the event occurred to student his/herself). While, happy-for, pity, gloating, resentment or envy concern (if the event occurred to someone else)

• **Eligibility:** Check if the student evaluates the event as joy or distress.

• **Controllability:** The reason is checked if it is appraised by the student as controllable/uncontrollable.

• **Motivation state:** Check if the student is motivated or not.

• **Commendation:** The student is asked if someone praise or reprove him/her. Accordingly, the following emotion can be induced; pride or shame relate (if someone praiseworthy/blameworthy his/her action). While, admiration or reproach relate (if student praiseworthy/blameworthy the other action)

• **Stability:** The reason is checked if it is stable/unstable.

• **locus of reason:** The reason is checked if it is internal/external.

According to the reason of emotion the system analysis the **locus** of the reason (internal/external), **stability state** (stable/unstable) and **controllability** (controllable/uncontrollable). Also determine Goal **pertinence** to recognize if the event affects the learning goal or not, by considering the reason
of the current emotion. Finally, by considering the personality type we can determine the confidence level of the student. Please note that we teach the students how to define their emotion. In our system we analyze the impact of positive and negative emotions on student performance, check whether the reason behind this emotion is internally or externally, and try to increase/decrease the internal emotion by increasing/decreasing the external one.

8.4 Dialog Planning in PANDA.TUTOR

Our system represents how we can use the hybrid planning technique to represent the dialog based emotional intelligence principle.

Using HTN planning is more appropriate to generate the dialog for a number of reasons. First, the decomposition of HTN planning is better suited to the type of large-scale dialog planning. Second, the hierarchical decomposition minimizes the search time. Third, the hierarchy information is also useful in determining the appropriate way to refine the corresponding abstract task. However, By using the hybrid planning mechanism we can benefit from the hierarchical of plan in considering different emotion and different interpretation, and benefit from the POCL in considering the causality between event and emotion development and also between the between the student appraisal and the appropriate way to cope with the situation and the student personality).

Form the previous review, we can note that all the previous planning systems generate a set of primitive tasks as a result of the interaction with the user (or as advices to the user like Travel by train, Travel car, use your bike). And they used the re-planning technique when the user refuses the decision of the plan or if something in the current state is changed. We propose a dialog planning system to improve the emotional intelligence of student and to grasp the information necessary for apprising process. The Planner creates its plan by simulating the effects of the operator actions on a model of the student. In PANDA.TUTOR dialog, which is based on the HTN planning tech-
nique, the student interacts with the predicate(s) (i.e. the effects of the generated primitive tasks). The planner initially generates ad hoc dialog controllers which support acquisition of specific user preferences via dialog. The system operates through several cycles of planning and dialog. To this end, in each cycle a set of predicates is added as new facts in the initial state of the given planning problem (see Figure 8.3).

![Figure 8.3 Dialog based HTN-planning in PANDA.TUTOR](image)

The dialog planning will start with the planning problem which includes in its initial state these facts: Student_Info that has two parameters ?student_id such as 101ID and ?student_name such as Sara; Student_Personality that has two parameters ?student_id such as 101ID and ?Personality_Type such as Brooder; and Student_Cognitive that has three parameters ?student_id such as 101ID, ?quantity(Level) such as Intermediate ?quality(Level) such as Very−good. The initial plan in this planning problem contains one abstract task Panda_dialog.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<problem domainModel="atDomainModel" name="atProblem">
  <initialStateDescription>

```
Chapter 8 Affective Planning

The main abstract task "Panda.Dialog" can be refined by eight decomposition methods according to the student’s personality types. In each decomposition method the dialog will ask the student about his/her emotion and the reason behind this emotion. The first step for improving the emotional intelligence is preserving or electing the emotion. As we explained in the previous section, the first process is to teach the student to identify his or her emotional state. The second process is understanding emotion by defining the reason behind this emotion.

For example, in the first cycle, the following plan (a set of operators) will be generated to the student. In which, the student can interact with the effects of these operators. Note that, what will be presented to the student are only the effects of these operators. As in the following example, a set of questions will be presented to ask about his emotion and the reason behind this emotion (such as question in $T_1$ and $T_2$).

$T_1$: Check_current_emotion

Pre: student_info(st_id?, St_name?)

eff: Welcom_how_are_you_today(st_name)

Are_you_Angry(Y_N)

Are_you_Happy(Y_N)
8.4 Dialog Planning in PANDA.TUTOR

Are_you_Afraid(Y_N)
Are_you_Annoy(Y_N)
Are_you_Sad(Y_N)
Are_you_Normal(Y_N)

T2: Check_the_reason
Pre: true

eff: What_is_the_reason(st_name)
Course_reason(Y_N)
Exam_reason(Y_N)
Your_friend(s)(Y_N)
other_reason(Y_N)

Thus, the planning problem is refined through various cycles of planning and dialog according to the interaction process between the system and the student. The dialog tries to manage and regulate the student’s emotion. The system will try to enhance the positive one and reduce the negative one.

With the Proposing that the student answered thesequires as follows

Welcome_how_are_you_today(Heba)
Are_you_Angry(N)
Are_you_Happy(Y)
Are_you_Afraid(N)
Are_you_Annoy(N)
Are_you_Sad(N)
Are_you_Normal(N)
What_is_the_reason(Heba)

Course_reason(N)
Exam_reason(Y)
Your_Teacher(N)
Your_friend(s)(N)
other_reason(N)

these parameters with the student’s answer will consider as initial state to the new problem.

<?xml version="1.0" encoding="UTF-8"?>
<problem domainModel="atDomainModel" name="atProblem">
  <initialStateDescription>
    <fact name="Are_you_Angry">
      <constant name="No" sort="no" type="rigid"/>
    </fact>
    <fact name="Are_you_Happy">
      <constant name="yes" sort="yes" type="rigid"/>
    </fact>
    <fact name="Are_you_annoy">
      <constant name="No" sort="no" type="rigid"/>
    </fact>
    <fact name="Are_you_sad">
      <constant name="No" sort="no" type="rigid"/>
    </fact>
    <fact name="Are_you_normal">
      <constant name="No" sort="no" type="rigid"/>
    </fact>
  </initialStateDescription>
</problem>
The following plan will be generated

xT2: check_happy_exam_confirmation

T2: Pre: Exam_reason(Y)
    Are_you_Happy(Y)
    eff: Are_you_happy_for_finishing_the_exam(Y_N)
    Are_you_happy_for_getting_high_score_in_exam(Y_N)

Then a new problem will be generated as follows

<?xml version="1.0" encoding="UTF-8" ?>
<problem domainModel="atDomainModel" name="atProblem">
    <initialStateDescription>
        <fact name="Are_you_happy_for_getting_high_score_in_exam">
            <constant name="Yes" sort="yes" type="rigid" />
        </fact>
    </initialStateDescription>
</problem>
Then a new plan will be generated

T1: congratulate_student
   pre: not call_massage(congratulation_m)
   eff: call_massage(congratulation_m)
       Store_Student_confidence(high)

T2: encourage_student_to_study
   pre: not call_massage(encourage_m)
   eff: call_massage(encourage_m)

T2: Check_the_exam_result
   Pre: Are_you_happy_for_finishing_the_exam(Y)
   eff: Have_you_solved_good(Y_N)
       Have_you_got_good_score(Y_N)
       Is_the_Exam_canceled(Y_N)

T2: check_student_appraising
   Pre: true
   eff: What_do_you_think_about_the_reason_()
       You_have_done_good_effort(Y_N)
       Exam_was_easy_ (Y_N)
Your_ability(Y_N)

For instance, after several cycles of dialog the output could be as follows. Accordingly this dialog will be given to the student:

```
"Congratulations I am happy for you"
"You are clever student continue"

Have_you_solved_good(Y_N)
Have_you_got_good_score(Y_N)
Is_the_Exam_canceled(N_N)
What_do_you_think_about_the_reason_( )
You_have_done_good_effort(Y_N)
Exam_was_easy(Y_N)
Your_ability(Y_N)
```

During the dialog phases, the appraisal ingredients are considered. At the end, the dialog produces a set of predicates as diagnosis facts about the student. The course generation uses these facts in its planning problem to generate an appropriate course to student. These predicates contain the emotional and motivational states of the student. Accordingly, different motivational and emotional goals are considered, for instance; Increase Confidence, Increase Effort, Increase Happiness, Decrease Fear (see examples 8.4 and 8.4).

Note that, recognizing the student self appraisal helps us to change the appraisal of the student and consequently the emotion of the student. Hence, our emotional state influences our beliefs about information available in our environment. There are several theoretical perspectives on the relationships between emotion and cognition. Thus, we could alter student beliefs about the reason for failure from low ability to low effort or low concentration. In this way we can help students to recognize their emotions and connect between their emotion and the reason behind this
emotion, which will help them to overcome their problem. Also, the student’s appraisal helps us to address the student’s emotional reasoning when it occurs by comparing the student’s beliefs and the performance history in the student module. Accordingly, the system establishes a dialog to manipulate the thinking error. For instance, if the student does not believe in his/her ability, however has a good quality level, then the dialog will try to change student’s beliefs, increase his/her self confidence and encourage student to achieve learning tasks. It is worth mentioning, the system considers the personality types during the dialog. Thus, the personality type plays an important role in understanding the student, and determining the appropriate scenario of the dialog. For instance, if the student is a nervous we should avoid blaming him/her or interacting

**Figure 8.4** Diagnoses emotion state in PANDA.TUTOR: Example-1
in an aggressive way. Instead, we should try to relax or calm them. The system also generates a different dialog for each personality type considering the appraisal of the student. Thus, we analyze the impact of positive and negative emotions on the student performance, check whether the reason behind this emotion is internally or externally, and try to increase/decrease the internal emotion by increase/decrease the external one. Note that, we do not cope with the situation on the event in the correct way. Everyone cope with the incoming event according to his/her appraising way and depends on the personality type of the person. Accordingly we will enhance the good coping way and adapt or regulate the bad way to teach the student how to be emotionally intelligence. In other words, in our system we analyze the impact of positive and negative emotions on student
performance, check whether the reason behind this emotion is internal or external, and try to increase/decrease the internal emotion by increasing/decreasing the external one. As in traditional learning environments in which, the teacher attempts to select the best conditions for learning and specially the emotional conditions. For example, the teacher could introduce a joke to the classroom in order to change the current emotional states of the students, if they are bored or frustrate. For the reason that, the student how is bored, anxious, anger or depressed, could not learn or think efficiently. For example, assume we have two personality types, the first one is a complicated student (high E, high N, high C), and the second student is an Brooder (low E, high N, high C). The extroversion factor in the first type could help him/her to be less depressed than the introversion factor in the brooder type. In the second type, the student will cope by blaming his/her self, and lose his/her confidence more easily than the complicated type if the student fails in the exam. While the complicated type could be more anxiety than the second type at the first impression, but s/he has a confidence in his/her self. Therefore, the system will try to reduce the nervous level for both types, and increase the confidence level of the brooder type. In addition, the system will also help the complicated person to concentrate more. These adaptation can be done by applying different regulation strategies. For Instance, the dialog will try to change the student appraisal of the situation (as considering the white side or seeing the glass as half full and think you will come through this difficult time). Moreover, the dialog will be empathetic with the very sensitive person, blaming the careless student and teach him/her how to appreciate his/her responsibility. The system might behave rigorously with the student who did not make enough effort in exam preparation, and behave in an easy-going way with the student who did all his/her best but got a low exam score.
8.5 Affective Planning for Lesson Generation

The main idea is to improve the instructional planning for constructing Content plan and presentation plan considering the hybrid planning paradigm. The integration of cognitive emotion and motivation with intelligent Tutoring System aims to individually assist the students to achieve their goal. E-learning environments need to consider all of these factors and practically the emotional and motivational state of the student. It is worth taking into consideration, when we plan to achieve the same goal, that different plans could be created even though their external environment is the same. The different plans arise as a result of differences in the internal states and personality types of those students, or we can say according to motivational and emotional reactions of each student. Thus, our approach builds on the idea of personalized the instruction theory and learning style for student. Because each student has individual features as personality type, educational progress, learning style, goal..etc, there is a need to tailor the course structure and content to the student’s needs. In other words, we aim to develop an adaptive learning environment in which the course Content and pedagogical aspects are adapted to the student. The design of an adaptive learning system requires a huge number of rules which in most of the system are represented as if - then rules. In addition the relations between the system, the student, and course content are complex relations. The proposed methodology is based on an intelligent mechanism that tries to mimic an instruction designer model. Such that, all the adaptation rules and the relations between the system modules are modeled in the planning domain. However, the learning materials are defined by teacher-expert. In this context we have chosen to illustrate an approach by applying a personalized tutoring system by considering relevant student states and the actions which cause a new course to be generated. The action is applicable if its preconditions hold; their effects define properties of the successor state. The planner creates its teaching plan by simulating the effects of the operator actions on a model of
Chapter 8 Affective Planning

the student. Beginning from the current student module, the planner attempts to find a sequence of operator actions which will achieve the instructional goal. An operator is much like a rule in a production system. It has a set of preconditions which specify the conditions under which the operator action may be invoked, and a set of expected effects, which indicate the changing in the student module which are expected to occur when the operator action is invoked. Using the preconditions and expected effects of the operators, the planner can simulate the invocation of the various steps of a teaching plan, and thereby determine whether the plan is likely to be successful.

A successful plan is one which can be applied to the current student module to achieve the instructional goal. Since planning begins from the current student module, the planner automatically tailors teaching plans for the particular student being taught.

In addition, our approach follows the deep learning rather than surface learning approach [115]. The deep learning approach encourages students to understand the concept, relate new content to the previous knowledge, which helps students to structure and organize their knowledge. While the surface learning approach is interested only in a part of the material and memorized facts, ignoring the attempt to construct the student’s knowledge. So, the PANDA.TUTOR system is a constructive system which helps the student not only to complete their course more accurately and efficiently, but also to build new knowledge in an enjoyable way. The plan is based on the course structure, especially the concept goal structure and its relations with other concepts and the student module. The system generates lessons tailored to the student personality type, cognitive state, and emotional state of each student with the aim to individualize the instructions.

The educational material provided for each outcome concept, is organized in different levels of performance which should be achieved by the student in order to master the concept. The instruction strategies are adapted for the presentation of the educational material of the outcomes concepts follow student learning style. This can be used to design, develop, and deliver educational material or resources to maximally motivate and simulate the student’s acquisition of the subject matter in
an attempt to individualize the instruction. Moreover, the generated course is adapted differently according to the emotional and motivational state of the student.

The course generator automatically assembles learning objects retrieved from one or several repositories to generate the lesson for student according to student model characteristics, to help students to reach their learning goal. Hence, the pedagogical knowledge guides the instructional planning to generate a personalized plan for the student. Our goal is to reuse existing learning objects in the learning process. Each learning object in the repository should be associated with its pedagogical role (an introduction, an example, a test, ect.). By using our ontologies (course ontology, student ontology) for specifying the course concepts and associated teaching materials together with learning object meta-data, and for recalling student information needed to tailor the presentation for an individual student which is the role of the pedagogical module.

The basic goals for personalized e-learning is to improve the instructional planning for selecting the appropriate concept, and to teaching the material to the student. In order to attain information provided by the student model, it is necessary to determine which instructional strategies are suitable for the student’s state and how these strategies could be supported by the knowledge representation schema of the system.

Adaptation of the course content implies that different students receive different versions of the same concept lesson. Common approaches are:

- Adapting the scenario type.
- Adapting the level of depth (quantity level) according to student level and goal (ranging from high level overview to in-depth explanation).
- Adapting the level of difficulty (quality level) appropriate to level of understanding.
- Considering different learning styles.
- Considering different motivational states.
• Considering different emotional states.

According to the instructional planning process, the content planning in our system determines the concept goal to be achieved, while the delivery planning selects the learning objects that were considered appropriate for achieving the desired scenario. The course module ontology plays an important factor in defining and assembling the learning objects to achieve the desired learning outcomes. Therefore, using the ontology for finding the learning objects for the generated lesson is very useful.

In PANDA.TUTOR, we integrate the motivational and emotional goals into planning. It is a fact that students with different motivations and emotions interact differently with the learning process. So, student motivations and emotions explicitly generate goals in response to changes of the student’s motivational values, such goals are posed to a planner according to the appraisal process. The motivation and emotional goal is modeled within the planning domain model. This model is responsible to generate plans which satisfy the constraints imposed by the motivations and their associated goals. In addition, different methods of motivational and emotional goals are considered. We will describe course generation based on pedagogical tasks and methods, formalized in hierarchal task network planning. But first we will explain the regulation process in our system and describe the motivational and emotional regulation strategies that are used for adapting the student behavior.

8.5.1 Regulation Process in PANDA.TUTOR

It is important to consider the emotional state during the planning process, such that; the bad emotion can inhibit or prevent the learning process. So, it is important to plan what a student can do if he/she has such emotion, or how a student can achieve his/her learning goal in that emotion. So, the system tries to adapt the emotional state of the student or change the student’s goals according to his/her emotional state. For example if the student emotion is bad for learning or studying a
new lesson (note that some bad emotions are good for learning and some good emotions are bad for learning), then it is better to change the learning goal to a training goal, revision goal, or even decide to take a break, because if the student studies with such an emotion there are no results that can be achieved. Therefore, it is important to consider different emotional and motivational states during the planning design, which helps students to achieve their goal. Depending on the psychological and pedagogical theory and experience, the ways by which a person perceives, acts, and reacts is influenced by his or her personality type. We aim to cope or regulate with the emotional state of the student. The student’s personality and the emotional state help the pedagogical module to decide the best regulation and instruction strategy. According to the student’s appraisal, the system will expect the coping way depending on the personality type of the student. For instance, anger and shame, have a negative effect on learning. But, if students believe in their ability; the student’s failure can motivate student to increase their effort. Thus, The system will follow the good coping way and try to regulate the bad one.

The planning system will activate corresponding to the student’s learning, motivational, and emotional goals to induce the optimal learning environment. Thus, the system provides an environment in which the learning takes place through teaching, coaching, testing, and companion scenarios. Various affective parameters are used to determine which tutoring strategy to use and which instructional act to perform (sympathizing or non-sympathizing feedback, motivation, explanation, steering, etc.).

Thus, according to the appraisal of the event and the effects of the event on the learning goal we generate the appropriate plan to cope or regulate with the current student emotion and motivation state considering student personality. Also, by analysis of the reason for the current emotion the system decides the type of regulation strategy. Thus, if the reason can be manipulated (controlled), problem regulations strategies are considered, otherwise, emotion regulation strategies is selected. Table 1 includes our (coping/Regulation) strategies.
The Regulation Process has two main sub-processes:

- Selecting a convenience learning scenario, and
- Generating the lesson of concept goal for the selected scenario.

Two types of regulation strategies are considered, the problem focused regulation strategy and emotion focused regulation strategy as in table (tab:logicalLanguage):

### 8.5.2 Affective Planning Problem

Planning is the process of generating a plan based on three essential inputs; an initial state of the world, a set of possible actions that can be executed, and a set of goals to be reached.

For the hybrid planning environment the main goal task will be the learning goal. We also consider two types of goals, the learning goal and motivation and emotion goals according to the student state. For example we increase confidence or effort, and decrease sadness, anger, or disappointment. The first step in the course generation process is to determine the concepts that should be taught in the course. In other words we should first generate the content plan for the current lesson, which depends on the concept goal that is needed to be learned.

In the beginning, the student can select a course and the system presents the learning content according to the student’s level. Then, the student can select a concept from the concept list of the course and the system generates a lesson to the student.

### 8.5.3 Affective Planning Domain

As we deal with a deterministic domain all actions have a well-defined outcome. This allows us to predict feature statues of the tutoring system such as if we know the state of student and the plan action, we can define the student state after action application. The planning system traditionally
### Table 8.1 Some regulation strategies.

<table>
<thead>
<tr>
<th>Problem focused regulation</th>
<th>Situation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Change the scenario type</td>
</tr>
<tr>
<td></td>
<td>Change the game strategy</td>
</tr>
<tr>
<td></td>
<td>Change learning style</td>
</tr>
<tr>
<td></td>
<td>Change type of example or exercise</td>
</tr>
<tr>
<td></td>
<td>Change the level of example or exercise</td>
</tr>
<tr>
<td></td>
<td>Facilitate success.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Emotion focused regulation</th>
<th>Encourage student.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Praise/punish student high/low effort</td>
</tr>
<tr>
<td></td>
<td>Praise ability</td>
</tr>
<tr>
<td></td>
<td>Suggest challenges</td>
</tr>
<tr>
<td></td>
<td>Remind student for his/her the previous success</td>
</tr>
<tr>
<td></td>
<td>Praiseworthy as well as blameworthy</td>
</tr>
<tr>
<td></td>
<td>Shift blame from the actual reason to another</td>
</tr>
<tr>
<td></td>
<td>Positive interpretation to the actual reason</td>
</tr>
<tr>
<td></td>
<td>Ignoring the emotion reason</td>
</tr>
<tr>
<td></td>
<td>Promotion for increasing difficulty level</td>
</tr>
<tr>
<td></td>
<td>Change student belief from low ability to low effort</td>
</tr>
<tr>
<td></td>
<td>Suggesting seeking for social support, take vacation, play game, etc</td>
</tr>
</tbody>
</table>
assumes that all goals are known in advance and are externally imposed and that the planning process is finished once a plan has been generated in which all goals are achieved. Once a plan has been generated, its execution is no longer the concern of the planner.

The plan is a sequence of students’ actions to achieve desires or a concrete goal. In this context we have chosen to illustrate an approach by applying a personalized tutoring system by considering relevant student states and actions which cause a new course to be generated. An action is applicable if its preconditions, which their effects define properties of the successor state.

The regulation strategies will be considered during the domain as a primitive tasks or complex tasks. Primitive tasks can assemble to achieve a complex goal by different way.

The goal, for example, could be a complex task as **Facilitate Success**, which can be achieved by different ways as deliver easy exercise or deliver easy example.

In the following subsections, we will present the domain model that we have developed for our course generation system. As we described in chapter 3, the foundations of our domain model \( D = \langle Q, M, T \rangle \) is defined according to the given logical language \( L \). Note that the domain model \( D \) is consistent if and only if the set of tasks \( T \) and decomposition methods \( M \) are consistent (see definitions 9 and 10).

**Definition 9 (Consistent Task).** The task \( \langle \text{type}, \text{Prec}(t(\bar{\tau})), \text{eff}(t(\bar{\tau})) \rangle \) is called consistent if and only if the following conditions are satisfied:

1. All predicates \( R(\bar{\tau}) \) in \( \text{eff}(t(\bar{\tau})) \) are built over flexible relations.

2. A \( \text{Prec}(t(\bar{\tau})) \) must be applicable. This means, if there is no task achieves this precondition. then there will be no state in which the specified task becomes applicable.

3. The effects \( \text{eff}(t(\bar{\tau})) \) are conflict-free. This means, \( \forall e, \bar{e} \in \text{eff}(t(\bar{\tau})) \) the following holds:

   if \( e = +R(\tau_1, \tau_2, \cdots, \tau_n) \) Then \( \bar{e} = -R((\bar{\tau}_1), (\bar{\tau}_2), \cdots, (\bar{\tau}_n), (\tau_1) \neq (\bar{\tau}_1) \lor \cdots \lor \tau_n \neq (\bar{\tau}_n)). \)
4. For every task \( t(\bar{t}) \) has abstract type, there exist at least one decomposition method \( m \in \mathbb{M} \) such that \( m = \langle t(\bar{t}), p \rangle \).

**Definition 10 (Consistent Decomposition Method).** The decomposition method is called consistent if and only if the following conditions hold:

1. For every method \( m \in \mathbb{M} \) with \( m = \langle t(\bar{t}), p \rangle \), \( t(\bar{t}) \) is an abstract task schema in \( \mathbb{T} \) and the partial plan \( p = \langle \text{TE}, \prec, \text{VC}, \text{CL} \rangle \) is consistent. A partial plan is consistent if and only if the transitive closure of the ordering constraints in \( \prec \) are irreflexive, the variable constraints \( \text{VC} \) are consistent and the ordering constraints do not contradict the causal links \( \text{CL} \).

2. The recursive decomposition method must allows for termination.

Now we will illustrate how to build concrete domain for the presented PANDA.TUTOR system. Our motivation for this is to construct general domain model. That means our domain model can be used with any course (i.e. it is applicable to construct any kind of course over it).

### 8.5.4 Sorts and Relations

For a proper formal description of our domain, we can not refer to the IPC domain repository because non of the current PDDL versions supports modeling for course generation. Therefore, we will present the **PANDA-Course Generation** domain (PCG) in the formalization that we have introduced in chapter 3, that means we are defining a decomposition domain model \( D = \langle \mathbb{Q}, \mathbb{M}, \mathbb{T} \rangle \) according to a logical language \( \mathbb{L} \).

The language \( \mathbb{L}_{\text{PCG}} \) has to be defined by an appropriate tuple

\[
\mathbb{L}_{\text{PCG}} = \langle Z_{\text{PCG}}, \prec_{\text{PCG}}, R_{\text{PCG}}, \text{Const}_{\text{PCG}}, V_{\text{PCG}}, O_{\text{PCG}}, T_{\text{PCG}}, L_{\text{PCG}} \rangle.
\]

Our domain consists of 221 sorts, 82 relations, 105 task schemata, and 62 decomposition methods declarations. Note that we do not present all details about the domain components, but we
Table 8.2 A subset of sort elements of the logical language $\mathbb{L}$, which are used to model the functionality of learning student.

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality_level</td>
<td>···</td>
<td>Constants of this sort are used to represent the possible quality level of student (abstract).</td>
</tr>
<tr>
<td>Learning_Style</td>
<td>···</td>
<td>Abstract</td>
</tr>
<tr>
<td>Quantity_Level</td>
<td>···</td>
<td>Abstract</td>
</tr>
<tr>
<td>Personality_Type</td>
<td>···</td>
<td>Abstract</td>
</tr>
<tr>
<td>Performance_State</td>
<td>···</td>
<td>Abstract</td>
</tr>
<tr>
<td>Confidence_Level</td>
<td>···</td>
<td>Abstract</td>
</tr>
<tr>
<td>Personality_type</td>
<td>···</td>
<td>Abstract</td>
</tr>
</tbody>
</table>

will incrementally introduce some representative pieces from all of the above components in the following sections. The first thing to do when creating the PCG domain is to introduce a sort hierarchy $Z_{PCG}$ for describing the occurring objects.

$$Z_{PCG} = \{\text{Student, Course, Scenario type}\}$$

$$\prec_{PCG} = \{(\text{Personality type, Student}), (\text{Emotion style, Student}), (\text{Motivation state, Student})\}$$

The PCG sort hierarchy is however too large to be displayed here, Figure 8.6 depicts a simplified model fragment. Regarding the geometric shapes, the abstract sorts are represented in ellipses while sort names in boxes represent sorts for which constants can be provided. For example, the super-sort student represents the properties of students such as personality type, emotion, motivation ... etc (cf. chapter 6). While, the super-sort course represents the course properties such as concepts and knowledge object ... etc (cf. chapter 5).

Concerning the relation symbols, we will focus on a representative fragment of the complete
8.5 Affective Planning for Lesson Generation

**Relations** $R$

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studet_Personality</td>
<td>$Student \times Personality_type$</td>
<td></td>
</tr>
<tr>
<td>Studet_level_homwork</td>
<td>$Student \times concept \times homework \times Quality_level$</td>
<td></td>
</tr>
<tr>
<td>Student_quantity_level</td>
<td>$Student \times course \times quantity_level$</td>
<td></td>
</tr>
<tr>
<td>Call_exercise_concept</td>
<td>$Concept \times Exercise \times Quality_level$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

**Constants** $Const$

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low_level</td>
<td>Quality_level</td>
<td></td>
</tr>
<tr>
<td>Good_level</td>
<td>Quality_level</td>
<td></td>
</tr>
<tr>
<td>Haidi</td>
<td>Student_name</td>
<td></td>
</tr>
<tr>
<td>Peter</td>
<td>Student_name</td>
<td></td>
</tr>
<tr>
<td>Object_Oriented</td>
<td>Concept_name</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>
set of definitions. Let us begin with the state-independent relations. They basically cover two static aspects of the learning processes: student specifications and course specifications.

\[ R_{rigid-PCG} = \{ \text{Student\_Personality}_{\text{student\_ID, personality\_type}}, \text{Selected\_Concept}_{\text{student, concept}}, \text{Concept\_Prerequisite}_{\text{concept, prerequest}} \} \]

The \textit{Student\_Personality} relation is used for expressing the personality type of the student. The \textit{Selected\_Concept} and \textit{Concept\_Prerequisite} relations reflect that the concept goal and its prerequisite.

Concerning the dynamic state attributes, the following flexible relation symbols are an assortment of properties of student.

\[ R_{flexible-PCG} = \{ \text{Student\_Performance}_{\text{student\_ID, concept, Not\_Learned}}, \text{Student\_Learning\_style}_{\text{student\_ID, Learning\_style}}, \text{Selected\_Scenario}_{\text{scenario\_type}} \} \]

The \textit{Student\_Performance} relation clarifies the performance state of the concept goal. While, the learning style and the learning scenarios are specified in the relations \textit{Student\_Learning\_style} and \textit{Selected\_Scenario} respectively.

### 8.5.5 Tasks and Methods

A task \( t \) is a tuple \(< \text{pre}; \text{post} >\), specifying a precondition and a post condition. Pre-conditions and post-conditions are sets of literals over the relations of the logical language \( \mathbb{L} \) and depend on the task parameters \( T(t) = T_1(t), \cdots, T_n(t) \), where \( n(t) \) is the length of this sequence, also called the variety of task \( t \). For convenience, we also write \( t(T) \) to refer to a task \( t \).

In our approach, we present the hybrid planning technique to view the complex personalized tutoring system by describing various functions for an individual user of the teaching system. This includes generating a plan for a specific user in order to achieve a given student goal.

At the beginning the type of learning scenario can be selected either by the student or by the system according to the emotional and motivational state of the student and the performance history of the
Let us begin with the most important abstract task that is the entry point to the expansion hierarchy with respect to HTN problem definition: the abstract task $PANDA - Tutor - Course$. For variables $s$ of sort student, $p$ of sort personality, $c$ of sort concept, $e$ of sort emotion, $m$ of sort motivation and $sc$ of sort scenario. The PANDA-Tutor-Course is defined as it is common for learning-like schemata as follows:

$$PANDA-Tutor-Course (s: Student, p: Personality, c: Concept, e: Emotion, m: Motivation, sc: Scenario)$$

Pre:
- Student-Personality($s$: Student, $p$: Personality)
- Student-Emotion($s$: Student, $e$: Emotion)
- Student-Motivation($s$: Student, $m$: Motivation)
- Scenario-type($sc$: Scenario)

Eff:
- Student-Performance($s$: Student, $c$: Concept, Learned)

According to scenario types we provide exactly four refinements, in which the PANDA-TUTOR-Course activity is decomposed into a task network with subtasks. In the first cycle, the learning scenarios are considered to differentiate between different decomposition methods that solve the abstract task Panda-Tutor-Course. We actually have four decomposition methods are used to refine the abstract task Panda-Tutor-Course: Method-Lesson-Scenario, Method-Game-Scenario, Method-revision-scenario, and Method-Test-Scenario (see Figure 8.7). For instance, if the student prefers to study the concept according to lesson scenario, the abstract task for the lesson scenario is $Generate_Lesson_Scenario$. For variables $s$ of sort Student, $p$ of sort Personality, $c$ of sort Concept and $sc$ of sort scenario. The schemata of task $Generate_Lesson_Scenario$ is as follows:
The intended semantics of this schema is of course that the student is to be taught the specified concept. The intended semantics of this schema is that the student is to be taught the specified concept. On this level of abstraction neither the learning style nor the emotional and motivational state is relevant, the only significant state change concerns the student personality type. In the next cycle, the abstract task Generate-Lesson-Scenario can be decomposed by two different decomposition methods according to student’s performance history of the selected concept. For instance, if a student studies the concept for the first time, the sub-tasks Check-PreConcept and Learned-Concept are defined:

M-Learn-Concept-FT (?s: Student, ?p:Personality, ?c:Concept,  
?pc:Concept, ?ph:Student-Performance-History) 
Sub-Tasks:  
T1: Check-PreConcept(?cs:Student, ?cp:Personality,  ?cc:Concept,  
?cpc:Concept)  
T2: Learn-Concept(?ls:Student, ?lp:Personality, ?lc: Concept)  
Variable Constraints:  
Ordering Constraints:

\[ T_1 < T_2 \]

We provide two refinements, in which the concept is studied for the first time or if it studied before. The first refinement is decomposed into a task network with six sub-tasks, in which the emotion and motivational states are considered and then the generated course is manipulated. The second refinement is decomposed into a task network with three sub-tasks according to the history of the student’s performance for the respective concept (See Algorithm 8.10).
Algorithm 4: Course Generation Algorithm

1 Generation (Goal)

2 foreach learning goal (G) do

3 Find the prerequisite relation

4 Check the student state in Pre-concept

5 if student studied this concept before then

6 Check student Homework Level in Pre-requisite concept

7 if it high Level then

8 Deliver summary

9 if it intermediate then

10 Deliver summary

11 Deliver exercise

12 if it low then

13 Deliver revision

14 else

15 Deliver the prerequisite concept Lesson

16 Find the goal Concept

17 Check student state in this concept

18 if student studied this before then

19 Check student level

20 else

21 Deliver the Lesson for the selected concept

22 Find the relevant instructional object sub-Concept, Parts, process, principle
Figure 8.8 shows the editor tool of PANDA domain with the `Generate_Lesson_Scenario` task’s schema definition on the right and the decomposition hierarchy on the left. The latter shows for each of the sub-tasks, `Consider_emotional_State`, `Consider_Motivational_State`, `Teach_first_Level`, `Teach_Second_Level`, `Teach_Third_Level`, `Teach_Fourth_Level`, their respective methods. While the implementation of the handover procedures, for which `Consider_emotional_State` and `Consider_Motivational_State` stand, are organized according to the emotion and motivation strategy that manipulate different emotional and motivational state, the actual learn procedure, encoded by `Teach_first_Level`, `Teach_Second_Level`, `Teach_Third_Level` and `Teach_Fourth_Level`.

In our learning context, the learning process is organized in four levels as follows (see figure 8.9):

1. In the first level, a presentation (Tell) (i.e. definition, introduction...) for the whole view of concept is introduced, as well as presentation (Tell) and demonstration (Show) of the concept’s part(s) of the concept. Note that, we consider different learning styles for each KO. The task Select Learning style connects between the personality type and the appropriate learning style.

2. In the second level, we deliver activation (remember) and application (Do) for both concept and the concept’s part(s) and then teach the student presentation and demonstration about the Process (How to) and the principle (what happen) of concept with different learning styles.

3. In the third level, the activation application is delivered for concept part, process and principle of the selected concept. Also, presentation and demonstration about the principle subclasses with different learning styles are delivered.

4. In the fourth level, the student is asked to apply all KO of the concept. Finally, deliver an assignment about the whole concept to assess the student’s level.
For teaching each knowledge object, four different decomposition methods are defined to represent the learning styles. For instance, to model the Pragmatist style for the concepts process (How to), the generated plan will call demonstration about the process, then presentation about the concepts process, and finally application about the process. In the Reflector style, the presentation will be generated then demonstration and application. The connection between the personality type and the appropriate learning style will be considered during the domain of the planning. Note that during the refinement process, we connect between the personality type and the appropriate learning style. Thus, for each KO we considered different learning styles. For instance, the appropriate learning style for the personality type Entrepreneur (high E, low N, high C) is the Pragmatist style.

Moreover, for each instruction component strategy, different motivational goals are considered. For instance, the abstract task Do-Application-Concept is refined according to student motivational state (i.e. low confidence, low effort, high confidence) by three methods: M-Simulate-Challenges, M-Increase-Effort, and M-Facilitate-Concept. For example, the method M-Facilitate-Concept has only one primitive task so-called Deliver-Easy-Application.

The generated plan consists of a set of tasks which include querying predicates. These predicates are used to retrieve the learning objects from the course ontology. The effect of this task is a set of calling predicates. These predicates are used to retrieve the course materials from course ontology.

Deliver-Easy-Application (?conf :LowConfidence, ?c: Concept)

Precondition:

Student-Motivation(?s :Student, ?conf : LowConfidence)

Calling-easy-application(Application, Easy, ?c :concept)

Effect:

Calling-easy-application(Application, Easy, ?c :concept)
Figure 8.6 A fragment of the PANDA.TUTOR sort hierarchy
Figure 8.7 The decomposition hierarchy for the complex task schema PANDA – TUTOR – Course.
Figure 8.8 The decomposition hierarchy for the complex task schema Generate_Lesson_Scenario and its definition.
Figure 8.9 The decomposition hierarchy for the learning process.
Figure 8.10 The decomposition hierarchy for the learning process.
Chapter 9

Conclusion

9.1 Research Contributions

In this thesis we introduced the PANDA.TUTOR system, a personalized intelligent tutoring system. The goal of this system is to improve the emotional intelligence of students as well as the intellectual intelligence.

PANDA.TUTOR is an authoring/course generation system based on hybrid planning and semantic web techniques. The authoring system helps the non-programmer teacher to configure different aspects of the domain model with the keeping all the authoring process low in terms of time and effort. Thus the authors only need to define the course structure and content being taught, and enrich it with different teaching materials, which helps the pedagogical module to tailor the lesson to a student module. For that purpose, we have developed a general ontology that can be used for different types of lessons and for different adaptation expectation for personalized e-learning. The course ontology provides an adaptive learning environment to facilitate sharing and reuse of learning materials. In addition, we have considered realizing separately reusability of the course content and the reusability of learning objects. Also, the ontology based semantic web permits the
retrieval of learning materials after the pedagogical module generates the course plan. Besides, PANDA.TUTOR keeps a model of the student that represents not only the cognitive state, but also the personality type, current emotional and motivational state. Accordingly, we modeled the students’ data as an ontology. The pedagogical module in PANDA.TUTOR based on the hybrid planning uses different aspects from the student module. As a consequence, students will be able to achieve learning goals more efficiently when pedagogical procedures are adapted to their individual differences. However, the process of arranging personalized adaptations is usually complex because people with different personality cope differently with a problem. Accordingly, we generated a dialog based planning system to improve the emotional intelligence of the student and grasp the information necessary for the appraisal process. A new dialog based planning approach is introduced to model this dialog. The system can generate different styles of dialog considering the personality type of the student, the result of the dialog model is stored according to our emotional ontology. In this dialog, we build an individual’s interpretation of how external events relate to their goals and desires (the person’s environment relationship), appraisal characterizes this interpretation of a number of abstract features that are useful for guiding the student’s behavior. Based on the the appraisal process, the planning system uses the appraisal parameters, and the coping way of the student (depending on his/her personality type) to select the the appropriate learning scenario. Then consequently, the hybrid planner generates an individual lesson structure and an individualized lesson content for each student by selecting the most optimal learning concepts and the most relevant learning web resources. In other words, the system follows the positive coping way, while it regulates the negative way for the aim of teaching student how to be emotionally intelligence and adjust the weak points in student’s personality. Note that, besides the learning goal of the student, we have considered the emotional and motivational goals. The system contains with different learning scenarios, teaching strategies and learning styles. In PANDA.TUTOR the students personality type is taken as a guide to select the appropriate dialog, as well as determin-
ing the appropriate regulation strategy, according to the emotional and motivational states of the student.

9.2 Future Work

In our system we deal with each student individually. We handled different types of emotion for each personality type and generate an appropriate plan. Another application could be mobile phones. If some one is an emotional distress the phone could suggest calling a friend (for the introverted type), or meeting a group of friends (for extroverted type). The phone could also start entertain a bored user or remember the user of a good upcoming event the system found in the calender. Moreover, the phone could send a text message to friend, asking them to aid, congratulate or encourage the user. It is the idea to have a personal machine or wearable computer not just for the foundation strong and retrieval, but because it deals with you on a personal way and delivers plan according to your emotional state.
Bibliography


[69] A. Tate, “INTERPLAN: a plan generation system which can deal with interactions between goals,” Memorandum MIP-R-109, Machine Intelligence Research Unit, University of Edinburgh (1974).


