KING'S College LONDON

University of London

An Intelligent Tutoring System for First-Year Level Java

6CCS3PRJ Final Project: Final Report Submission

BSc Computer Science with a Year in Industry

Author: Miruna-Ioana Pîrvulescu
Supervised by: Dr. Sahar Al-Sudani
Student Number: 1701543

March 11, 2021

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary.

Signature: *Miruna-Ioana Pîrvulescu*

Date: March 11, 2021

Abstract

The contemporary period is relying more and more on the use of technology, and the education sector is no exception. An increasing number of schools is opting for smart pedagogical solutions in their classrooms through the use of specialised hardware (such as the smart board) and software applications (for example, learning management systems).

This project aims to implement a simple Intelligent Tutoring System meant to teach first-year university students in Computer Science how to code in Java. The implemented solution is a piece of software that queries and infers details from an ontology that encapsulates the topic. The application is able to provide information about Java statements, types, operations, and object orientation.

Acknowledgements

I would like to thank Dr. Al-Sudani for all the guidance, feedback, and support offered throughout the development of this project. I found our meetings to be very helpful in choosing the best approaches that led to the creation of this application.

Many thanks to the Department of Informatics for the quality of the modules taught throughout this degree. All the knowledge I accrued over the past four years has enabled me greatly to produce this project.

I would also like to thank everybody who provided feedback during the evaluation stage of the project from the point of view of the end user.

Contents

Chapter 1

Introduction

1.1 Motivation

The biggest exposure that a Computer Science student has to cutting-edge technologies comes from their personal rather than academic environments, through video games, streaming applications, or online shopping. The academic curriculum is mostly theory-based and human-taught, with some components of programming assignments and classroom examples that are only lightly explained.

This traditional method is a very effective and informative approach to teaching. However, it is lacking an engagement factor of practical demonstration. This factor could potentially drive students to seek further information on their course topics and to achieve better results from their learning, meaning that they are exploiting their academic resources to the fullest extent.

This project is meant as a step forward for smarter education. By developing a well-rounded knowledge formalism and creating a small application to display it with the use of Artificial Intelligence, student engagement has the potential to rise through exposure to a modern teaching style. Rather than having a human explain a type of technology, that technology can "gain a voice" and teach the student about itself.

1.2 Key Terms

An **Intelligent Tutoring System** (ITS) is a type of software application aimed at teaching students about various topics, requiring little to no intervention from a human expert[1].

A **Domain Model** refers to the knowledge that the ITS manipulates in order to produce human-readable data through referencing given knowledge and inferring new facts.

The most important term of this project is the **ontology**. In the field of Computer Science, an ontology is a method of representing entities in rapport to one another through means of relations. For example, the entity Conditional Statement *is* a sub-category of the entity Statement and it *has* a Condition that enables it to run.

1.3 Aims and Objectives

The first objective of the project is to create a basic Intelligent Tutoring System. This system needs to teach first-year Computer Science undergraduate students about programming in Java. The main area of study is on different statement types, such as Conditional and Loop Statements.

The second objective of the project is to build a Domain Model as a computer-readable formalism, such as an ontology. The system must be able to make inferences based on its knowledge. Moreover, it must present all the domain-specific information, both given and inferred, to the student.

The application must be intelligible, informative, engaging, and easy to use.

For the purpose of this project, a research component was essential in order to understand how such a system can be developed.

Chapter 2

Background

This chapter covers an extended overview of the key terms described in section 1.2, including explanations for the choices that were made during the research and development phases. I will also discuss the technologies chosen for building the Intelligent Tutoring System and draw some comparisons with the other possible alternatives.

2.1 Intelligent Tutoring Systems

As previously defined, an Intelligent Tutoring System is a type of software that can teach a subject to students without need for additional support. Such a system is constructed based on four pillars, called *models*.
First, we can identify a *Student Model*, which is meant to track the progress made by the learner throughout their studies and offer corrections where necessary.
Secondly, the Intelligent Tutoring System has a *Pedagogical Model*. Its purpose is to identify an optimal teaching strategy based on the learning style of the student.
The third pillar of an ITS is the *User Interface Model*. This enables the student to access the knowledge behind the ITS through a Graphical User Interface, such as

the front end of a website.
Finally, an Intelligent Tutoring System has a Domain Model. This is also known as the knowledge model, and it is represented by a knowledge representation formalism. As this is the main component of the project, it will be described in detail in the following section.

2.2 Knowledge Representation Formalisms

Knowledge representation formalisms are means of expressing the human-readable information in a computer-readable fashion. Some examples of such formalisms are ontologies, semantic networks, frames, and rule-based systems.
The process of deciding which of these types of formalisms better suits this project involved significant research, comparisons between findings, and personal preference.

2.2.1 Ontologies

Ontologies encapsulate the formal design and definition of a class hierarchy. They are widely used within the field of education[2], as well as in knowledge representation and design. Due to the rigorous hierarchical nature of ontologies, their use became best candidate for the development of this project.

2.2.2 Semantic Networks

Semantic networks are visually represented by graphs, where the entities are the vertices and the relations are the edges[3]. They share a lot of similarities with ontologies, however semantic networks are less rigorous formalisms.[4]
Because of this, only the graphical representation of semantic networks has been used for the initial structuring of the knowledge model. The first data diagrams were created in the form of mind maps, which perfectly resemble the visual component

brought by semantic networks. (An example of such a mind map will be provided in the Development section.)

2.2.3 Frames

Frames are very widely used for knowledge modelling in Artificial Intelligence. They make use of smaller ontologies that are interconnected to generate the bigger picture. This type of formalism is a better fit for larger scopes than the one of this project's domain, and therefore its usage would have brought a risk of increasing the complexity of the project above necessity. However, for the description of a larger variant of this domain, where the model would have to also cover various APIs of Java or more in-depth object oriented practices, frames would have been a perfect fit.

2.2.4 Rule-Based Systems

As the name suggests, a rule-based system is a set of rules (if-else statements). These rules are used to teach the expert system about the domain and how to infer new information from it. This is the classic method of creating expert systems.
While rule-based systems are very strong competitors for ontologies, a general comparison study between the two suggests that rule-based systems are less descriptive than their adversary[6].

2.2.5 Final Notes on the Comparison

The conclusion drawn at this point in research was that ontologies are the best fit for creating a knowledge model for Java.
While all the assessed knowledge representation formalisms are widely accepted, they are also fit for different types of requirements. Semantic networks are not formal enough in this situation, frames are better for higher-scaled projects, and rule-based

systems are the only true alternative that is appropriate for the scope of this domain. There was also an element of subjectivity involved in this selection. Since it is the domain expert whose knowledge needs to be passed on to the system, I needed to choose the type of formalism closest to my own learning style and understanding. Knowledge is best passed down when thoroughly understood by the teacher, otherwise the quality of the passed information is sub-optimal.

2.3 From Lightweight to Heavyweight Ontologies

2.4 Java Web Applications

2.5 Hosting the Application Online

Chapter 3

Specification and Design

Chapter 4

Evaluation

Chapter 5

Conclusions and Future Work

Bibliography

[1] Intelligent Tutoring Systems, Wikipedia.org

[2] Wikipedia.com, Ontology (information science), Overview

[3] obitko.com, Ontologies and the Semantic Web, Semantic Networks

[4] Salem, Alfonse, 2008, Ontology versus Semantic Networks for Medical Knowledge Representation, p 5

[5] Golden, 2013, A Unified Formalism for Complex Systems Architecture, p 14

[6] Czarnecki & Sitek, 2013, Ontologies vs. Rules — Comparison of Methods of Knowledge Representation Based on the Example of IT Services Management, p 108

Appendix

Chapter 6

User Guide

Chapter 7

Program Listings