Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Authoring Gamified Intelligent Tutoring Systems

Diego Dermeval Medeiros da Cunha Matos

Thesis submitted to the Coordenação do Curso de Pós-Graduação em Ciência da Computação of the Universidade Federal de Campina Grande - Campus I as part of the necessary requirements to obtain the degree of Doctor in Computer Science.

Concentration Area: Computer Science
Research Line: Computational and Cognitive Models

Ig Ibert Bittencourt Santana Pinto
(Advisor)

Campina Grande, Paraíba, Brasil
©Diego Dermeval Medeiros da Cunha Matos, 03/15/2017

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

M433a Matos, Diego Dermeval Medeiros de Cunha Matos.

Authoring gamified intelligent tutoring systems / Diego Dermeval Medeiros de Cunha Matos. ‾ Campina Grande, 2017.

225 f. : il. color.

Tese (Doutorado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2017.

"Orientação: Prof. Dr. Ig Ibert Bittencourt Santana Pinto".

Referências.

1. Autoria de Sistemas Tutores Inteligentes. 2. Sistemas Tutores Inteligentes. 3. Gamificação. I. Pinto, Ig Ibert Bittencourt Santana. II. Universidade Federal de Campina Grande, Campina Grande (PB). III. Título.

CDU 004.89(043)

"AUTHORING GAMIFIED INTELLIGENT TUTORING SYSTEMS"

DIEGO DERMEVAL MEDEIROS DA CUNHA MATOS

TESE APROVADA EM 15/03/2017

IG IBERT BITTENCOURT SANTANA PINTO, D.Sc, UFAL
Orientador(a)

DALTON DARIO SEREY GUERRERO, Dr., UFCG
Examinador(a)

ALAN PEDRO DA SILVA, Dr., UFAL
Examinador(a)

SEIJI ISOTANI, PhD., USP
Examinador(a)

JULITA IVANOVA VASSILEVA, PhD
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Sistemas Tutores Inteligentes (STIs) têm recibo a atenção de acadêmicos e profissionais desde da década de 70. Tem havido um grande número de estudos recentes em apoio da efetividade de STIs. Entretanto, é muito comum que estudantes fiquem desengajados ou entediados durante o processo de aprendizagem usando STIs. Para considerar explicitamente os aspectos motivacionais de estudantes, pesquisadores estão cada vez mais interessados em usar gamificação em conjunto com STIs. Contudo, apesar de prover tutoria individualizada para estudantes e algum tipo de suporte para professores, estes usuários não têm recebido alta prioridade no desenvolvimento destes tipos de sistemas. De forma a contribuir para o uso ativo e personalizado de STIs gamificados por professores, três problemas técnicos devem ser considerados. Primeiro, projetar STI é muito complexo (deve-se considerar diferentes teorias, componentes e partes interessadas) e incluir gamificação pode aumentar significativamente tal complexidade e variabilidade. Segundo, as funcionalidades de STIs gamificados podem ser usadas de acordo com vários elementos (ex.: nível educacional, domínio de conhecimento, teorias de gamificação e STI, etc). Desta forma, é imprescindível tirar proveito das teorias e práticas de ambos os tópicos para reduzir o espaço de design destes sistemas. Terceiro, para efetivamente auxiliar professores a usarem ativamente estes sistemas, faz-se necessário prover uma solução simples e usável para eles. Para lidar com estes problemas, o principal objetivo desta tese é projetar uma solução computacional de autoria para fornecer aos professores uma forma de personalizar as funcionalidades de STIs gamificados gerenciando a alta variabilidade destes sistemas e considerando as teorias/práticas de gamificação e STI. Visando alcançar este objetivo, nós identificamos o espaço de variabilidade e o representamos por meio do uso de uma abordagem de modelagem de features baseada em ontologias (OntoSPL). Desenvolvemos um modelo ontológico integrado (Ontologia de tutoria gamificada ou *Gamified tutoring ontology*) que conecta elementos de design de jogos apoiados por evidências no domínio de e-learning, além de teorias e frameworks de gamificação aos conceitos de STI. Finalmente, desenvolvemos uma solução de autoria (chamada AGITS) que leva em consideração tais ontologias para auxiliar professores na personalização de funcionalidades de STIs gamificados. As contribuições deste trabalho são avaliadas por meio da condução de quatro estudos empíricos: (1)

conduzimos um experimento controlado para comparar a OntoSPL com uma abordagem de modelagem de features bem conhecida na literatura. Os resultados sugerem que esta abordagem é mais flexível e requer menos tempo para mudar; (2) avaliamos o modelo ontológico integrado usando um método de avaliação de ontologias (FOCA) com especialistas tanto de contexto acadêmico quanto industrial. Os resultados sugerem que as ontologias estão atendendo adequadamente os papeis de representação do conhecimento; (3) avaliamos versões não-interativas da solução de autoria desenvolvida com 59 participantes. Os resultados indicam uma atitude favorável ao uso da solução de autoria projetada, nos quais os participantes concordaram que a solução é fácil de usar, usável, simples, esteticamente atraente, tem um suporte bem percebido e alta credibilidade; e (4) avaliamos, por fim, versões interativas (do zero e usando um modelo) da solução de autoria com 41 professores. Os resultados sugerem que professores podem usar e reusar, com um alto nível de aceitação, uma solução de autoria que inclui toda a complexidade de projetar STI gamificado.
Palavras–chave: Sistemas Tutores Inteligentes, Gamificação, Ferramentas de Autoria

Abstract

Intelligent Tutoring Systems (ITSs) have been drawing the attention of academics and practitioners since early 70's. There have been a number of recent studies in support of the effectiveness of ITSs. However, it is very common that students become disengaged or bored during the learning process by using ITSs. To explicitly consider students' motivational aspects, researchers are increasingly interested in using gamification along with ITS. However, despite providing individualized tutoring to students and some kind of support for teachers, teachers have been not considered as first-class citizens in the development of these kinds of systems. In order to contribute to the active and customized use of gamified ITS by teachers, three technical problems should be considered. First, designing ITS is very complex (i.e., take into account different theories, components, and stahekolders) and including gamification may significantly increase such complexity and variability. Second, gamified ITS features can be used depending on several elements (e.g., educational level, knowledge domain, gamification and ITS theories, etc). Thus, it is imperative to take advantage of theories and practices from both topics to reduce the design space of these systems. Third, in order to effectively aid teachers to actively use such systems, it is needed to provide a simple and usable solution for them. To deal with these problems, the main objective of this thesis is to design an authoring computational solution to provide for teachers a way to customize gamified ITS features managing the high variability of these systems and considering gamification and ITS theories/practices. To achieve this objective, we identify the variability space and represent it using an ontology-based feature modeling approach (OntoSPL). We develop an integrated ontological model (Gamified tutoring ontology) that connects evidence-supported game design elements in the e-learning domain as well as gamification theories and frameworks to existing ITS concepts. Finally, we develop an authoring solution (named AGITS) that takes into account these ontologies to aid teachers in the customization of gamified ITS features. We evaluate our contributions by conducting four empirical studies: (1) we perform a controlled experiment to compare OntoSPL against a well-known ontology-based feature modeling approach. The results suggest that our approach is more flexible and requires less time to change; (2) we evaluate the ontological integrated model by using an ontology evaluation method (FOCA) with experts from academic and industrial settings. The results suggest that our ontologies are properly targeting the knowledge representation roles; (3) we evaluate non-interactive versions of the designed authoring solution with 59 participants. The results indicate a positive attitude towards the use of the designed authoring solutions, in which participants agreed that they are ease to use, usable, simple, aesthetically appealing, have a well-perceived system support and high credibility; and (4) we also evaluate interactive versions (scratch and template) of our authoring solution with 41 teachers. The results suggest that teachers can use and reuse, with a high acceptance level, an authoring solution that includes all the complexity to design gamified ITS.

Keywords: Intelligent Tutoring Systems, Gamification, Authoring tools

Acknowledgements

First of all, I would like to thank God for my life and to allow me to live in a happy family that gave me unconditional support in all decisions I made in my life.

I am very thankful to my parents, Amauri Matos and Laura Matos, for always taking care of me and giving me support at all times that I needed. For all the happy moments and for all the dedication in always seeking the best for me and for my brothers. I also thank my brother Amauri Junior and my sister Laura Regina. I love you all!

I am thankful to my fiancée Beatriz Lyra who was always by my side. Bibi, despite the moments away in the period of development of this thesis, my heart has always been with you. Thank you so much for always supporting me. I want to be forever by your side. Love you so much!

To all my relatives: my grandmother Grinauria and my grandmother Maria Regina (In memory), my uncles, aunts, and cousins, who have always supported me at this stage of my life. Thank you!

To all my friends, especially my friends of life (Felipe Zanotto, Felipe Costa, Douglas Véras, also brother-in-law, and Rafael Vergetti) for the companionship and friendship of always. Thank you very much, my friends!

I thank my father-in-law, mother-in-law, my brother-in-law and my sister-in-law for treating me like a son/brother and for all the support. Thank you very much! I also thank Ronaldinho (dog) for the companionship.

I also thank my friend and companion of struggle, Ranilson Paiva, to share good and bad times during the development of this thesis. Ranilson, you were a true warrior from the beginning until the end. Thank you for everything and a hug for your whole family (Andreia, Sofia, and Pedro).

I thank my advisor, Prof. Dr. Ig Ibert Bittencourt, for all advice, always trying to keep me motivated so that I could do my best in my work. Ig is more than an advisor, he is also a friend who helped me making several important decisions for my life. Ig, I will be eternally grateful for everything!

To my friends from the Center for Excellence in Social Technologies (NEES) for all the support and friendship during the development of this thesis. I am especially grateful to Josmário for the support at several moments of this thesis, helping in the execution of activities of this work. Thanks to the people who directly helped me with this thesis, Wansel, João, Glauber, André, Armando, Wilk, Denys, Kamilla, and Rubem. Thank you also to the Professors Alan Pedro and Sean Siqueira for the valuable contributions given to this work and to my colleagues from the Federal University of Alagoas in Penedo, Professors Dalgoberto and Thyago. Thank you so much, NEES family!

Thank you!

To my friends from Campina Grande for all the support that I received on the trips to Campina Grande. Thanks especially to Danilo for helping me from beginning to the end of this Ph.D., always willing to welcome us into his house and helping with the bureaucratic issues of the Ph.D. Special thanks also to Fabio who always received us in the best way possible in his house, especially at the beginning of the Ph.D. Thank you so much, my friends!

Thanks to the COPIN staff, Rebeka, Lyana, and Paloma, for the kindness in supporting me to solve the issues regarding my Ph.D.

To my Ph.D. sandwich supervisor in Canada, Prof. Dr. Julita Vassileva, for all the insightful feedback given to my work and for all the support I received during my stay in Saskatoon. Thank you also to Ralph for all the support.

To my friends from the MADMUC lab of the University of Saskatchewan (Canada), for all the support and friendship during my stay in Canada. I am specially grateful to Kewen, Kiemute, and Yash for the feedback on my thesis work. Thank you very much!

To the Brazilian friends, I found in Saskatoon, Igor and Felipe. Thank you for pushing me to go to the gym, for the friendship, and for the funny moments!

Thanks to the mentors and professors I met in the UMAP'16 conference in Halifax (Canada). Their feedback was very important to improve this thesis.

Thanks to the professors that were examiners (either in the qualification or in the thesis defense) of this Ph.D. thesis, Professors Alan Pedro, Dalton Serey, Evandro Costa, Julita Vassileva, Márcio Ribeiro, and Seiji Isotani.

Contents

1 Introduction **1**

1.1 Context . . . 1

1.2 Motivation . . . 3

1.3 Problem . . . 5

1.3.1 General problem . . . 5

1.3.2 Technical problem and research questions . . . 6

1.4 Objectives . . . 9

1.5 Methodology . . . 13

1.6 Scope . . . 15

1.7 Thesis organization . . . 16

2 Theoretical background **18**

2.1 Intelligent tutoring systems . . . 18

2.1.1 Types of ITSs . . . 20

2.2 Gamification . . . 22

2.2.1 6D framework . . . 24

2.2.2 Brainhex player model . . . 26

2.3 ITS authoring tools . . . 28

2.3.1 Classification . . . 29

2.3.2 Design issues . . . 29

2.4 Feature modeling and software product line . . . 32

2.4.1 Feature Modeling . . . 33

2.5 Ontologies . . . 35

2.5.1 Types of ontologies . . . 36

2.5.2 Ontologies and software engineering 37
2.5.3 METHONTOLOGY 41
2.6 Concluding remarks 42

3 State of the art analysis 43
3.1 ITS and feature modeling/software product line 43
3.1.1 Review of the literature 43
3.1.2 Related works 44
3.2 Ontology-based feature modeling 45
3.2.1 Review of the literature 45
3.2.2 Related works 47
3.3 Gamification and intelligent tutoring systems 49
3.3.1 Review of the literature 49
3.3.2 Related works 50
3.4 ITS authoring tools 52
3.4.1 Review of the literature 52
3.4.2 Related works 52
3.5 Concluding remarks 54

4 Gamified ITS ontology-based feature model 56
4.1 Gamified ITS feature modeling 56
4.2 OntoSPL: an ontology-based feature modeling approach 60
4.2.1 OntoSPL description 62
4.2.2 Empirical evaluation in changing scenarios 65
4.3 Gamified ITS ontology-based feature modeling 67
4.4 Concluding remarks 70

5 Gamified tutoring ontology 71
5.1 Gamification target behaviors in e-learning context 71
5.2 GaDO: Gamification Domain Ontology 74
5.2.1 Gamification Domain Ontology (GaDO) – Core 75
5.2.2 Gamification Domain Ontology (GaDO) – Full 77

5.3 GaTO: Gamified Tutoring Ontological Model 81
5.4 Evaluation of the ontologies 84
5.4.1 Method 84
5.4.2 Procedure and participants 86
5.4.3 Results 87
5.4.4 Analysis and discussion 90
5.4.5 Threats to Validity 92
5.5 Concluding Remarks 93

6 AGITS: an authoring solution for designing gamified intelligent tutoring systems 95

6.1 Gamified ITS development process 96
6.2 Authoring computational solution 101
6.2.1 Requirements engineering 101
6.2.2 Authoring prototyping 104
6.2.3 Architectural design and implementation 107
6.3 Experiment #1: laboratory settings 112
6.3.1 Materials and methods 113
6.3.2 Procedure and participants 118
6.3.3 Results 120
6.3.4 Analysis and discussion 127
6.3.5 Threats to the validity 128
6.4 Experiment #2: with teachers 130
6.4.1 Materials and methods 130
6.4.2 Procedure and participants 135
6.4.3 Results 137
6.4.4 Analysis and discussion 141
6.4.5 Threats to the validity 142
6.5 Concluding remarks 144

7 Conclusions and future works 147

7.1 Conclusions 147

7.2 Main contributions 149
7.2.1 List of publications, papers under evaluation and papers to submit . 153
7.3 Limitations 154
7.4 Future works 156

A **183**
A.1 SLR method 183
A.1.1 Research questions 184
A.1.2 Inclusion and exclusion criteria 185
A.1.3 Sources selection and search 186
A.1.4 Quality assessment 189
A.1.5 Data extraction and synthesis 190
A.2 Quality assessment results 191
A.3 Overview of the studies 192
A.3.1 Publication year 192
A.3.2 Application context 192
A.3.3 Type of source 194
A.3.4 Research method 194
A.3.5 Country 195
A.4 RQ1: SPL in educational environments 196
A.5 RQ2: Use of ontologies along with SPL to develop educational systems . . 196
A.6 RQ3: Use of SPL to develop gamified educational systems 197

B **198**
B.1 SLR method 198
B.1.1 Research questions 198
B.1.2 Inclusion and exclusion criteria 199
B.1.3 Sources selection and search 201
B.1.4 Quality assessment 204
B.1.5 Data extraction and synthesis 205
B.2 Quality assessment results 205
B.3 Overview of the studies 206

B.3.1 Publication year 206
B.3.2 Application context 208
B.3.3 Type of source 208
B.3.4 Research method 210
B.4 RQ1: Authoring tools in ITS components 211
B.4.1 Results 211
B.4.2 Analysis and Discussion 211
B.5 RQ2: ITS types 213
B.5.1 Results 213
B.5.2 Analysis and Discussion 214
B.6 RQ3: Features for aiding authoring process 217
B.6.1 Results 217
B.6.2 Analysis and Discussion 218
B.7 RQ4: Authoring technologies 221
B.7.1 Results 221
B.7.2 Analysis and Discussion 222

List of Symbols

ITS - *Intelligent Tutoring Systems*

ENEM - *Exame Nacional do Ensino Médio*

SPL - *Software Product Line*

DGD1 - *The first demographic game design model*

DGD2 - *The second demographic game design model*

OWL - *Web Ontology Language*

RDF - *Resource Description Framework*

W3C - *World Wide Web Consortium*

FODA - *Feature-Oriented Domain Analysis*

SWEBOK - *Software Engineering Body of Knowledge*

SLR - *Systematic Literature Review*

StArt - *State of the Art through Systematic Reviews*

CSCL - *Computer-Supported Collaborative Learning*

MOOC - *Massive Online Open Course*

LMS - *Learning Management System*

IJAIED - *International Journal of Artificial Intelligence in Education*

AIED - *Artificial Intelligence in Education*

CTAT - *Cognitive Tutor Authoring Tools*

GIFT - *Generalized Intelligent Framework for Tutoring*

GaDO - *Gamification Domain Ontology*

GaTO - *Gamified Tutoring Ontology*

SDT - *Self-Determination Theory*

AGITS - *Authoring Gamified Intelligent Tutoring Systems*

List of Figures

1.1 Overview of the objectives of this thesis. Adapted from Dermeval [2016] . 13

1.2 Technology transfer model. Adapted from Gorschek et al. [2006], Wohlin et al. [2012] 14

2.1 Classical architecture of intelligent tutoring systems 19

2.2 Adapted from Werbach and Hunter [2012] 23

2.3 Outcomes of gamification. Retrieved from Hamari et al. [2014b] 24

2.4 BrainHex conceptual model. Extracted from *http://blog.brainhex.com/* . . . 27

2.5 Smartphone SPL features model in the FODA notation 34

2.6 Ontology types according to the level of dependency of particular task or viewpoint. Arrows represent the specializations relations. Adapted from Guarino [1998] 37

2.7 Taxonomy on the use of ontologies in software engineering. Based on Ruiz and Hilera [2006] 39

2.8 METHONTOLOGY states and activities. Adapted from Fernández-López et al. [1997] 42

4.1 Gamified ITS feature model 58

4.2 Configuration of the gamified ITS feature model for the MeuTutor ENEM . 61

4.3 OntoSPL classes hierarchy. Extracted from Tenório et al. [2014] 62

4.4 Relationship between the OWL files of OntoSPL, gamified ITS (GITS) feature model, and arbitrary configurations of gamified ITS 68

4.5 Ontology individuals of the gamified ITS feature model represented in GITS-PL.owl 68

5.1 Ontological model illustrating the relationship between gamification and ITS ontologies . . . 75
5.2 Excerpt of the Gamification Domain Ontology (GaDO) – Core. For the sake of clarity, we suppress the specialization for the Game Design Element types and some axioms. We use the prefix "gc" to refer to the concepts of this ontology . . . 77
5.3 Excerpt of the Gamification Domain Ontology (GaDO) – full. Some classes and relations are omitted for clarity. We use the prefix "gf" to refer to the concepts of this ontology. . . . 80
5.4 Excerpt of the Gamification Tutoring Ontology (GaTO). Some classes and relations are omitted for clarity. We use the prefix "gt" to refer to the concepts within GaTO ontology and "its" to refer to concepts from Bittencourt's ontology. . . . 83
5.5 Boxplots comparing the five scores for the three ontologies . . . 88

6.1 General gamified ITS development process . . . 96
6.2 Gamified ITS authoring . . . 98
6.3 Illustration on how the ontologies are used to interoperate the authoring solutions and a third-party gamified ITS . . . 100
6.4 Authoring solution development activities . . . 101
6.5 Authoring solution use cases . . . 102
6.6 Authoring flow execution . . . 103
6.7 Example of prototypes specified for the authoring solution . . . 104
6.8 Prototype illustrating how the tunneling persuasive strategy and reuse capabilities are designed . . . 105
6.9 Prototype for defining curriculum and subjects . . . 106
6.10 Prototype for selecting a gamification target behavior . . . 107
6.11 Prototype for selecting an educational level . . . 108
6.12 Prototype of the last step to apply a template . . . 108
6.13 Prototypes for authoring educational resources . . . 109
6.14 Authoring computation solution modules view . . . 111

6.15 Sequence diagram showing the behavior of the architecture to configure a new tutor . . . 111
6.16 Sequence diagram showing the behavior of the architecture to create a new problem . . . 112
6.17 Trials definition illustrating the flow of steps for each version. Steps with same colors are highlighted to identify that flows are using the same treatment 116
6.18 Boxplots comparing the four versions regarding perceived ease of use, perceived usability, novelty, complexity, aesthetics, attitude towards use, unity, intensity, perceived system support, and credibility. . . . 124
6.19 Boxplots comparing the understandability, complexity3, and perceived system support with respect to the two prototypes for authoring gamification 125
6.20 Trials definition of flow of tasks performed by teachers to participate of the second experiment . . . 133
6.21 Example of a gamified ITS prototype authored that can be generated in the experiment according to teachers' choices . . . 133
6.22 Boxplots comparing the two versions regarding complexity, usefulness, time, perceived ease of use, usability, attitude towards use, behavioral intention to use, perceived system support, credibility, representability, satisfaction, and utility . . . 140

A.1 Sources and selection flow of the SLR on the use of SPL in online learning environments . . . 188
A.2 Temporal view of the studies . . . 193
A.3 Distribution of papers by application context . . . 193
A.4 Distribution of papers by type of publication . . . 194
A.5 Distribution of papers by research method . . . 195
A.6 Distribution of papers by country . . . 195
A.7 Using ontologies to support SPL engineering of educational environments . 197
A.8 Using SPL along with gamified educational environments . . . 197

B.1 Paper selection flowchart . . . 202
B.2 Temporal view of the studies . . . 208

B.3 ITS types over ITS components 216
B.4 ITS components and types over technologies subareas 225

List of Tables

2.1 AI features of ITS. Retrieved from Woolf [2010] 21

2.2 ITS authoring tools categories. Extracted from Murray [2003] 30

3.1 Comparison of our proposal to related works with respect to the research questions of the SLR 44

3.2 Research questions and motivations. Extracted from Dermeval et al. [2015b] 46

3.3 Comparison of our ontology for conceptualizing feature model and related works 49

3.4 Comparison of our ontology for conceptualizing feature model and related works 50

3.5 Comparison of our authoring solution against related works 54

4.1 Summary of statistics of the metrics evaluated (O1 = Ontology by Wang et al. [2007], O2 = OntoSPL). Time is measured in milliseconds, impact is measured by the total number of ontology elements changed, and the correctness is a ratio between the number of correct steps performed from participants and the total number of correct steps. Adapted from Dermeval et al. [2015a] 66

4.2 P-values after applying Wilcoxon tests (O1 = Ontology by Wang et al. [2007], O2 = OntoSPL). Adapted from Dermeval et al. [2015a] 67

5.1 Summary of target behaviors and game design elements 73

5.2 Goals, questions and metrics (along with a range of possible scores) of the FOCA methodology. Adapted from Bandeira et al. [2016]. 85

5.3 Participant experience per each topic and settings 87

5.4 Summary of statistics of the five metrics per each ontology evaluated 88
5.5 Hypotheses of the evaluation 89
5.6 P-values after applying Kruskal-Wallis and Wilcoxon tests (O1 = GaDO-core, O2 = GaDO-full, O3 = GaTO). 90% and 95% confidence levels are represented, respectively, by * and ** 90

6.1 Factors levels 114
6.2 Questions used to measure the constructs. The score of each construct is computed by the average of the questions using a Likert scale from 1 (completely disagree) to 7 (completely agree). Credibility is measured in a scale from 1 to 9. 117
6.3 Hypotheses of the first experiment 119
6.4 Formal definition of the research hypotheses 120
6.5 Participant demographics 121
6.6 Summary of statistics and normality tests for the ten metrics evaluated per version 123
6.7 Summary of statistics and normality tests for the three metrics evaluated per steps with respect to the two ways for authoring gamification 125
6.8 P-value results for the hypotheses of this experiment 126
6.9 Factors levels 131
6.10 Questions used to measure the constructs. 134
6.11 Hypotheses of the second experiment 135
6.12 Formal definition of the research hypotheses' second experiment. V1 = scratch and V2 = template 136
6.13 Participant demographics of the second experiment 138
6.14 Summary of statistics and normality tests for the ten metrics evaluated per version 139
6.15 P-value results for the hypotheses of the second experiment 141

A.1 Research questions and motivations 184
A.2 Inclusion/exclusion criteria 185
A.3 Study quality assessment criteria 189

A.4 Extraction form 190
A.5 List of papers included in the review along with their quality scores 191
A.6 Distribution of works using SPL over educational environments 196

B.1 Research questions and motivations 199
B.2 Inclusion/exclusion criteria 200
B.3 Study quality assessment criteria 204
B.4 Extraction form 206
B.5 List of papers included in the review along with their quality scores 207
B.6 Distribution of studies over publication sources. 209
B.7 Authoring tools in ITS components 211
B.8 Authoring tools in ITS Types 213
B.9 Features for aiding authoring process 217
B.10 Technologies used to build authoring tools 222

Code Listings

4.1 Excerpt of the MeuTutor-ENEM.owl 69

Chapter 1

Introduction

In this chapter, we present the context (Section 1.1) and motivation (Section 1.2) of this work as well as the research problem (Section 1.3) we are targeting. Next, we describe the objectives (Section 1.4) and the methodology used in the conduction of this thesis (Section 1.5). We also bound the scope of this work in Section 1.6 and finally describe how this document is organized in Section 1.7.

1.1 Context

Intelligent Tutoring Systems (ITSs) have been drawing the attention of academics and practitioners since early 70's [Woolf, 2010]. These systems are concerned with the use of artificial intelligence techniques for performing adaptive tutoring to learners according to what they know about the domain [Sleeman and Brown, 1982]. As reported by du Boulay [2016], there have been a number of recent positive reviews in support of the effectiveness of ITSs [Kulik and Fletcher, 2015, Ma et al., 2014, Steenbergen-Hu and Cooper, 2014, 2013, VanLehn et al., 2011]. Thus, it is well known that well-designed ITS can successfully complement and substitute other instructional models (e.g., human tutoring or computer-aided instruction) at all educational levels and in many common academic subjects [Ma et al., 2014].

From the learner perspective, positive empirical evidence is consistent with the most frequently implemented ITS features enabled by student modeling, namely high individualized task selection, prompting and response feedback [Ma et al., 2014]. However, as argued by Arroyo et al. [2007], Baker et al. [2010], Bell and McNamara [2007], Jackson and McNamara [2013], it is very common that students become disengaged or bored during the learning process by using ITSs, particularly in a long-term period of instruction. Moreover, for those students who continue to interact despite lack of interest, boredom may trigger a vicious cycle that prevents them from actively reengaging in constructive learning processes (Baker, Corbett, & Koedinger, 2004; D'Mello, Taylor, & Graesser, 2007). By contrast, motivated, challenged and intrigued students tend to have better learning results [VanLehn, 2011]. Hence, ITSs may benefit from design features that enable proper intervention to enhance student motivation and engagement during instruction [Jackson and McNamara, 2013, Woolf, 2010].

In this way, relying on several theories and models of motivation and human behavior (e.g., Fogg's behavior model [Fogg, 2009], need theories [Goble, 2004] [Alderfer, 1969, Gagné and Deci, 2005] and Skinner's reinforcement theory [Skinner, 2011]), many works have been using persuasive technologies in connection with education [Hamari et al., 2014a]. These technologies intend to change human behavior through the use of computers [Fogg, 1999, King and Tester, 1999]. As such, researchers have been using different persuasive techniques aiming to address educational contexts, for example, goal setting, (self-) monitoring, feedback, rewards, competition and so on [Hamari et al., 2014a, Masthoff and Vassileva, 2015, Michie et al., 2008].

Considering current persuasive technologies, it is noteworthy that game-based approaches include a diverse set of these technologies that are effective to engage participants and to change behavior [Berkovsky et al., 2010]. For instance, in 2010, users had spent about three billion hours per week playing different types of games [McGonigal, 2010]. In educational settings, games have been used to change students' attitude, to develop good habits or to learn [Masthoff and Vassileva, 2015]. Although digital games are immersive environments that may be effective to drive behavior changing, the cost of constructing such kind of system is high [Economist, 2014]. Thus, researchers have been investigating the application of gamification at different areas, including online education [de Sousa Borges et al., 2014, Hamari et al., 2014b, Seaborn and Fels, 2015], as an alternative way that takes advantage of the power of games, but with a better cost-effectiveness, to address the students' disengagement and lack of motivation problems [Hamari et al., 2014a].

Deterding et al. [2011] define gamification as the use of game design elements in non-game contexts. These contexts (e.g., education, e-commerce, healthcare, and so on) mostly converges to a common final objective, the use of gamification to engage and motivate users to achieve better results and create enhanced solutions and experiences [Hamari and Tuunanen, 2014]. In the educational context, gamification may motivate action, promote learning, and facilitate problem solving [Seaborn and Fels, 2015] as well as drive desired learning behaviors [Kapp, 2012].

To explicitly consider students' motivational aspects, recent works are increasingly interested in using gamification along with ITS [Andrade et al., 2016, González et al., 2014, Shi and Cristea, 2016]. Applying gamification to ITS must deal with the development of the four classic ITS components (i.e., domain, student, tutoring, and interface) as well as a gamification model in order to connect concepts, theories, and technologies from both topics.

1.2 Motivation

Although the increasing interest in applying gamification to ITS contexts, expecting to benefit from both areas to provide adaptive instruction with explicit focus on learners' motivation. According to Woolf [2010], teachers might take advantage of traditional ITS to gain insight into students' learning processes, to spend more time with individual students, to save time by letting the tutor correct homework, to identify and predict knowledge gaps, to personalize instruction and tailored content to each student's individual learning path, and so on [Woolf, 2010].

Despite providing individualized tutoring to students and some kind of support for teachers, one might note that teachers have been not considered as first-class citizens in the development of these kinds of systems [Lemann, 2015, ProjectTomorrow, 2013, 2014]. In general, ITSs are not personalized for teachers (i.e., one-size-fits-all approach). However, teachers have different expectations and/or methodologies as well as could use ITSs in different contexts, i.e., domains, educational levels, and features. In face of this, we can say that teachers are much more passive than active in the design of these systems.

Meanwhile, teachers are increasingly demanding to act as active users of adaptive and intelligent educational systems. For instance, a recent survey [ProjectTomorrow, 2014] with 41,805 K-12 teachers in USA reports that more than half of them consider learning how to use educational technologies which distinguish instructions to students (i.e., ITS) the most important item for their professional development. Moreover, another survey [ProjectTomorrow, 2013] with aspirants teachers in USA reports that they consider the access to educational technologies with support to customized instructional plans as one of the main factors that will determine their future success as teachers. With respect to the Brazilian context, a recent survey with 1,000 public middle-school teachers points out that more than 80% of the teachers that were included in the survey rely on the potential of students' adaptive learning as well as in the support of technologies to monitor students' learning process. Ninety-two percent of the teachers also demand training to use educational technologies [Lemann, 2015].

Furthermore, there is also a current and relevant discussion about the intelligence nature of tutoring systems. Baker [2016] argues that the tutoring systems that are currently being used at scale are much simpler than the initial vision of ITS. He also raises the possibility that we need "stupid tutoring systems" that are augmented with human intelligence. It means that we probably need tutors that are designed intelligently, and that leverage human intelligence, rather than relying only on artificial intelligence. To leverage human intelligence, humans should be involved as early as possible in ITS design. Hence, a natural way to accomplish it is relying on stakeholders such as teachers since the beginning of an ITS design and throughout the instruction life-cycle.

Regarding teachers' attitudes towards use of games to engage students in the context of personalized learning, in the aforementioned survey [ProjectTomorrow, 2014], it is reported that teachers say games enable them to address various learning styles (70% of more than 40,000), differentiate instruction (60%), and create classrooms that are more learner centric (44%). Moreover, teachers also say that by using gamification they can motivate students because of the intrinsic entertaining nature of games and can facilitate students' learning via entertainment and a higher motivation [Martí-Parreño et al., 2016, Sánchez-Mena and Martí-Parreño, 2016]. These studies also present that teachers show a positive attitude towards gamification, but there is not an intense use it their courses. On the other hand, it was also found that a main barrier preventing teachers to use gamification is the lack of time and other resources (e.g., classroom setting) available and, in particular, lack of time to prepare materials and training in gamification [Martí-Parreño et al., 2016, Sánchez-Mena and Martí-Parreño, 2016].

In this way, as teachers play a key role in introducing pedagogical innovations in the classroom, they deserve to be considered as active users of gamified ITSs. By active participation we mean that teachers may be primary actors of gamified ITSs, for example, by selecting which functionalities they are interested to incorporate in ITSs, by defining which gamification behaviors they expect from their students, by choosing which pedagogical strategies they may consider or by creating and/or reusing content. Thus, contributing to the active participation of teachers in the use of intelligent tutoring systems that consider motivational aspects of the students by using gamification is of utmost importance to amplify teachers' participation in the development process of gamified ITS.

1.3 Problem

All things considered, in this section we formulate the problem we are targeting in this thesis. Our problem is divided into two perspectives: (i) **general problem**, which is broader, encompasses the articulation of several researches in different knowledge areas and when solved directly impacts the society (e.g., teachers and students); (ii) **technical problems**, which are more specific to the computer science field, particularly, in the computers and education/artificial intelligence in education area, and contribute to the partial resolution of the general problem addressed in this work.

1.3.1 General problem

The general problem we are addressing in this work is the following: "How could we support teachers to use gamified intelligent tutoring systems in an active and personalized way?". Despite the interest of teachers in using intelligent tutoring systems and gamification in the context of their pedagogical interventions, they are not actively included in the design process of these systems. This general problem must consider several aspects, which might include educational (e.g., supporting the use of gamified ITS in classrooms) and technological perspectives (e.g., providing information and technology tools to aid teachers actively use this kind of system).

1.3.2 Technical problem and research questions

In order to contribute to the active and personalized use of gamified ITS by teachers, we are considering three dimensions. First of all, one might note that the design of traditional ITS (with no gamification) is sufficiently complex. There are theories and technologies from interdisciplinary areas (e.g., computer science, psychology, and education) that must be considered in the design of ITS [Woolf, 2010] as well as it should take into account the four classic ITS components such as domain (what to teach), student (to whom teach), pedagogical (how to teach) and interface models (how to communicate with learners) [Sleeman and Brown, 1982, Sottilare et al., 2015, Woolf, 2010].

Traditional ITS development time estimations show that 200-300 hours of authoring are needed for 1 hour of instruction with students [Aleven et al., 2006]. Moreover, designing ITS also has to consider different stakeholders: (i) developers, to implement software functionalities; (ii) authors, to personalize the execution in an specific context; and (iii) final users, which are not concerned with the system complexity and demand a friendly graphical interface to interact with the system [Silva et al., 2011].

The inclusion of gamification features in ITS design significantly increases the complexity of constructing these systems. Besides considering the variable software requirements (technological perspective) and different educational strategies (pedagogical perspective) of systems, several gamification elements (e.g., badges, points, leaderboard, avatar, and so on) could be combined to aspects of each one of the others perspectives. Thus, it is noteworthy that the design of gamified ITSs should deal with a huge variability of features.

To motivate the high variability presented in gamified ITS, we mention an example of a system named *Meu Tutor*[1] (in english, My Tutor). It is a gamified ITS that aims to help high-school Brazilian students to be prepared to take the high-school national exam (called ENEM[2]). Regarding the technological perspective, there are more than fifty features (e.g., login, register, social integration, evaluation, reports and so on) provided by the system, where at least fifteen of them can be optionally included in a particular configuration of the system. Considering the pedagogical perspective, it uses a problem-based learning strategy

[1] http://enem.meututor.com.br/

[2] This exam is used by public and private universities in Brazil to select the entry of new students in college.

and takes into account ten courses (related to the high-school exam domain, for instance, math, physics, biology, chemistry and so forth) and around twenty subjects per course. Moreover, in the current configuration of Meu Tutor, there are six gamification elements (e.g., points, levels, badges, mission, leaderboard and progress bar) that could be combined with each other resulting in a total of sixty-three possibilities (using the equation $2^n - 1$, where n is the number of gamification elements). Hence, multiplying the number of optional features, disciplines, subjects, and number of gamification elements combinations, leads to a total of 189,000 possible combinations of configurations that a system like *Meu Tutor* could have to manage – this is the maximum number of combinations.

Note that the aforementioned design space is related to a single gamified ITS, i.e., Meu Tutor. When considering the features of other gamified intelligent tutoring systems (e.g., Duolingo[3] and Knewton[4]), this variability could be even higher. In this way, it would be important to identify such variability design space taking into account different gamified ITS platforms. Moreover, enabling management of this variability space by third-party systems would aid the design of independent-platform systems.

Thus, in order to deal with the variability issues for designing gamified ITS, we present the first technical research question of this thesis, which is: **"How could we identify and manage the variability of gamified ITS features?"**.

A second dimension that we may consider in the business problem targeted in this thesis is related to the huge design space aforementioned. Considering such a huge design space and that all combinations might not be necessarily effective for students' learning and motivation since some features may be more or less amenable depending on several elements such as educational level, knowledge domain, ITS and gamification theories, and design principles; asking teachers to customize gamified ITS under these circumstances would be very confusing, demotivating and not helpful at all for them.

To constrain the design space in order to aid teachers to actively customize gamified ITS, it is imperative to take advantage of theories from both topics. ITSs are knowledge-intensive systems that handle knowledge about the domain of the tutor, students' behaviors, tutoring theories, and so on [Dillenbourg and Self, 1992, Du Boulay and Luckin, 2001, Self, 1990,

[3]https://www.duolingo.com/
[4]https://www.knewton.com/

1998]. Moreover, the inclusion of gamification generates extra knowledge to handle. Gamification is supported by several concepts and theories, i.e., motivation theories (e.g., self-determination theory [Deci and Ryan, 2010]), player models (e.g., Brainhex [Nacke et al., 2014]), and gamification design frameworks (e.g., 6D framework [Werbach and Hunter, 2012]). Thus, the application of gamification in ITS must deal with knowledge from both topics.

Furthermore, there are mixed results on the effects of game elements for different contexts (e.g., education) [de Sousa Borges et al., 2014, Hamari et al., 2014b, Nacke and Deterding, 2017, Seaborn and Fels, 2015]. In this way, as stated by Masthoff and Vassileva [2015], there is a need to personalize gamification elements for the following reasons: (i) people are motivated by different things and pursue different goals; (ii) a method that motivates one type of person may actually demotivate a different type of person and (iii) there are mixed findings and unexpected failures of gamification. For instance, a system that uses a leaderboard to show high scores may encourage a competitive player, but may discourage players who do not thrive in a competitive environment. Thus, in order to support teachers to customize gamification aspects in the context of ITS, it would be also important to rely on empirical evidence about the effect of game elements on students' learning performance and motivation [Nacke and Deterding, 2017].

In this way, either the knowledge about gamification and ITS theories or about gamification empirical evidence in education context should be considered to effectively constrain the design space of gamified ITS. Additionally, all this knowledge might be represented in a way that allows automated reasoning in order to leverage this knowledge to aid teachers customizing gamified ITSs. As a result, we present the second technical research question of this thesis, which is: **"How could we constrain the design space of gamified ITS making use of gamification and ITS theories as well as design principles?"**.

The third dimension we are considering to target the general problem presented is the simplicity and usability to customize gamified ITS by teachers. To effectively enable participation of teachers in the design process of these systems, providing these qualities is imperative [Dağ et al., 2014, Murray, 2004, Sottilare, 2015]. However, in order to enable to teachers to feel in the control of the design process, it is also important to provide a fair level of flexibility to allow teachers customize gamified ITS according to their preferences.

Hence, trade-off issues between usability and flexibility must be considered in the design process [Murray, 2003, Woolf, 2010].

Thus, as previously mentioned, designing these systems present a huge variability and not all feature combinations might be necessarily effective for learners. In this context, gamification and ITS theories and design practices should also be considered to constrain the design space based on such knowledge. In this context, assuming that a teacher intends to customize such a complex system with this huge variability for his/her own educational context taking advantage of the knowledge about gamification and ITS theories as well as gamification design practices, we could not expect from him/her to have advanced technical skills, for instance, on programming, artificial intelligence and/or software engineering.

As a result, to address these issues, we describe our third technical research question, which is: **"How could we design a computational solution considering gamification and ITS theories as well as design practices to aid teachers deal with the high variability of customizing gamified ITS features in a simple and usable way and with no advanced technical skills?"**.

1.4 Objectives

Considering the presented research questions, we present some theoretical concepts as well as important technologies that are used to target our technical problem. Then, we describe the objectives of this work.

The concept of Software Product Line (SPL) [Clements and Northrop, 2001] [Pohl et al., 2005], from software engineering research, has been drawing attention of academics and practitioners promoting to offer characteristics such as rapid product development, reduced time-to-market, quality improvement, and more affordable development costs. A software product line is a set of software systems that have a particular set of common features and that satisfy the needs of a particular market segment or mission [Clements and Northrop, 2001]. In comparison to other reuse strategies, for instance frameworks, services and components, SPL may be more efficient since its reuse is systematically designed and there is a way to customize the production of software from a same family [Helferich et al., 2007]. In this context, considering the huge variability presented in gamified ITS and also the need to personalize components of all three perspectives (technological, pedagogical and motivational) of it, the use of an SPL-inspired approach appears to be appropriate and promising in order to aid the customization of gamified ITS.

Feature modeling [Kang et al., 1990] is one of the key activities involved in the design of SPLs. It is broadly used to support variability management of SPLs in order to represent common and variable functionalities of a software family as well as to be used to instantiate applications based on SPL. In general, a feature model is produced to represent the commonalities and variabilities of SPLs. In order to deal with the high variability of gamified ITS, such activity could be performed to identify common and variable features of these systems in a manageable way. However, a gamified ITS in a specific context may demand different requirements, pedagogical strategies and gamification elements. Thus, allowing a particular gamified ITS to be reconfigured at runtime to change, for instance, a pedagogical strategy, can improve the flexibility of a system to be adapted to fluctuations in teachers needs.

In this way, enabling the automatic analysis of feature models and hence providing the automatic management of the gamified ITS variability would allow automated reasoning/changing at runtime. Thus, when comparing the mechanisms for automatic analysis of features models (i.e., propositional logic based analysis and constraint programming based analysis) [Benavides et al., 2013], description logic (DL) based methods (i.e., ontology-based feature modeling) promise to provide improved automated inconsistency detection, reasoning efficiency, scalability and expressivity [Benavides et al., 2010, Wang et al., 2007]. In this way, to allow automatic analysis of gamified ITS feature model, an ontology-based feature modeling approach could also be used.

Ontologies have gained significant attention by the computer science community since they aim to solve one of the biggest problems that arises when using machines to reason on information generated by human agents – they try to reach the formal representation of a real domain by using computational systems [Hepp et al., 2007]. Ontology is defined as "explicit specification of a conceptualization" [Gruber, 1993]. It is "explicit" because of its classes and properties visibility. Conceptualization is understood to be an abstract and simplified version of the world to be represented. Moreover, ontologies can be logically reasoned and shared within a specific domain [Guarino, 1998]. Thus, ontologies are a standard form for representing the concepts within a domain, as well as the relationships between those concepts in a way that allows automated reasoning.

Ontology is considered as one of the most appropriate ways to facilitate the interoperability between heterogeneous systems involved in a domain of common interest. This is true especially because ontologies offer a shared understanding of a particular domain and a formalization that allows its data to be interpretable by machines [Hepp et al., 2007]. In this way, considering a variability model (i.e., feature model) of gamified ITS, the use of ontologies might be used to enable its management (i.e., reasoning) by different gamified tutors.

There is also a growing interest on the use of ontologies to address e-learning problems. Particularly, in the context of ITS, ontologies have been used to represent domain model concepts, to represent students' modeling allowing automated reasoning, to interoperate heterogeneous ITSs, and so on [Al-Yahya et al., 2015]. As previously explained, gamified ITSs are knowledge-intensive systems that handle knowledge about the domain of the tutor, students' behaviors, tutoring theories, and so on. Formally representing gamification and ITS theories by using ontologies could provide several benefits to the design of gamified intelligent tutoring systems. It could allow the automated reasoning of all knowledge manipulated by these systems, which could also favor machines to automatically handle it. It might also provide a standard representation for the infrastructure of gamified ITSs, which may enable the interoperability (e.g., to interoperate educational resources) between different architectures of these systems. Furthermore, it may also leverage the transparency of the theories used to design these systems as well as allowing representing design practices for applying gamification in ITS – i.e., the later benefits could be very useful to aid teachers customizing gamified ITS.

Due to the high cost for designing ITS, for many years, researchers are developing ITS authoring tools in order to speed up ITS development, to reduce production efforts, to decrease the level of ability needed to build ITS, to support good design principles, to increase the number and diversity of available tutors, to extend the number of participants in ITS development process and so on [Murray, 2003, Sottilare et al., 2015, Woolf, 2010]. Although researchers' interests in the development of ITS authoring tools, the inclusion of a gamification model may require new authoring tools in order to effectively deliver gamified

ITSs. In this way, the development of a gamified ITS authoring tool that automatically relies on gamification and ITS theories and on gamification design practices to constrain the design space of gamified ITS would leverage teachers' participation in the design of these systems.

To answer the research questions presented and considering the concepts and technologies previously explained, the main objective of this thesis is to design and implement an authoring solution in order to provide for teachers a way to actively customize gamified ITS features. This platform takes into account an ontology-based feature model to deal with the high variability of these systems at runtime as well as an integrated ontological model to consider theories and gamification design practices for designing gamified ITS. In following we present our specific objectives according to the research questions they are targeting:

RQ1: How could we identify and manage the variability of gamified ITS features?

- **(O1)** Define a reference feature model for representing the variability of gamified intelligent tutoring systems;
- **(O2)** Conceptualize an ontology for representing feature models and represent the reference feature model using it;

RQ2: How could we constrain the design space of gamified ITS making use of gamification and ITS theories as well as design principles?

- **(O3)** Identify evidence-supported combinations of game design elements that might be more amenable to be effective for achieving particular behaviors in the e-learning domain;
- **(O4)** Design and develop a gamification domain ontology considering theories, frameworks, and design practices;
- **(O5)** Propose and develop an integrated ontological model that connects the gamification domain ontology with existing ITS ontologies;

RQ3: How could we design a computational solution considering gamification and ITS theories as well as design practices to aid teachers deal with the high variability of customizing gamified ITS features in a simple and usable way and with no advanced technical skills?

(O6) Design an authoring solution to provide for teachers a way to customize gamified ITS features taking into account our ontology-based feature model as well as our integrated ontology model.

Figure 1.1 presents an overview of our work, mapping the specific objectives of this thesis.

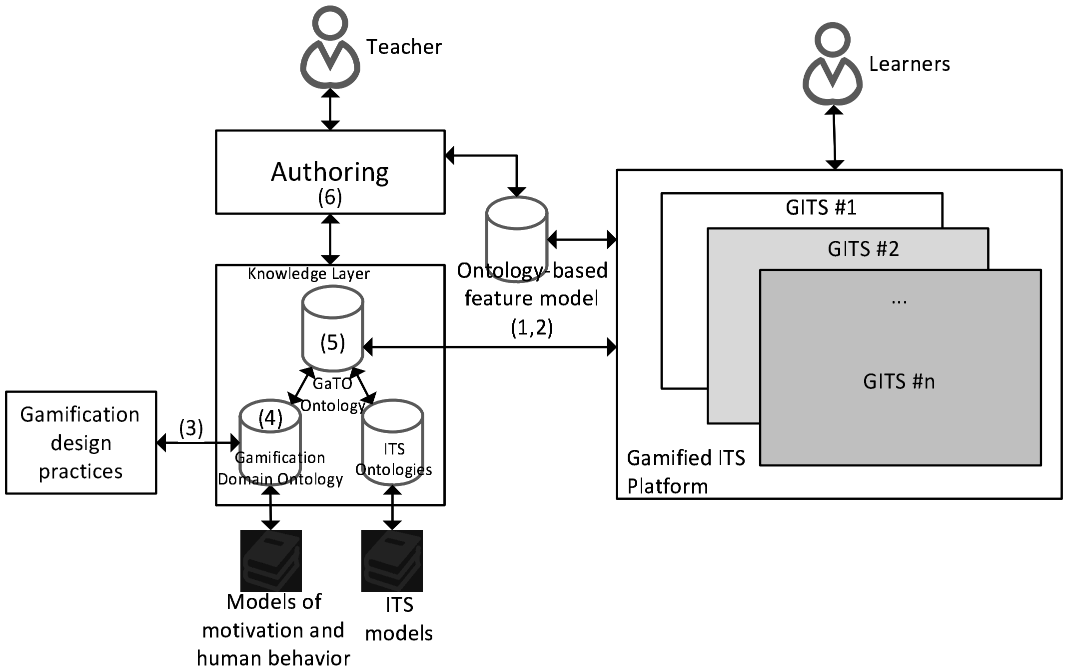


Figure 1.1: Overview of the objectives of this thesis. Adapted from Dermeval [2016]

1.5 Methodology

This work was raised from the identification of a problem in the industry. Thus, we follow in this thesis a methodology that is based on a technology transfer model presented by Gorschek et al. [2006]. This model is illustrated in the Figure 1.2 and in following we describe the seven steps that are part of the model. The main focus of this model is to use different empirical methods to solve a real problem from industry.

Step 1: Identification of industrial problem/issue. In this step, the industrial problem/issue is identified. The objective of this step to capture the challenges

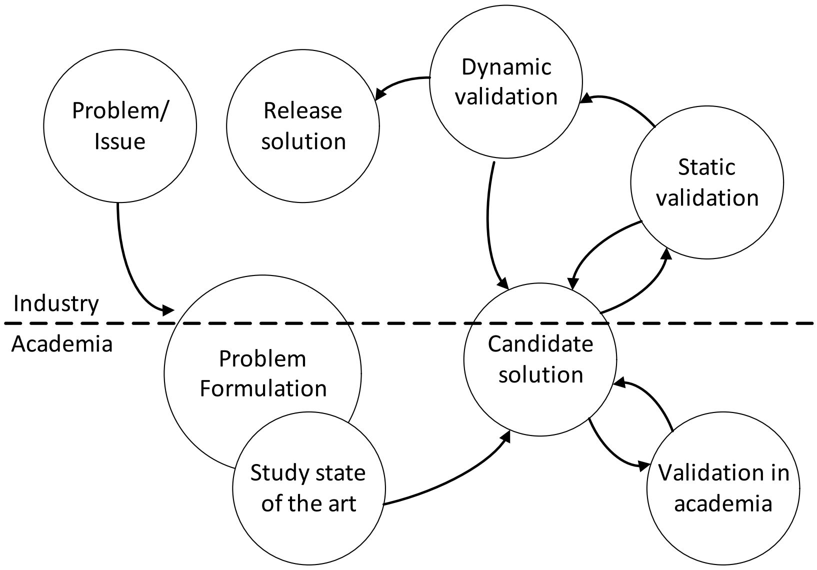


Figure 1.2: Technology transfer model. Adapted from Gorschek et al. [2006], Wohlin et al. [2012]

and particular questions suitable for research. A major benefit of doing this step thoroughly is that it creates an opportunity to build a joint trust and ensures that the industrial partner(s) and its employees get used to having researchers present in their environment [Wohlin et al., 2012];

Step 2: Problem formulation. Based on the identified challenges, a specific challenge must be formulated as a research problem and research questions are defined. According to Wohlin et al. [2012], as a natural part of the formulation of the research problem, the researchers conduct a literature search (e.g., performing a systematic literature review). A literature survey is needed to know about existing approaches to the identified industrial challenge. It provides a basis for understanding the relationship between approaches available and the actual industrial needs;

Step 3: Candidate solution. Based on existing approaches and industrial needs, a candidate solution is developed and may include the adaptation of current processes, methods, technologies and tools used in the company. The solution is preferably developed in close collaboration with the company so that the applicability can be continuously ensured. Although a specific solution for a company may be derived, the intention of the researcher is to develop a generic solution, which then is instantiated in a specific context [Wohlin et al., 2012];

Step 4: Validation in academia. The first validation of the proposed solution is conducted in academic settings to minimize risk. Such validation can be performed through controlled experiments or case studies. Both students or industrial partners can be subjects of this validation;

Step 5: Static validation In this step, industry representatives evaluate the candidate solution off-line, i.e., internally and not with final users. Based on this evaluation, the candidate solution may be changed according to the received feedback. The seven steps are iterative and, hence, may not be seen as a cascade model with no feedback cycles;

Step 6: Dynamic validation Once the new solution is statistically validated and there's an agreement and compromise to implement the new solutions, it is time to move forward to the dynamic validation. The new solution may be used in a project, a subproject or for parts of a system, or for a specific activity. Wohlin et al. [2012] recommends the conduction of case studies in this step;

Step 7: Release solution This step is not primarily the responsibility of the researchers, but they must support their collaborative partners to support the transfer of the new solution to the organization before moving to the next industrial challenge [Wohlin et al., 2012].

Note that, as described in the next section, although our work is following this methodology, it is out of scope of this thesis performing the Steps 5, 6 and 7.

1.6 Scope

The scope of this thesis is constrained to address the objectives defined in Section 1.4. However, we point out on below some objectives that are not in the scope of this work:

- Targeting authoring for other types of educational systems beyond intelligent tutoring systems. This work might also not be applied to all types of ITS, as will be further explained throughout this thesis;
- Considering authoring for intelligent tutoring systems that use other persuasive technologies, beyond gamification. Particularly, it is out of scope to address authoring of game-based intelligent tutoring systems;
- Target adaptation of instruction using gamification according to learners' characteristics. Note that although our ontological model considers different players types, we are not considering this objective in the scope of this thesis;
- Verify the quality of gamified ITS authored by teachers. Notice that our ontology models intend to include gamification and ITS theories and design practices to support the effective design of gamified ITS, however, it is out of the scope of this thesis to assess the quality of the authored gamified tutors with students;

1.7 Thesis organization

The remaining of this document is organized into the following chapters:

Chapter 2. Theoretical background: in this chapter, we present the main theoretical concepts and technologies used in this thesis, which include background about intelligent tutoring systems, gamification, ITS authoring tools, feature modeling and software product line, and ontologies.

Chapter 3. State of the art analysis: this chapter describes how we investigated the literature and discusses the main works that are related to our works, which were identified through literature analysis.

Chapter 4. Gamified ITS ontology-based feature modeling: this chapter presents the reference feature model for gamified ITS that we propose as well as the ontology-based feature modeling approach developed.

Chapter 5. Gamified tutoring ontology: this chapter includes the objectives related to the identification of gamification design practices, the definition of a gamification domain ontology, and an integrated ontological model that connects gamification to existing ITS ontologies.

Chapter 6. An authoring tool for designing gamified intelligent tutoring systems: in this chapter, we present the authoring tool that we have developed to enable teachers to customize gamified ITS considering the ontology-based feature model for gamified ITS and the gamified tutoring ontology.

Chapter 7. Conclusions and future works: Finally, this chapter presents our final considerations, pointing out our contributions and limitations, besides describing our future works.

Chapter 2

Theoretical background

In this chapter we present the main theoretical concepts and technologies used in this thesis. We describe in the following sections concepts regarding intelligent tutoring systems (Section 2.1), gamification (Section 2.2), ITS authoring tools (Section 2.3), feature modeling/software product line (Section 2.4), and ontologies (Section 2.5)

2.1 Intelligent tutoring systems

Sleeman and Brown [1982] define Intelligent Tutoring System as a computer-based program that uses artificial intelligence to represent knowledge and to conduct an adaptive interaction with students. According to their definition, an ITS should have in its basic structure features such as, (i) what to teach, (ii) how to teach and (iii) teaching for whom. Shute and Psotka [1994] noted that almost all researchers agreed that the most critical feature provided by ITSs is the student modeling. The next most frequently cited feature is adaptive behavior.

A broader definition [Ma et al., 2014] qualify ITS as a computer system that for each student: (i) performs tutoring functions, for example, by presenting information to be learned, by asking questions or assigning learning tasks, by providing feedback or hints, by answering questions posed by students or by offering prompts to provoke cognitive, motivational or metacognitive change; (ii) computes inferences from students responses and constructs either a persistent multidimensional model of the student's psychological states or locates the student's current psychological state in a multidimensional domain model; and (iii) uses the student modeling functions to adapt one or more of the tutoring functions.

It is generally accepted that an ITS has four major components [Nkambou, 2010, Sleeman and Brown, 1982, Sottilare et al., 2013, 2015, Woolf, 2010]: the domain model, the student model, the tutoring/pedagogical model, and the user interface model. Figure 2.1 presents the classic ITS architecture considering these components.

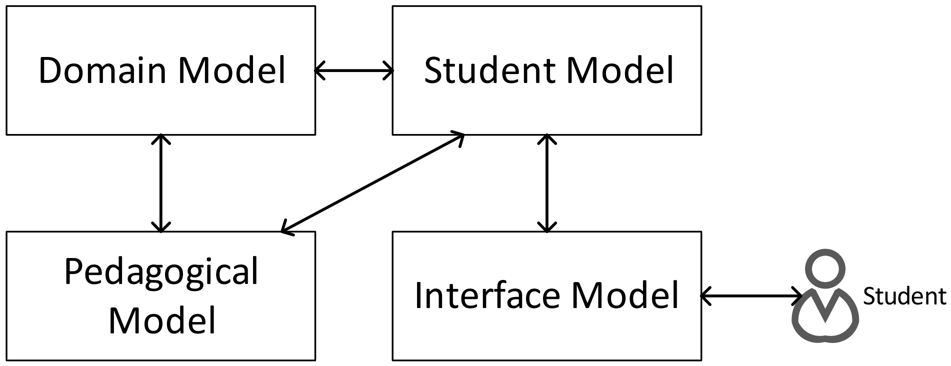


Figure 2.1: Classical architecture of intelligent tutoring systems

These components are described as follows:

1. A domain model represents the knowledge the student is intended to learn. It normally contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit. The model is a set of logical propositions, production rules, natural language statements, or any suitable knowledge representation format (e.g., ontologies). Generally, it requires significant knowledge engineering to represent a domain so that other parts of the tutor can access it.

2. A student model represents relevant aspects of the student's knowledge determined by the student's responses to questions or other interactions with the interface. There are many methods for representing information about the student. Two commonly used techniques are overlay models and Bayesian networks [Beck et al., 1996]. In the overlay model, student's knowledge is considered to be a subset of the expert's knowledge, whereas bayesian networks probabilistically reason about a student's knowledge state based on his interactions with the tutor. Each node in the network has a probability indicating the likelihood of the student "knowing" that piece of knowledge.

3. A tutor model (also known as pedagogical model or instructional model) represents instructional strategies. It takes the domain and student models as input and selects tutoring strategies, steps, and actions on what the tutor should do next in the exchange. It is a model of how someone skilled in a particular domain represents the knowledge.

4. An interface model interprets the learner's contributions through various input media (speech, typing, clicking) and produces output in different media (text, diagrams, animations, agents). This model is often constrained to the subject domain (e.g., algebra).

2.1.1 Types of ITSs

There are several ways of categorizing ITSs, we concentrate this classification on the functionalities that tutors provide, as presented by Woolf [2010]. Table 2.1 presents and describes seven artificial intelligence-based features that may be included in ITSs. In fact, few tutors have all these functionalities and, to provide them, more researches are needed. For instance, to provide a complete student modelling, it is necessary that tutors reason on the human affective states (e.g., motivation, confidence, and engagement), besides reasoning on students' cognition.

The first feature presented in the table, generativity, is the ability of generating proper resources (i.e., customized problems, hints, or help) based on representing subject matter, student knowledge, and human tutor capabilities. The second and third features are, respectively, student modeling (dynamically recording learned tasks based on student action) and expert modeling (representing topics, concepts, and processes of the domain). Student modeling may be seen as the student model component of ITSs, previously described. In a similar way, the third feature is equivalent to domain model component.

The fourth feature is mixed initiative, i.e., the ability for either student or tutor to take control of an interaction [Woolf, 2010]. Most of intelligent tutors are mentor-driven, for example, they define an agenda, ask questions, and determine the path students will take through the domain. The implementation of this functionality supports students to ask novel questions and set the agenda, and typically requires the understanding and generation of answers in natural language. The fifth functionality is interactive learning, i.e., being

Table 2.1: AI features of ITS. Retrieved from Woolf [2010]

ITS Feature	**Description**
Generativity	The ability to generate appropriate problems, hints, and help customized to student learning needs
Student modeling	The ability to represent and reason about a student's current knowledge and learning needs and to respond by providing instruction
Expert modeling	A representation and way to reason about expert performance in the domain and the implied capability to respond by providing instruction
Mixed initiative	The ability to initiate interactions with a student as well as to interpret and respond usefully to student-initiated interactions
Interactive learning	Learning activities that require authentic student engagement and are appropriately contextualized and domain-relevant
Instructional modeling	The ability to change teaching mode based on inferences about a student's learning
Self-improving	A system's ability to monitor, evaluate, and improve its own teaching performance based on its experience with previous students

responsive to student's learning needs. This feature is strongly related to the way students communicate with the tutor, hence, it is closely related to the interface model component of ITS.

The sixth feature is the instructional modeling, which may be equivalent to the pedagogical model of ITS. This feature defines how the tutor modifies its guidance for each student. Instructional modeling receives as input a student model, because students with less prior domain knowledge clearly require more instructional and guidance than do students with more knowledge. The seventh feature is the self-improving, or modifying the tutor performance based on experience with previous students. This feature is frequently implemented using machine learning and data mining techniques that evaluate previous students' learning experiences, judge which interventions are effective, and use this information to change tutor responses.

Although the agreement on the four main ITS components that these systems should include, there is no agreement in the literature on which features, processes, methods, and so on; to include in tutors. In this way, in addition to this classification provided by Woolf [2010], we created our categorization for ITS types based on the conduction of a systematic review of literature on existing ITS authoring tools. We explain these categories in Section B.5 pointing out which type of ITS we are targeting in this thesis.

2.2 Gamification

Before defining gamification, we need to conceptualize games and differentiate from them. There are lots of descriptions and conceptual expositions about games. Salen and Zimmerman [2004] define games as "systems in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome". Juul [2010] proposes that all games have six main features: rules, variable, quantifiable outcomes, value-laden outcomes, player effort, player investment, and negotiable consequences, with respect to real life effects. In summary, as stated by Seaborn and Fels [2015], games emerge from a variety of combinations of rules, structure, voluntariness, uncertain outcomes, conflict, representation and resolution criteria in different proportions, and whether an experience is a game of gameful is determined by participant perception.

Gamification takes the power of games, and applies it to a given context to solve a problem. An important aspect of gamification understanding of what game elements are adequate in each problem and situation. Werbach and Hunter [2012] describe game elements as smaller pieces used to define building blocks that form the integrated gameplay experience. According to the same authors, these game elements are included in the dynamics, mechanics and components categories, as described below and showed in Figure 2.2:

- Game dynamics – the "big picture" aspects of the gamified system that you have to consider and manage but which can never directly enter into the game. For instance, constraints, emotions, narrative, progression, relationships, and personalization.

- Game mechanics – the basic processes that drive the action forward and generate player engagement. For instance, challenges, chance, competition, cooperation, feedback, resource, acquisition, rewards, transactions, turns, win states, and profiles.
- Game components – the specific instantiations of mechanics and dynamics. For instance, achievements, badges, collections, leaderboards, levels, notifications, points, progress bars, quests or missions, status, teams, virtual goods, and so on.

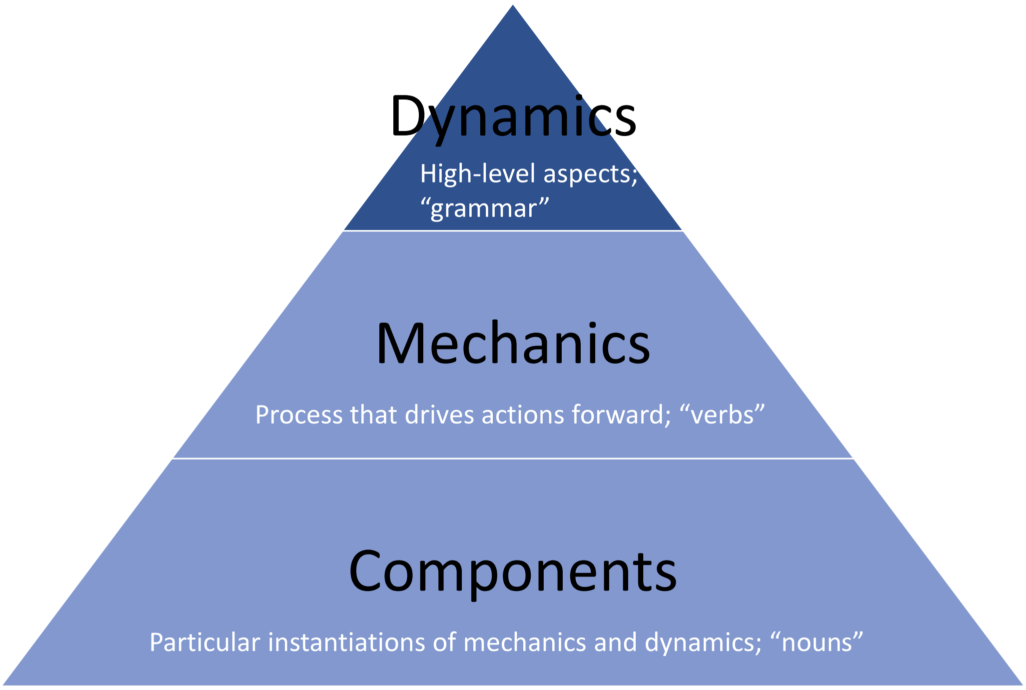


Figure 2.2: Adapted from Werbach and Hunter [2012]

While there is no standard conceptualization of gamification, most sources agree that gamification is generally defined as the use of game elements and mechanics in non-game contexts. Seaborn and Fels [2015] summarize the intersection of gamification conceptualizations provided by Deterding et al. [2011], Werbach and Hunter [2012], and Huotari and Hamari [2012] and presents an emerging standard definition of gamification: the intentional use of game elements for a gameful experience of non-game tasks and contexts. As games elements, the authors consider patterns, objects, principles, models, and methods directly inspired by games.

Furthermore, Hamari et al. [2014b] conceptualize gamification as a process which includes motivational affordances, psychological outcomes and behavioral outcomes (Figure 2.3). According to this conceptualization, gamification is defined as a process of enhancing services with (motivational) affordances in order to invoke gameful experiences (psychological outcomes) and further behavioral outcomes.

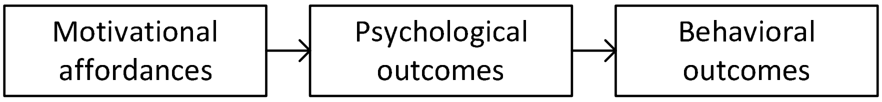


Figure 2.3: Outcomes of gamification. Retrieved from Hamari et al. [2014b]

With the aim of improving learning, several studies (e.g., the papers considered in the systematic literature review of Hamari et al. [2014b]) propose to use gamification in order to engage learners and to drive desired learning behaviors. For example, Li et al. [2012] investigated how story/theme, clear goals, feedback, challenge and rewards (motivational affordance by using game elements) could be used to increase the engagement and enjoyment (psychological outcomes) of students, and the results showed an increase in the speed of completion of tasks (behavioral outcomes).

In order to support the application of gamification to ITS, we rely on a gamification design framework proposed by Werbach and Hunter [2012]. In the following section we describe this framework, which is used to conceptualize gamified activities in the context of this thesis.

2.2.1 6D framework

As presented by Mora et al. [2015], among the gamification design frameworks, the best-known one is the 6D framework [Werbach and Hunter, 2012]. This framework is based on the Self-Determination Theory and is presented in six steps. It starts from a definition of business objectives and then proceeds to target the expected behaviours, describes the players, devises the activity loops without forgetting the fun, and finally, deploys the gamification system with the appropriate tools. In following we describe each one of these steps:

Define business objectives: This step includes defining the achievement of the project.

Werbach and Hunter [2012] establishes a process with three sub-steps in this activity: (1) make a list as concrete as possible and rank them; (2) eliminate the things that are not a final business objective; and (3) justify objectives;

Delineate target behavior: This step includes the behaviors that are intended for the users to reach. To define the target behaviors, some steps could be followed: (1) specify the tasks; (2) define the success metrics, the win states for every tasks; and (3) define the ways to measuring the win states [Werbach and Hunter, 2012]. An example of target behavior in the context of gamified ITS is increasing performance of students, as will be further presented in Chapter 5;

Describe your players: This step includes the description of the users (players) of a gamified system [Werbach and Hunter, 2012]. It can consider demographics, age groups, psychographics, kind of behavior, and so on. There are several player models that could be used in this step such as Bartle model [Bartle, 1996], Yee's player model [Yee, 2006] or BrainHex [Nacke et al., 2014].

Devise activity loops: This step includes identifying and evaluating the repetitives and recoursives structures, which focuses on two kinds of tasks: engagement loops and progressive loops [Werbach and Hunter, 2012]. The first kind of loops are based on the motivational design rules. Tasks that wanted to be repeated by users should be identified, motivated, and feedback should be given users. There are three elements in the engagement loop: (i) motivation, motivate the users to do something expected by the designers; (ii) action, where the user indeed do the task; and (iii) feedback, an immediate feedback given to the user to become them motivated and iterate the loop. By contrast, progressive loops are included on the design to drive users from a beginner to a master of a task. These loops consider activities from start to finish and a set of intermediate steps. They also provide small challenges to the user to arrive to a final goal. A gamification design must provide engagement and progressive loops as a natural way to help users to learn and to become a master in a gamified system.

Don't forget the fun: This step highlights the importance of considering fun in the design of gamified systems. As argued by Werbach and Hunter [2012], this is probably the most difficult part of framework because this issue is much more subjective than the others.

Deploy appropriate tools: This step include considering all the necessary tools to apply dynamic, mechanics, and components considering the particularities of players as well as the loops to drive users to achieve the business objects in a funny way.

As will be explained in Chapter 5, this gamification design framwork is used in the conceptualization of our ontological model in order to connect the gamification elements (i.e., player types, game design elements, etc) to the ITS concepts. Moreover, in the following section, we describe the BrainHex player type, which is used to describe players in the scope of the ontological model presented in thesis.

2.2.2 Brainhex player model

Several player models (also known as gamer types) have been proposed in the literature to describe different player types [Hamari and Tuunanen, 2014]. For instance, Bartle's model [Bartle, 1996], Yee's model [Yee, 2006], DGD1 [Bateman and Boon, 2005] and DGD2 [Bateman et al., 2011], BrainHex model [Nacke et al., 2014], and so on. In the context of e-learning, identifying students' player types would be of utmost importance to enable personalization of gamified activities based on particular characteristics of students.

Among the player models, the archetypes proposed in the BrainHex player model [Nacke et al., 2014] are based on neurobiological research, previous player models, discussions on patterns of playing, and literature on game emotions. It is the first model capable of identifying seven player types categories (Seeker, Survivor, Daredevil, Mastermind, Conqueror, Socializer, and Achiever) and classifying the players in classes and sub-classes related to each other, allowing a more accurate classification. As such, we use this player model in our ontological model for applying gamification to ITS. In the following, we describe each one of these player types and Figure 2.4 presents the BrainHex conceptual model.

- Conqueror: some players are not satisfied with winning easily – they want to struggle against adversity. Anger serves to motivate opposition and hence to encourage

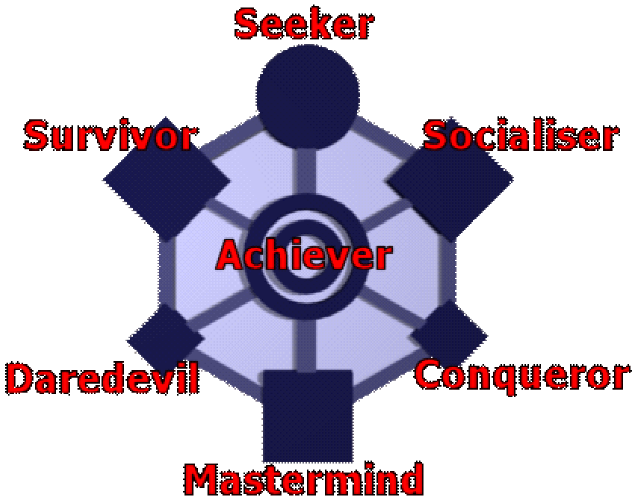


Figure 2.4: BrainHex conceptual model. Extracted from *http://blog.brainhex.com/*

persistence in the face of challenge, and testosterone may also have an important role in this behavior (irrespective of gender) [Nacke et al., 2014].

- Achiever: while a Conqueror can be seen as challenge-oriented, the Achiever archetype is more explicitly goal-oriented, motivated by long-term achievements. Achievers therefore prefer games amenable to ultimate completion, especially digital RPGs, whose self-adjusting difficulties ensure completion as a result of perseverance [Nacke et al., 2014].

- Daredevil: This play style is all about the thrill of the chase, the excitement of risk taking and generally playing on the edge. The behavior related to this type is focused around thrill seeking, excitement and risk taking, and thus epinephrine, which can be seen as a reward enhancer [Nacke et al., 2014].

- Mastermind: A fiendish puzzle that defies solution or a problem that requires strategy to overcome is the essence of fun to this archetype. Whenever players face puzzles or must devise strategies, the decision center of the brain and the close relationship between this and the pleasure center ensures that making good decisions is inherently rewarding [Nacke et al., 2014].

- Seeker: This archetypal is motivated by interest mechanism, which relates to the part of their brain processing sensory information (i.e., the sensory cortices) and the memory association area (i.e., hippocampus). The Seeker type is curious about the game world and enjoys moments of wonder [Nacke et al., 2014].

- Socialiser: People are a primary source of enjoyment for players fitting a Socialiser archetype – they like talking to them, they like helping them, they like hanging around with people they trust. The name of this archetype pays tribute to Bartle's Socialisers, verified by Yee's relationship motivation [Nacke et al., 2014].
- Survivor: While terror is a strong negative experience, certain people enjoy the intensity of the associated experience, at least within the context of fictional activities such as horror movies and games. The state of arousal associated with epinephrine becomes that of terror as a result of the action of the fear center, which becomes hyperactive when a situation is assessed as frightening (based on prior experience, and certain instinctive aversions). It is not yet clear whether the enjoyment of fear should be assessed in terms of the intensity of the experience of terror itself, or in terms of the relief felt afterwards [Nacke et al., 2014].

This player model along with its seven player types are considered in the conceptualization of out integrated ontological model (Chapter 5) in order to provide possible description of players that would use a gamified ITS customized by using the authoring solution presented in this thesis.

2.3 ITS authoring tools

Due to the large potential of intelligent tutoring systems to improve education by the use of technology, one of the main questions that could be asked, as discussed by Woolf [2010], is: why aren't thousands of effective educational resources available for teachers in various disciplines?. Moreover, another important question related to the broadened use of ITS is: where are the repositories of intelligent tutors? Woolf [2010]. In order to answer these questions, the high complexity to build those kinds of systems may be considered as well as the lack of tools to aid constructing those systems easily. As previously discussed, to build a new tutor, many stakeholders (e.g., developers, teachers, domain experts) should collaborate with each other and 200 hours of development for providing 1 hour of instruction is needed [Aleven et al., 2006].

In this way, the answer to these aforementioned questions includes noting that there few ITS authoring tools. Providing more authoring tools could support the rapid development of tutors, reducing the effort to produce them, increasing the number and diversity of available tutors, and favoring that more stakeholders be part of tutors instruction. In general, existing authoring tools provide a bag of tricks, rather than off-the-shelf tools [Murray, 2003, Sottilare et al., 2015, Woolf, 2010].

This section describes questions related to ITS authoring tools. First, these tools are classified according to the literature, and, then, we present design issues that might be considered when developing these types of systems.

2.3.1 Classification

Murray [2003] categorizes ITS authoring tools according to tasks performed in the tools and to the authored tutors. In summary, these systems are classified into two broad orientation categories: pedagogy-oriented and performance-oriented. Pedagogy-oriented authoring tools target on how sequencing and teaching educational resources that are relatively fixed. Most of these tools address pedagogical strategies and tactics representation. The performance-based authoring tools aim to provide richer learning environments in which students may learn skills by practicing these skills and receiving feedback. Table 2.2 lists seven categories of ITS authoring tools, relating them to the two orientation categories as well as describing their advantages and disadvantages. In the table, some possible variations of authoring tools per category are also pointed out.

It is worth noting that we can classify the authoring solution proposed in this thesis into the “Special Purpose”. As will be further described, our authoring solution intends to aid teachers in customizing gamified ITS features with high-level usability and simplicity.

2.3.2 Design issues

It is difficult to develop ITS authoring tools and design issues may confound their development process. The main objective to develop these systems is simplifying the

Table 2.2: ITS authoring tools categories. Extracted from Murray [2003]

Orientation	Category	Strengths	Limits	Variations
Pedagogy	Curriculum Sequencing and Planning	Rules, constraints, or strategies for sequencing courses, modules, presentations	Low fidelity from student's perspective; shallow skill representation	Whether sequencing rules are fixed or authorable; scaffolding of the authoring process
Pedagogy	Tutoring Strategies	Micro-level tutoring strategies; sophisticated set of instructional primitives; multiple tutoring strategies	(same as above for most systems)	Strategy representation method; source of instructional expertise
Performance	Device Simulation and Equipment Training	Authoring and tutoring matched to device component identification, operation, and troubleshooting	Limited instructional strategies; limited student modeling; mostly for procedural skills	Fidelity of the simulation; ease of authoring
Performance	Domain Expert System	Runnable (deeper) model of domain expertise; fine grained student diagnosis and modeling; buggy and novice rules included	Building the expert system is difficult; limited to procedural and problem solving expertise; limited instructional strategies	Cognitive vs. performance models of expertise
Pedagogy	Multiple Knowledge Types	Differential predefined knowl. representation and instructional methods for facts, concepts, and procedures, etc.	Limited to relatively simple fact, concepts, and procedures; predefined tutoring strategies	Inclusion of intelligent curriculum sequencing; types of knowledge/tasks supported
Performance	Special Purpose	Template-based systems provide strong authoring guidance; fixed design or pedagogical principles can be enforced	Each tool limited to a specific type of tutor; inflexibility of representation and pedagogy	Degree of flexibility
Pedagogy	Intelligent/Adaptive Hypermedia	WWW has accessibility & UI uniformity; adaptive selection and annotation of hyperlinks	Limited interactivity; limited student model bandwidth	Macro vs. micro level focus; degree of interactivity

construction process of ITSs. However, particular issues must be targeted regarding each tool under development. According to Murray [1999, 2003], Sottilare et al. [2015], Woolf [2010], development team of ITS authoring tools must consider some steps during the development of such tools such as: (i) identify the tutors; (ii) identify the authors; and (iii) identify target audience. In following, we describe each of these steps.

Identify the tutors to be produced: To construct ITS, the first decision to be made is if the authoring tool will produce specific tutors, designed for a explicit niche teaching, or if will produce generic tutors, which can be used at several domains. Specific ITS authoring tools generally produce a copy of an existing tutor [Murray, 1999, 2003]. These tools are good to create several types of tutors, although limited, they are powerful since they encapsulate the logics and reasoning needed for a tutor works and require less knowledge by authors. To author the development of this kind of tutor, these tools usually require simple inputs, thus non-programmers can use them as well as they can support extensions to particular tutors. However, they have a clear limitation that they only generate similar knowledge to the original tutor and result in the production of similar tutors. By contrast, ITS authoring tools that generate generic tutors may produce a large variety of tutors, even though they need more expertise about the student, tutoring or domain model, which might increase the learning curve to authors [Murray, 2003]. Using tools, authors may reason on which tutoring strategy to consider and the context on which it would be used. Generic ITS authoring tools require a large effort with respect to development tools and are so generic that can result in not so powerful and intelligent tutors.

Identify the authors: ITS authoring tools may be designed to a variety of authors (e.g., teachers, with limited skills in the use of computers) or software engineers who work in the construction of tutors [Murray, 1999, 2003]. The design process of using an authoring tool can be more or less scaffolded or automated based on questions about the author's skill level, available time for training, design and development, and knowledge of the target audience [Murray et al., 2003]. Limitations in the answer to these questions may imply in constraints about skills level and time needed to author tutors. Reducing the complexity of tools to enable unskilled authors to participate invariably reduces the capabilities of the resulting system [Murray, 1999].

Identify the students: It is of utmost importance to also identify, in the design of ITS authoring tools, the students who might learn in several different contexts (workplace, home, school, and so on), at several levels (middle-school, high-school, college, and so on), and whether students are, for example, workers or trainees [Woolf, 2010].

In this thesis, we consider all these design issues to propose our authoring solution, as will be further explained in Chapter 6.

2.4 Feature modeling and software product line

As previously explained, in this thesis we conceptualize the design space of ITS by conducting a feature modeling activity. This activity is one of the key steps to develop software product lines, thus, in order to contextualize the use of this activity in our work, we generally describe some software product line concepts.

The most important aspect provided by SPLs is the systematic reuse of all artifacts in the software development process. This systematic reuse is supported by two fundamental principles: reusable platform and customization [Pohl et al., 2005]. A reusable platform involves the identification of all common features of a family of products and the specification of these commonalities in all assets of the SPL. We mean by assets all artifacts that constitute a software development: requirements, software architecture, code, tests and so forth [Clements and Northrop, 2001]. In order to provide the customization of the products in a software product line, the notion of variability is also explored in all artifacts that are developed.

Software Product Line Engineering [Pohl et al., 2005] defines a process that specifies a set of activities of software development that supports the systematic creation of software artifacts aiming to manage the commonalities and variability of an SPL. There are two specific sub-processes to each one of the essential aspects of an SPL: (i) Domain Engineering - responsible for establishing a reusable platform and defining common and variable aspects of a software product line for a given domain, it consists of all types of software artifacts; (ii) Application Engineering - responsible for deriving the product line application from the platform established in the domain engineering. It exploits the variability of the product line and ensures that variability is consistent with specific needs of an application.

2.4.1 Feature Modeling

The variability of SPLs is commonly expressed through features represented in feature models. A feature is a property of the system that is relevant to some stakeholder and is used to capture similarities and variabilities of software systems. Feature modeling has been proposed as an approach for describing variable requirements for software product lines [Czarnecki et al., 2006]. It is an important activity of the software product line development process, since it is in such phase that the common and variable features of the product family are specified.

Features are organized in feature models according to one of the following types:

- Mandatory – the features in this category must be present in all products derived from a software product line;
- Optional – a feature of this type may or may not be included in a product derived from an SPL, hence its presence is optional;
- Alternative – in the alternative feature, exactly one feature from a set of features must be included in a product;
- Or-feature – one or more features from a set of features can be included in a product from an SPL.

The most widely used technique for modeling features was originally presented by Kang et al. [1990], named Feature-Oriented Domain Analysis (FODA). FODA provides a graphical tree-like notation that shows the hierarchical organization of features. The root of the tree represents the whole SPL node and all other nodes represent different types of features that are part of an SPL.

Figure 2.5 presents an example of a smartphone SPL feature model represented in the FODA notation. This feature model was adapted from a repository[1] of feature models and is used to illustrate this notation.

[1] Available at `http://www.splot-research.org/`

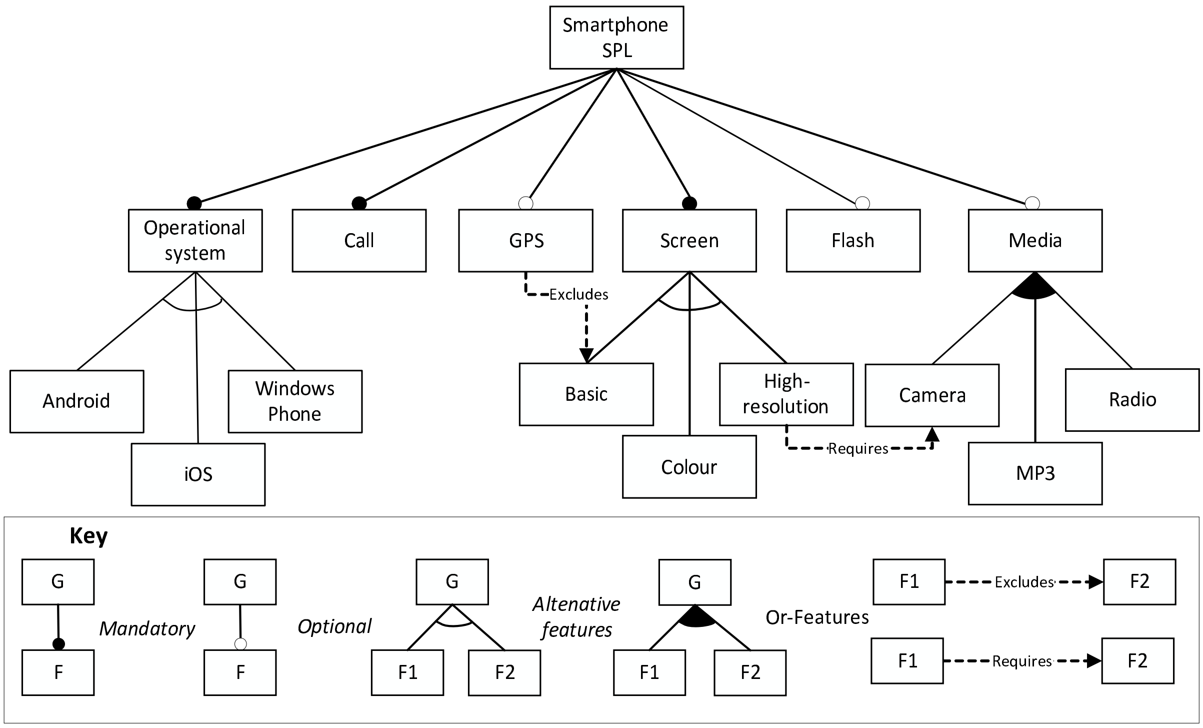


Figure 2.5: Smartphone SPL features model in the FODA notation

Figure 2.5 shows the graphical notation of each type of feature (mandatory, optional, alternative and or-features). Mandatory features are graphically represented by a small, filled black circle above the feature name (e.g., Operational system, Call, and Screen). Optional features are graphically specified by an open, non-filled white circle (e.g., GPS, Flash, and Media). Alternative features share the same parent's feature and are graphically represented by an open arc situated just below the parent's feature (e.g., Android, iOS, and Windows Phone). Finally, the or-features (e.g., Camera, MP3, and Radio) are represented by a filled arc, similar to the alternative features.

Additionally, in the feature modeling using FODA notation, it is possible to represent dependency rules between features, which can be one of two types: (i) Requires, when one feature requires the existence of another feature (they are interdependent), and (ii) Excludes, when one feature is mutually exclusive to another one (they can not coexist).

One of the contributions of this thesis is identifying and representing in the FODA notation, a feature modeling for gamified ITS. This model is further described in Chapter 4.

2.5 Ontologies

Ontologies are explored in this thesis in two different ways. First, it is used to formally conceptualize the gamified ITS feature model developed in order to enable softwares to reason on such model as well as to allow interoperability with third-party gamified tutors. Moreover, ontologies are also used to represent the knowledge about gamification theories and design practices, besides connecting such knowledge with ITS concepts. The representation of such knowledge intends to support our authoring solution in order to constrain the design space for customizing gamified ITS features aiming to facilitate teachers in the authoring process. As such, in this section, we first generally describe ontologies concepts, and then we describe a methodology that we used to construct our gamified tutoring ontology.

The term ontology comes from a branch of philosophy that deals with the nature of being. The term was introduced in computer science by artificial intelligence researchers who constructed computer models with some kind of automated reasoning. From the 90's, ontologies began to be treated as an integral part of knowledge-based systems, defined as an explicit specification of conceptualization [Gruber, 1993].

In the computer and information science context, an ontology defines a set of representational primitives in a particular knowledge area [Mika and Akkermans, 2004]. The usually adopted representational primitives are classes, attributes and relationships, including their meanings and restrictions. Ontologies are typically specified with languages that allow some kind of abstraction from data structures and from implementation strategies [Gruber, 1995].

Ontology languages are used for domain formalization by defining classes and properties for these classes, individuals (that instantiate the classes), properties of individuals, and statements on these individuals. It also allows to reason about these classes and individuals according to formal semantics defined by the language, which may support the automated reasoning and inference on such models.

Ontologies can be written down in a wide variety of languages and notations, such as Description logics [Baader, 2003], First-order logics, Relational-model, UML and so on. However, ontologies are generally represented on the web using one of the variants of the

Web Ontology Language (OWL) [McGuinness and Harmelen, 2004], which is part of the technologies stack defined by the World Wide Web Consortium (W3C) for Semantic Web.

The OWL language has its roots in Description Logics and provides formal and clear semantics for the definition of concepts and their relationships. OWL ontologies are often serialized using an RDF/XML representation – also part of the stack of W3C technologies – which is a triple format that models information using triples in the form of subject-predicate-object expressions. The information represented in RDF format (e.g., OWL ontologies) can be queried using a standard RDF query language called SPARQL [Pérez et al., 2009], which is an SQL-like language.

In the following sections, we present a classification of ontologies in order to situate the ontologies presented in this thesis. We also describe how ontologies could be used in the context of software development in order to explain the role of ontologies in the feature modeling activity of gamified ITS. Then, we describe an ontology engineering methodology that was used to develop the ontologies that represent gamification and ITS concepts

2.5.1 Types of ontologies

Researchers have observed the use of ontologies under different viewpoints, thus, we can find in the literature several ontology classifications with different emphasis, for example, by the level of generality, type of conceptualization structure, nature of real world issues, and so on [Calero et al., 2006]. Hereafter, we present a classification by the level of generality (Figure 2.6), proposed by Guarino [1998], since it is sufficient to situate the use of ontologies in the context of this thesis:

- Top ontologies – describe general concepts such as space, time, matter, object, event, action, etc; these concepts are independent from a specific problem or domain. Thus, it might be reasonable (at least in theory) to have unified top ontologies to larger user communities;

- Domain and task ontologies – describe, respectively, the vocabulary related to a generic domain (e.g., medicine, or automobile) or to a generic activity or task (e.g., diagnose or sell), by specializing the terms introduced in the top ontologies;

- Application ontologies – describe concepts that depend both from a particular domain or from a task, they are usually specialized from both related ontologies. These concepts normally correspond to the roles that are played domain entities while they perform a certain activity.

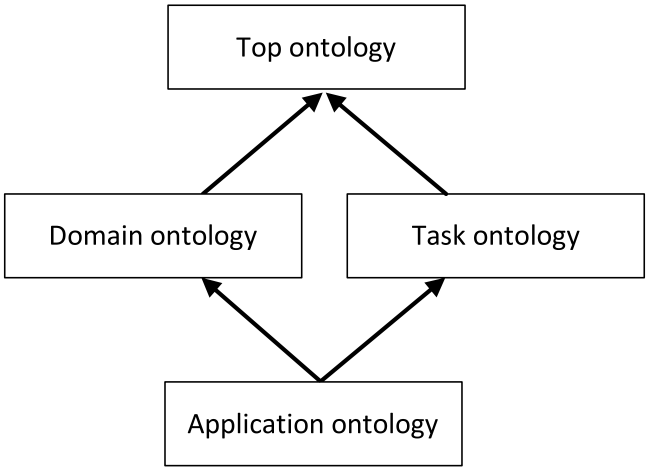


Figure 2.6: Ontology types according to the level of dependency of particular task or viewpoint. Arrows represent the specializations relations. Adapted from Guarino [1998]

In this thesis we propose ontologies that are classified into different types. The ontology for representing feature models that we present in Chapter 4 may be classified as an application ontology. Moreover, with respect to the ontologies presented in Chapter 5, the ontology that represents gamification concepts may be classified as a domain ontology, whereas the gamified tutoring ontology could be categorized as an application ontology.

2.5.2 Ontologies and software engineering

One of the contributions of this thesis is using ontologies to represent the knowledge about gamified ITS features variability in a way that it could be automatically analyzed by machine. In this way, we are also interested in using ontologies during the software development process, hence, this section presents concepts related to symbiosis between ontologies and software engineering.

Software is part of a technical category and is designed to perform particular tasks using computers, but can also be considered social since nowadays it is used in every aspect of people's life. Indeed, software may be seen as a knowledge repository where the knowledge is to a large extent related to an application domain [Armour, 2006]. Thus, it is important to allow sharing and interoperability of the knowledge contained in softwares, including the knowledge about all relevant aspects that surround and influence software (e.g., domain knowledge, new requirements, political and contextual issues on which people use and interact with) to leverage software to a more advanced level. Sharing and managing software demands the explicit use of the knowledge definition since it is a basic need to machine become able to interpret knowledge. This is the main reason for the software engineering community acknowledge the use of ontologies as a potential way to target several recurrent software engineering problems [Calero et al., 2006, Gašević et al., 2009, Happel and Seedorf, 2006, Isotani et al., 2015, Pan et al., 2012].

The use of ontologies in software engineering has gaining the attention of several researches recently. Many researchers have pointing out that the use of ontologies and other web-semantic related technologies have a large potential for impacting different activities of the software development such as developing models and languages more amenable to represent software; in the process of elicitation, analysis and specification of requirement, in the management of software development process, in the verification and validation of systems, in software maintenance, and so on [Calero et al., 2006, de Cesare et al., 2009, Dermeval et al., 2014, 2015b, Gašević et al., 2009, Pan et al., 2012].

Ontologies are used in the context of software engineering in several ways. According to a taxonomy proposed by Ruiz and Hilera [2006] (Figure 2.7), ontologies are typically used in software engineering as an alternative technique or artifact to be applied in the software development process. However, although less common, it is possible to use ontologies to represent the knowledge about the software engineering domain. Thus, in a basic level, the taxonomy is divided into two generic categories: domain ontologies and ontologies as software artifacts:

- **Domain ontologies** – this category refers to ontologies on which the main objective is to represent (at least partially) the knowledge about a certain sub-domain of the software engineering. Ruiz and Hilera [2006] argue that the domain ontologies classification must be based on norms, recommendations, and patterns published by prestigious organizations and associations (e.g., ACM and IEEE), been accepted and broadly known by the international community dedicated to such discipline. A

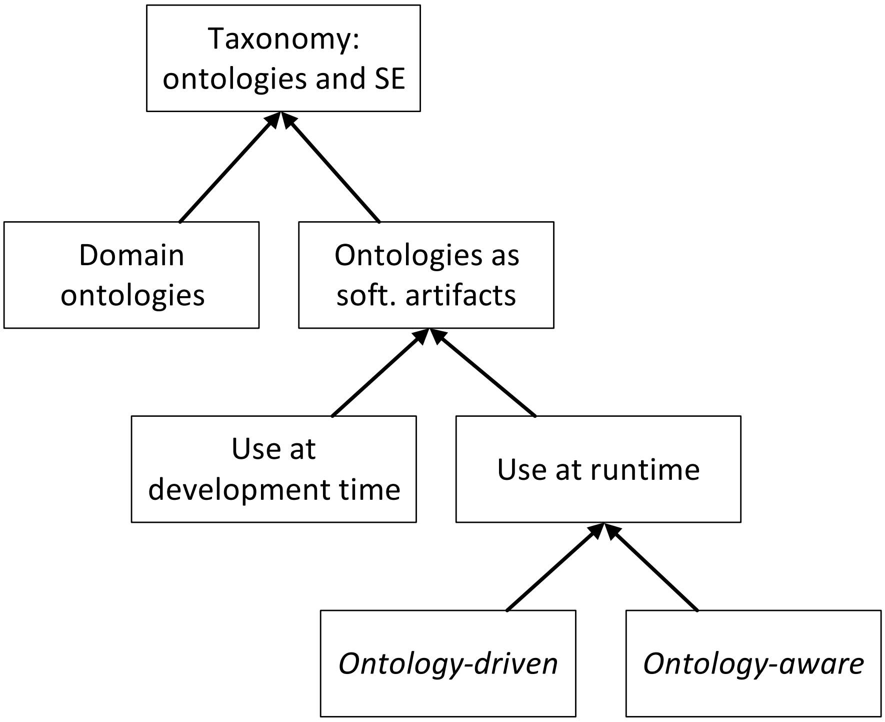


Figure 2.7: Taxonomy on the use of ontologies in software engineering. Based on Ruiz and Hilera [2006]

classification in this category includes generic ontologies that have the objective of modeling all software engineering body of knowledge, based on different sources of information such as glossaries (e.g., IEEE), guidelines (e.g., SWEBOK), and classic software engineering books. Other classifications (specific SE ontologies) intend to conceptualize sub-domains of SE such as requirements engineering, design, implementation, test, quality, and so on;

- **Ontologies as software artifacts** – there are many proposals that use ontologies as artifacts, with various characteristics and funtionalities, during the construction or functioning of software systems. According to Ruiz and Hilera [2006], once software artifact can be used both at development or runtime, a natural sub-classification of this category is separating them in these two types of using ontologies:
  - **Ontologies as software artifacts at development time**: this classification is given to the works that use ontologies to support some phase of the software development process, for example, in requirements engineering, architectural

design or implementation. As explained by Ruiz and Hilera [2006], most of the works included in this category apply a domain-oriented software development based on the use of knowledge about application domain to guide software developers through the software process steps, facilitating the understanding about the problem during development;

- **Ontologies as software artifacts at runtime**: this classification is given to the works that use ontologies at runtime in the context of software engineering. It can be sub-classified according to its use as architectural artifacts or informational resources, as described below:
  * **Ontologies as architectural artifacts**: this category can be also referred as ontology-driven software. In this classification, ontologies are part of the software architecture, as an additional component, cooperating with the other software components at runtime to help to achieve a task or objective. In the works included in this category, the software architecture is characterized by the use of one or more ontologies that are central elements of the proposed system. This knowledge-based system has an architecture that is mainly composed by a knowledge repository which includes an ontology and an inference engine that acts on such repository;
  * **Ontologies as resources (information)**: this category can be also referred as ontology-aware software. In this classification, ontologies are used by the software at runtime for a specific purpose (i.e., as an informational resource), normally remote, on which the software operates, running, for instance, specific queries. Within this category are those proposals which deal with software systems that use one or more ontologies at runtime in order to, for example, use their content in operations of information searching. In general, these applications use ontologies as database substitutes, for information storage.

In this thesis, we use ontologies in the context of software engineering to represent the knowledge about the gamified ITS features to be use as architectural artifacts (ontology-driven feature modeling[2]) to support the variability management of our authoring

[2]In this thesis, we often refer to ontology-based feature modeling instead of ontology-driven feature

solution. We may also consider the use of the integrated ontological model (GaTO ontology) as a resource of our authoring solution since the software must manage this ontology at runtime performing several operations on it.

2.5.3 METHONTOLOGY

METHONTOLOGY is a methodology that describes a set of phases and techniques to build an ontology either from scratch or by reusing other ontologies. The ontology development process by using this methodology identifies the required tasks when working on an ontology, i.e., planification, specification, knowledge elicitation, conceptualization, formalization, integration, implementation, evaluation, documentation, and maintenance. With the ontology life-cycle, these tasks acquire order and depth through the ontology lifetime. Therefore, the methodology framework was built based on these concepts,, specifying the used techniques, determining which products are obtained, and deciding how to evaluate each activity.

As shown in Figure 2.8, the METHONTOLOGY framework is structured in seven phases, Specification, Knowledge Acquisition, Conceptualization, Integration, Implementation, Evaluation, and Documentation. The specification phase aims to produce an ontology specification document written in natural language and contains information like the purpose, level of formality, and scope of the ontology. The knowledge acquisition (activity represented in the bottom of the figure) is independent and is worked simultaneously within the whole ontology development process; but, it is more intense in the specification phase. The conceptualization activity builds a conceptual model using terms of the domain vocabulary that were acquired in the specification activity. In addition, it uses that model to give users an overview of the domain's problems and their solutions. The integration phase searches for existing meta-ontologies that may help to speed the construction of the developing ontology by reusing its definitions instead of creating them from scratch. With the implementation activity, the ontology should be codified in a formal language, e.g., OWL. The evaluation phase gives a technical judgment of the ontology, verifying and validating it; so, this activity is worked during each phase and between phases of the life cycle. Finally, the

modeling. However, the meaning of both terms are semantically equivalent in this thesis: feature model used as an architectural artefact of our authoring solution.

documentation in METHONTOLOGY is performed during the whole ontology development process, producing a natural language document for each phase of the framework.

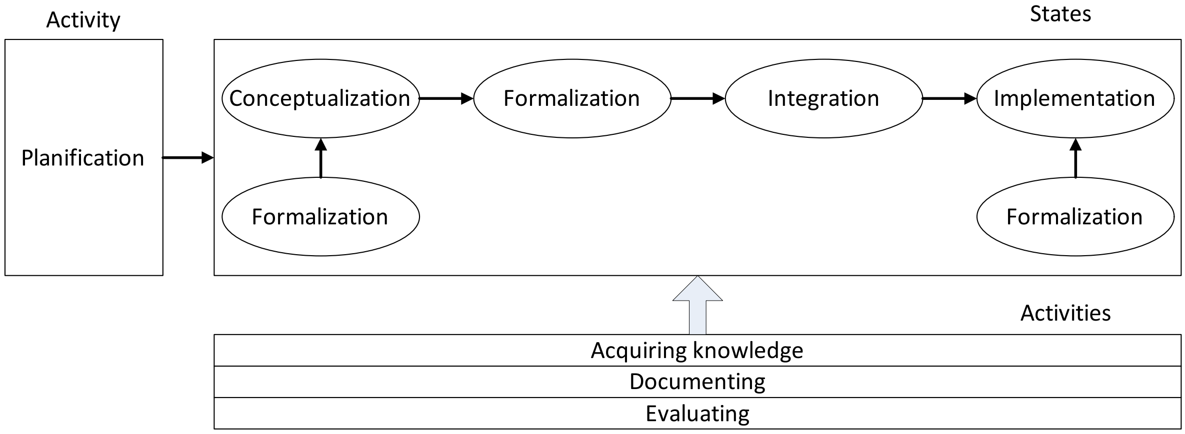


Figure 2.8: METHONTOLOGY states and activities. Adapted from Fernández-López et al. [1997]

As will be further presented in Chapter 5, we use this methodology to develop our integrated ontological model that is used to represent and connect gamification and ITS concepts.

2.6 Concluding remarks

In this chapter we described the main theoretical background that we rely to propose the contributions presented in this thesis. We depict the main concepts, definitions, and technologies used in this thesis, such as intelligent tutoring systems and their types, gamification (including the 6D framework and the BrainHex player model), ITS authoring tools along with a classification and some design issues to consider, feature modeling, and ontologies (including their types, their relation with software engineering and the ontological engineering methodology used to conceptualize our integrated ontological model).

In the next chapter, we investigate the literature on these topics to identify the related works to our contributions.

Chapter 3

State of the art analysis

In this chapter we present the results of the state of the art analysis conducted in this work. We consider the literature regarding the works that are somewhat targeting our research questions using similar concepts, theories, and technologies we are using in this thesis. As such, for each of our proposals, we describe the methodology used to conduct the literature review and, then, we present the main related works identified, comparing our work to them. Section 3.1 presents the analysis of the state of the art related to the use of feature modeling and software product line in ITS. In Section 3.2, we analyze the literature about ontology-based feature modeling. Section 3.3 describes works that use gamification along with ITS. Finally, in Section 3.4, we describe related works with respect to the use of ITS authoring tools for non-programmer authors.

3.1 ITS and feature modeling/software product line

In the following sections, we describe the methodology we have used to identify our related works on the use of SPL to develop ITS. We also present and discuss these works in the end of this section.

3.1.1 Review of the literature

To identify the related works with respect to the use of software product lines to develop intelligent tutoring systems, we conducted a systematic literature review of the literature.

The protocol use in the conduction of this systematic literature review is described in the Appendix A. Hereafter, we describe and discuss the works retrieved in the SLR in comparison to this thesis.

3.1.2 Related works

The results of the systematic review presented in the previous section suggest that only one paper target the construction of intelligent tutoring systems – which has a high variability – by using software product lines. The work by Silva et al. [2011] (S14 in our review) uses ontologies in the context of ITS to provide a semantic and consistent description of ITS knowledge. It describes a model to develop intelligent tutoring systems based on the use of software product lines and ontology.

However, even though presenting a platform for constructing ITSs that use SPL and ontology concepts, Silva's work do not consider in their solution the motivational perspective that we take into account in this thesis. Besides also using ontology along with SPL to develop ITS, it does not consider the extra variability produced by gamification when used together with ITS in their design. As previously mentioned, the motivational perspective of ITS (i.e., by using gamification) is of great importance to engage students and to drive desired learning behaviors.

Table 3.1 summarizes the comparison between the ontology-based feature model for ITS proposed in this thesis (as further explained in Chapter 4) against the only related work we have found during the conduction of the systematic literature review. Note that we include in the comparison only the work that applies SPL-based strategies to develop ITS.

Table 3.1: Comparison of our proposal to related works with respect to the research questions of the SLR

Work	**RQ1: Apply feature model or SPL to ITS**	**RQ2:Ontology**	**RQ3: Gamification**
Silva et al. [2011]	Yes	Yes	No
ITS Feature model proposed	Yes	Yes	Yes

As we propose our own strategy for using ontologies to represent feature models, in the following section, we describe how we investigated the literature regarding ontology-based feature modeling approaches, comparing the identified works to ours.

3.2 Ontology-based feature modeling

One of our contributions is an ontology-based feature modeling conceptualization used to formalize the gamified ITS feature model we define in this work. To investigate the literature about ontology-based feature modeling approaches, we describe in the following sections the methodology used to identify the related works to such contribution. In the end of this section, we also present and discuss these works.

3.2.1 Review of the literature

The literature review about ontology-based feature modeling started in the context of a broader systematic review of the literature about the use of ontologies in requirements engineering. Feature modeling is an activity commonly conducted in SPL engineering and mainly occurs at the requirements engineering phase. The preliminary results of this systematic review are published in Dermeval et al. [2014] and more complete and detailed results are published in Dermeval et al. [2015b]. We have followed a similar protocol to the review conducted in the previous section, based on Kitchenham and Charters [2007] guidelines. For the sake of clarity, we do not present all the details of the conduction of this review since the details are published in Dermeval et al. [2015b].

Table 3.2 presents the research questions investigated in the systematic review published in Dermeval et al. [2015b]. Particularly, the research question 2 identifies the requirements modeling styles, including feature models, used along with ontologies.

The results found in the conduction of this SLR identified three works that use ontologies to formalize the feature modeling activity[1] : Bagheri et al. [2011], Guo et al. [2012], Wang et al. [2007].

From the works identified in the systematic review, we performed a "snow-balling" search – technique in which the references of works are analyzed to identify other related

[1] Although the work by Ghaisas and Ajmeri [2013] uses ontologies and feature models, it uses ontologies to represent the requirements engineering domain considering of these models, i.e., they do not use ontologies do aid the feature modeling (ontology-driven). As a result, this work was excluded from the list of related works.

Table 3.2: Research questions and motivations. Extracted from Dermeval et al. [2015b]

Research Question	Description and Motivation
RQ1. What phases of the requirements engineering process have been supported by the use of ontologies?	This question provides a starting point to understand what are the main phases (elicitation, analysis, specification, validation and management) of the requirements engineering process supported by the use of ontologies.
RQ2. What styles (scenario-based, goal-oriented, feature model, etc) of software requirements modelling have been supported by the use of ontologies?	The answer to this question allows the identification of main styles of software requirements modelling (e.g., scenario-based, goal-oriented, textual requirements and so on) that have been supported by the use of ontologies. It may help to identify which requirements styles are attracting more attention to ontology community.
RQ3. What types (functional and/or non-functional) of requirements have been supported by the use of ontologies?	This question intends to identify what is the distribution of the studies with respect to the types of requirements (functional or/and non-functional) addressed. It is important to investigate if ontologies have been used to improve both functional and non-functional requirements.
RQ4. How are ontologies contributing to the solution of requirements engineering problems?	This question aims to describe contributions to solution of well-known RE problems. It is important because it provides a set of contributions regarding the use of ontologies to address some well-known RE research problems, which can be useful to researchers that might be interested in using ontologies in RE.
RQ4.1 What are the types of these contributions?	This sub-question intends to classify the contributions by its type, for instance, model, tool, process and method proposed in the study.
RQ5. Which ontology languages have been used in the ontology-driven requirements engineering methods?	This question identifies which are the main ontology languages (e.g., OWL, SPARQL, SWRL, UML and so on) been used to support requirements engineering methods. The answer to this question is also important because it can serve as a guide to researchers that might use some specific ontology language in RE.
RQ6. Which studies have reused requirements engineering ontologies?	The answer to this question indicates the existing RE ontologies and also presents how they are been reused in the studies included in the review. Thus, it is important because it identifies a set of RE ontologies which may be reused by researches on the use of ontologies in RE.
RQ7. Are there evidences of benefits of the use of ontologies in the RE Process?	This question intends to analyse if such studies provide some evidence that the use of ontologies benefits the requirements engineering process. These evidences should consider positive and negative results including empirical and non-empirical evaluation. They are important since they form a knowledge base about the use of ontologies in RE.

works. With this extra step, we identified other seven works that use ontologies to aid feature modeling: Asadi et al. [2012], Bošković et al. [2010], Filho et al. [2012], Kaviani et al. [2008], Lee et al. [2007], Noorian et al. [2011], Zaid et al. [2009].

3.2.2 Related works

In this section, we describe each one of these ten related works, comparing them to our ontology for representing feature models.

Wang et al. [2007] presents a technique to design ontology-based feature models, in which the feature model is represented using OWL classes and properties and reasoning mechanisms are used to automatically check configuration inconsistencies of the feature model. Lee et al. [2007] use ontologies to represent feature models and to analyze their variabilities and commonalities with the aim of analyzing the semantic similarity of feature models. To connect software procut lines and service-oriented architecture through the use of semantic web technologies, the work by Bagheri et al. [2011] propose an approach that semantically annotates feature models with the use of ontologies. Noorian et al. [2011] use description logics to identify inconsistencies in feature models and in configured products from a software product line, besides proposing possible corrections to it. However, all these works use an ontology modeling style based on OWL classes (which we characterize as receiving a medium flexibility with respect to be amenable to change). In addition, none of them use some kind of mechanism that favors the automatic analysis at runtime of the feature models.

Guo et al. [2012] present an approach to deal with inconsistencies in FM evolution scenarios. They formalize such models from an ontological perspective and define constraints that must be satisfied in FMs to be consistent. The work by Asadi et al. [2012] investigates the use of ontological theories (e.g., Bunge's ontology) to theoretically analyze variability languages. Although these works rely on ontological concepts to deal with feature model evolution, they do not provide neither any OWL implementation for representing feature models nor choose some ontology modeling style, hence, they have a low flexibility for changing. Besides, they do not propose any mechanism to deal with automatic analysis of feature models at runtime.

Filho et al. [2012] and Zaid et al. [2009] present ontologies for modeling feature models based on OWL individuals. The first work proposes an approach to automatically verify the consistency of feature models based on ontologies using OWL individuals. The second work presents an approach to enrich SPL using ontologies with the aim of providing information retrieval, inference and traceability properties to SPL life-cycle. However, although presenting a high level of flexibility, both works do not specify any mechanism to allow automatic reconfiguration based on the ontologies.

Kaviani et al. [2008] propose to use ontology to annotate feature models covering non-functional requirements modeling in the context of ubiquitous environments. Once a feature model is fully annotated in an ontology, analysis and reasoning are enabled in OWL. To achieve this purpose, the initial feature model is represented using the OWL language. This work uses the the ontology-based feature model approach proposed by Wang et al. [2007]. In a similar way, the work of Bošković et al. [2010] complements the approach of Wang et al. [2007] with an automatic configuration step by step. As such, a product configuration is realized as set of steps and the authors provide an algorithm to automatic specialize feature models based on description logics reasoning. However, although these works support feature model reconfiguration at runtime, they do not present a high level of flexibility.

As previously mentioned, the aim of using ontologies to aid feature modeling in our thesis is to represent the knowledge about the common and variable features of gamified intelligent tutoring systems in way that it can be automatic analyzed at runtime – which requires a high level of flexibility. In this way, some ontological modeling aspects may significantly impact on the flexibility to modify ontologies [Dermeval et al., 2015a]. To compare the flexibility levels of the existing works on the topic, we define the following levels: low, medium, and high. The works that present some kind of ontological conceptualization, but with no ontology implementation, receive a low flexibility. The works that have some OWL implementation, but use a modeling style based on OWL classes, receive a medium flexibility – since it is expected that they require a greater effort to change the classes defined in the ontology implementations in comparison to a modeling style based on OWL individuals – which receive a high level of flexibility. Thus, we consider two main criteria to compare the works that use ontologies to aid feature modeling to our proposal: flexibility and automatic analysis at runtime. Table 3.3 summarizes the comparison of our work to the related works according to these criteria.

Table 3.3: Comparison of our ontology for conceptualizing feature model and related works

Works	Flexibility	Automatic analysis at runtime
Lee et al. [2007]	Medium	No
Wang et al. [2007]	Medium	No
Kaviani et al. [2008]	Medium	Yes
Zaid et al. [2009]	High	No
Bošković et al. [2010]	Medium	Yes
Bagheri et al. [2011]	Medium	No
Noorian et al. [2011]	Medium	No
Asadi et al. [2012]	Low	No
Filho et al. [2012]	High	No
Guo et al. [2012]	Low	No
Ontology proposed (OntoSPL)	High	Yes

3.3 Gamification and intelligent tutoring systems

In this work we propose an ontological model that connects gamification to ITS theories, besides representing design principles for the use of gamification in the education context reported by the literature. As such, similarly to the previous sections in this chapter, in following we describe the methodology used to identify the related works to these contributions. Afterwards, we discuss the related works in comparison to our proposal.

3.3.1 Review of the literature

To identify the related works with respect to the application of gamification in ITS, we conducted an analysis of the papers included in three systematic reviews on the use of gamification, i.e., the works of de Sousa Borges et al. [2014], Hamari et al. [2014b], Seaborn and Fels [2015]. However, we could not identify any work that was targeting the design of gamified ITS. Thus, we also looked for related works in other sources, such as Google Scholar. As such, we identified five works than can be considered related to one of the contributions presented in this thesis: Andrade et al. [2016], Challco et al. [2014], González et al. [2014], Heyvaert et al. [2015], Shi and Cristea [2016]. Hereafter, we describe and discuss these works in comparison to this thesis.

3.3.2 Related works

In this thesis we use ontologies to conceptualize the knowledge about gamification theories and design principles to aid the application of gamification in ITS in a way that it can be automatic analyzed. Thus, we consider four criteria to compare this work to the related works identified: (i) apply gamification to ITS; (ii) use of gamification theories; (iii) define or use gamification evidence-supported design practices; (iv) consider ITS theories to connect gamification; and (v) formally conceptualize knowledge about gamification theories and design practices (e.g., using ontologies) in connection with ITS. In following we discuss these related works considering these criteria and Table 3.4 summarizes the comparison of our work to the related works.

Table 3.4: Comparison of our ontology for conceptualizing feature model and related works

Works	Apply gami. in ITS	Gami. theories	Gami. design practices	Connect gami. and ITS theories	Formal knowledge conceptualization
González et al. [2014]	Yes	No	No	No	No
Andrade et al. [2016]	Yes	Yes	Yes	Partially	No
Shi and Cristea [2016]	Yes	No	No	No	No
Challco et al. [2014]	No	Yes	No	No	Partially
Heyvaert et al. [2015]	Yes	No	No	No	Partially
Ontological model proposed	Yes	Yes	Yes	Yes	Yes

González et al. [2014] propose a conceptual architecture for building ITS taking into account gamification elements. The gamification elements are integrated into several modules of the system, such as game aesthetic in the student model's module and game feedbacks in the visualization module. In their work, Andrade et al. [2016] identify some problems about the use of gamification in existing gamified environments of the literature (e.g., addiction, undesired competition, and off-task behavior). For addressing such problems, they propose a framework to support the personalization of gamification for intelligent tutoring systems. Shi and Cristea [2016] explores how to approach gamification in social adaptive e-learning based on the Self-Determination Theory. They propose motivational gamification strategies rooted in such theory, achieving a high perceived motivation amongst students.

The aforementioned works present interesting approaches for using gamification in connection with ITS, for example, Andrade et al. [2016] explores the negative impact of gamification in learning to propose a framework for personalizing gamification, whereas Shi and Cristea [2016] achieved good effects on students' motivation using their gamification strategies. However, these works do not formally represent neither the knowledge about gamification theories nor the knowledge about ITS theories as well as how they are connected. In our work, we take advantage of ontologies to represented such knowledge in order to promote a more efficient reasoning and interoperability to support the development of tools that could intelligently design gamified ITS relying both on human and machine intelligence. Andrade et al. [2016] partially explores the ITS theories to apply gamification since their proposal considers some ITS components (e.g., student and tutor model). However, they do not rely on any specific ITS theories.

Ontologies have been significantly used in the domain of e-learning systems. Al-Yahya et al. [2015] present a survey of key contributions related to the development of and usage of ontologies in the e-learning domain. Their results suggest that most of the studies included in the review are using ontologies for supporting learning personalization, i.e., the main feature of ITS. However, none of these works make use of ontologies in order to support the application of gamification in ITS.

Regarding the use of ontologies for supporting the application of gamification in e-learning systems, few works are addressing such topic. Challco et al. [2014] present an ontological structure concerned with computer-supported collaborative learning (CSCL) systems to support the personalization of game design elements in collaborative learning contexts. To demonstrate its use, they show the personalization of a gamified collaborative learning scenario through a case study. However, once they target CSCL system, they only conceptualize gamification theories rather than ITS. Moreover, Heyvaert et al. [2015] present a framework that allows adding gamification to a digital textbook using standard technologies (i.e., EPUB 3 and Linked Data vocabularies). As part of their framework, they created a gamification ontology, representing some gamification concepts. This ontology is related to ours GaDO-core ontology, however, their ontology is limited to few gamification concepts (e.g., challenges, rewards and points systems). In summary, although their contributions use ontologies for leveraging the use of gamification in the e-learning domain, they are partially targeting the use of ontologies in comparison to our proposal since none of them are using ontologies to support the application of gamification in the ITS context.

3.4 ITS authoring tools

One of the main contributions of our thesis is providing a theory-aware authoring solution that considers the knowledge about ITS and gamification, besides taking into account evidence-supported design practices for using gamification in education. We describe in the following section the methodology we have used to identify our related works. We also present and discuss these works in the end of this section.

3.4.1 Review of the literature

To identify related works to our authoring solution we conducted another systematic review of the literature to investigate the existing works that propose ITS authoring tools for non-programmers since we are proposing a solution for teachers. The protocol used in the conduction of this systematic literature review is described in the Appendix B. Hereafter, we compare our proposal to the related works identified after conducting the review.

3.4.2 Related works

In the review presented above, we identified thirty-three papers that are proposing different kinds of ITS authoring tools for non-programmer authors. Some of these works might be more or less related to our work since we are proposing an authoring solution to aid teachers in the design of ITS with gamification capabilities. Our authoring solution uses an ontology-based feature model strategy to deal with the high variability of gamified ITS and relies on an integrated ontological model that represents knowledge about ITS components as well as gamificatin concepts and design practices. Moreover, our authoring proposal provides for teachers features for reusing pre-configured tutor designs and domain models in order to make the authoring process simpler and more usable. As such, we use the following criteria to compare our work to the ITS authoring tools found in the review described in the previous section: (i) is the ITS authoring tool targeting gamification?; (ii) is the authoring tool dealing (formally or not) with the high variability of ITS?; (iii) is the ITS authoring tool taking advantage of formal representation of ITS components?; and (iv) is there reuse features in the ITS authoring tool?. The first criterion can be identified by analyzing the first research question investigated in our SLR (Section B.4). The second and third criteria may be identified through the analysis of the fourth research questions (Section B.7), whereas the fourth criterion is identified by analyzing the results of the third research question (Section B.6).

Table 3.5 summarizes the comparison between our authoring solution against all the thirty-three papers found in the conduction of the systematic review of the literature. The papers S03, S04, S08, S09, S15, S17, S20, S21, S23, S26, and S32 are somehow dealing with variability inherent to ITS. However, among these papers, none of them are using a strategic reuse approach such as feature model or software product line to manage such variability. Moreover, as mentioned in Section B.7, the papers S01, S16, S28, and S29 rely on ITS formal representation (i.e., using ontologies) to deal with the knowledge involved in ITS design. Most of these works are aiding teachers in defining the domain model of tutors as well as relying on the reasoning and inference capabilities provided by ontologies to effectively use the domain model during tutoring. Four papers (S01, S13, S30, and S31) are providing some kind of reuse feature in different aspect of ITS authoring such as to reuse domain and content and tutors design. As shown in the table, none of the thirty-three papers are targeting the authoring of gamified ITS.

Finally, one might note that the ITS authoring tools identified and compared in this section might require more or less technical skills for non-programmer authors. As such, although we do not compare these works using this criterion, many of these authoring tools require more advanced technical skills such as the CTAT-based authoring tools. By contrast, as will be further presented in Chapter6, our authoring solution is designed for teachers, thus it requires no advanced technical skills.

Table 3.5: Comparison of our authoring solution against related works

Works	Gamification	Managing ITS high variability	Rely on ITS formal repr.	Provide reuse features
S01 [Abbas et al., 2014]	No	No	Yes	Yes
S02 [Alepis and Virvou, 2014]	No	No	No	No
S03 [Aleven et al., 2009a]	No	Partially	No	No
S04 [Aleven et al., 2016]	No	Partially	No	No
S05 [Barrón-Estrada et al., 2011]	No	No	No	No
S06 [Barron-Estrada et al., 2010]	No	No	No	No
S07 [Blessing et al., 2015]	No	No	No	No
S08 [Blessing et al., 2009]	No	Partially	No	No
S09 [Brawner, 2015]	No	Partially	No	No
S10 [Chakraborty et al., 2010]	No	No	No	No
S11 [Chou et al., 2011]	No	No	No	No
S12 [Devasani et al., 2012]	No	No	No	No
S13 [Escudero and Fuentes, 2010]	No	No	No	Yes
S14 [Fox et al., 2011]	No	No	No	No
S15 [Gilbert et al., 2015]	No	Partially	No	No
S16 [Grubisic et al., 2009]	No	No	Yes	No
S17 [Guin and Lefevre, 2013]	No	Partially	No	No
S18 [Heffernan, 2014]	No	No	No	No
S19 [Lane et al., 2015]	No	No	No	No
S20 [MacLellan et al., 2014]	No	Partially	No	No
S21 [MacLellan et al., 2015]	No	Partially	No	No
S22 [Marcus et al., 2010]	No	No	No	No
S23 [Matsuda et al., 2015]	No	Partially	No	No
S24 [Mitrovic et al., 2009]	No	No	Yes	No
S25 [Olney and Cade, 2015]	No	No	No	No
S26 [Olsen et al., 2014]	No	Partially	No	No
S27 [Paquette et al., 2010]	No	No	No	No
S28 [Refanidis, 2011]	No	No	Yes	No
S29 [Suraweera et al., 2010]	No	No	Yes	No
S30 [Troussas et al., 2014]	No	No	No	Yes
S31 [Virvou and Troussas, 2011]	No	No	No	Yes
S32 [Wilches and Palacio, 2014]	No	Partially	No	No
S33 [Zatarian-Cabada et al., 2011]	No	No	No	No
Proposed authoring solution	Yes	Yes	Yes	Yes

3.5 Concluding remarks

In this chapter we describe how we analyzed the literature related to our contributions. We firstly investigated the existing works (i.e., by conducting a systematic review of literature) that use software product line and/or feature modeling to deal with the high variability presented in gamified ITS design. Next, we compared our ontology for representing feature models with several works found in the literature. We also looked for works that apply gamification in intelligent tutoring systems and, finally, we identified (i.e., by using another SLR) and compared several ITS authoring tools that could be related to our authoring solution.

In the next chapter, we present our first contribution in this thesis, which is proposing an ontology-based feature model conceptualization to specify the variability of gamified ITS, enabling it to be automatic analyzed by third-party tutors.

Chapter 4

Gamified ITS ontology-based feature model

In this chapter we present how we achieve the two first objectives of this thesis, as described in Section 1.4. This chapter is divided into four parts, the first one (Section 4.1) describes the reference feature model that we have specified for representing the variability of gamified ITS. The second part (Section 4.2) presents the ontology-based feature modeling approach that we developed to represent the specified feature model, providing an automatic way to reason on the feature model as well as describing how we have evaluated our ontology-based feature modeling approach in comparison to a well-known approach of literature. In Section 4.3 we describe how we use the ontology-based feature modeling approach to specify the gamified ITS variability model and particular configurations of systems based on such model. In Section 4.4 we conclude this chapter by summarizing our contributions in this chapter.

4.1 Gamified ITS feature modeling

In order to specify a generic variability model for gamified intelligent tutoring systems, we have specified a reference feature model. We have identified the common and variable features of these systems by: (i) analyzing the features presented in a software product line designed for classic intelligent tutoring systems [Silva et al., 2012, 2011]; (ii) analyzing the features of ITSs that use gamification in industrial settings (i.e., *MeuTutor* and *Duolingo*); (iii) interviews with *MeuTutor* development team to gather information about new requirements that would be considered in future versions of *Meu Tutor* systems aiming to foresee variation points and variant features; and (iv) analyzing gamification [de Sousa Borges et al., 2014, Hamari et al., 2014b, Kapp, 2012, Seaborn and Fels, 2015, Werbach and Hunter, 2012] and ITS (i.e., [Sleeman and Brown, 1982, Sottilare et al., 2015, Woolf, 2010]) features investigated in the literature. In this way, the diagram presented in Figure 4.1 illustrates the features that we identified from these sources in the FODA notation [Czarnecki et al., 2006].

As shown in Figure 4.1, *Register* (along with a *Student Model* feature), *Login*, *Strategy*, *Evaluation*, *Gamification* and *Domain Model* are mandatory features, i.e., they have to be included in all gamified ITS that are based on this variability model. These features are mandatory since they are supported by majority of ITSs presented in the literature [Woolf, 2010], except for gamification, which is mandatory because we are investigating the use of game elements in ITS context. Additionally, the *Course Management*, *Social* and *Report* features are optional, i.e., they can be included or not in a system based on this feature model.

The *Register* feature offers a registration to the system and has an or-feature group to register the following actors: *Teacher* and *Student*. Once a student is registered in the system, it is also mandatory to build a *Student Model* for him. This feature represents the student model component of ITS, as previously described in Chapter 2.1.

The *Pedagogical Strategy* feature includes pedagogical approaches that could be selected in the system. There are basically two mandatory behavior strategies: *Outer Loop*, which has the main responsibility of selecting the next task to give to learners; and *Inner Loop*, which is related to the steps students perform in a task, i.e., an user's action that is part of the completion of a task [Vanlehn, 2006]. The *Outer Loop* feature is an or-feature group that allows choosing two features: *Syllabus* and *Curriculum Sequencing*. The first feature enables teachers to manage the schedule and subjects on which students will learn using the tutor – it is the reason there is a *Requires* constraint between this feature and the *Teachers* feature. The second feature defines the curriculum sequencing on which students must follow. In the *Inner Loop* feature there are two alternatives (or-feature type): pedagogical strategy based on bayesian knowledge tracing (BKT) [Corbett and Anderson, 1994] and a problem-based learning strategy.

Moreover, as can seen in Figure 4.1, the *Gamification* feature is mandatory. However, it

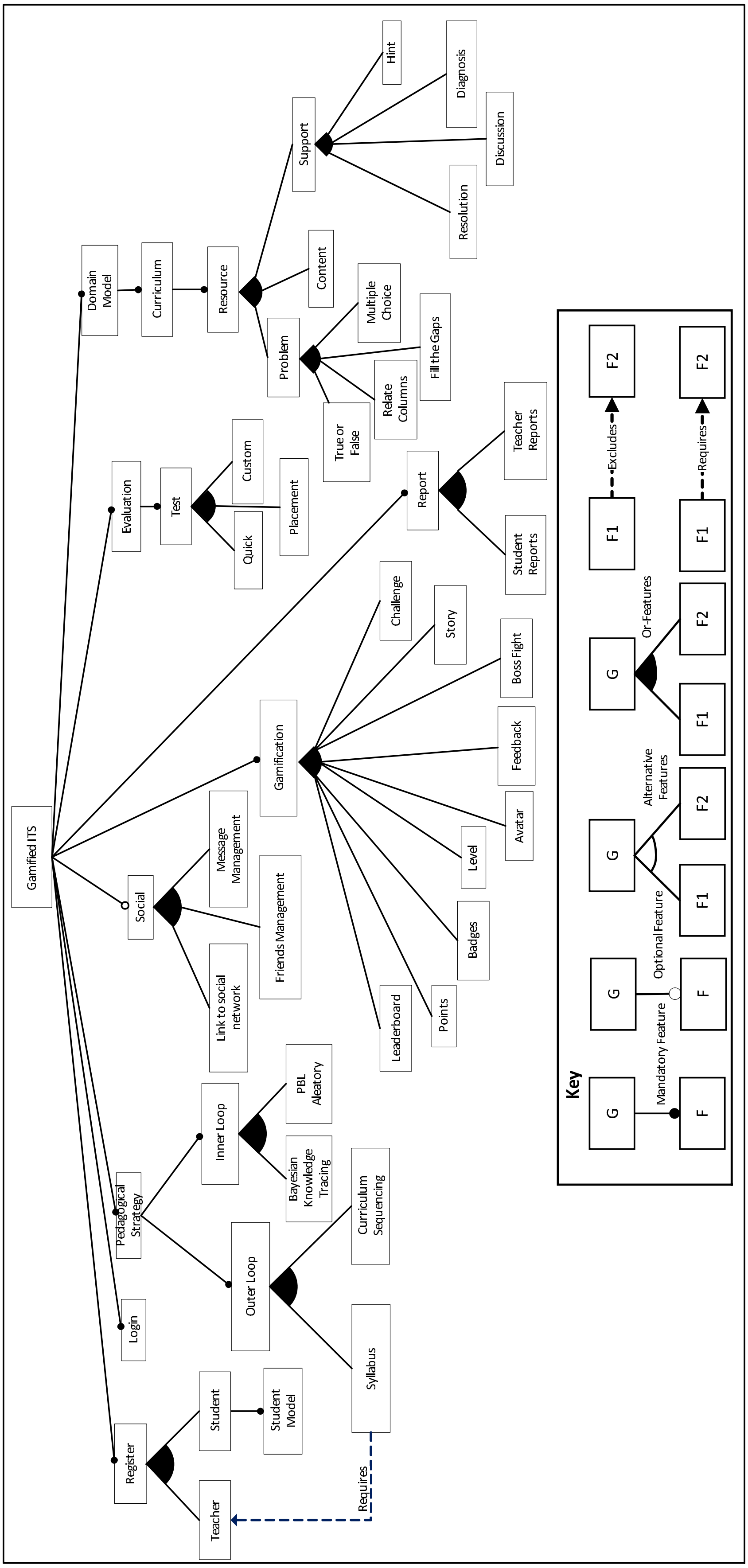


Figure 4.1: Gamified ITS feature model

has an or-feature group in order to represent that a particular system based on this variability model might select several combinations of game elements, i.e., *Leaderboard*, *Points*, *Badges*, *Level*, *Avatar*, *Feedback*, *Boss Fight*, *Story*, and *Challenge*. Using the *Leaderboard* feature, an user may compare his performance to other users. The *Points* feature represents the game element where a learner can earn points by using some types of resources, as explained in the previous section. The *Badges* feature represents the game element where an user may receive a badge by performing some action in the ITS, for instance, by solving a test. The *Level* feature represents the game element that sets a level to an user according to a certain condition (e.g., knowledge). The *Avatar* feature includes a virtual representation of a student's character in the system, this element is common in role-playing games in which the player might take on the role of a magical creature or a medieval warrior. The *Feedback* feature enables the system to give feedback with respect to interaction with game design elements in the system. The *Boss Fight* enables users to "fight" against a high-level opponent, called boss, i.e., this feature may often marks the end of a level or a section of a game. Finally, the *Story* feature activates a narrative description of a sequence of events using storytelling or theme. The *Challenge* feature may include mission or challenges for students using the tutor.

As previously mentioned, the *Evaluation* feature is mandatory. It includes a *Test* feature that may be used at different ways by students: *Quick*, *Customized*, and *Placement*. In the *Quick* test, a student requires a test to the system and receives a test that is automatically generated by the tutor every time a student wishes to test his/her knowledge. The *Custom* feature enables students to select particular subjects to have the knowledge evaluated. The *Placement* feature enables students to evaluate more advanced subjects than their current knowledge, if a student has success in this test she has her knowledge level updated to include learning about such subjects.

In addition, the *Domain Model* feature contains the curriculum (*Curriculum* feature) of a particular domain and a set of resources (*Resource* feature). A learner may use different types of resources, such as *Problem*, *Essay*, *Forum*, *Support* and *Content*. The *Problem* feature has an or-feature group representing the types of problems (*True or False*, *Relate Columns*, *Fill the Gaps* and *Multiple Choice* features) that could be selected in an arbitrary gamified ITS. The *Support* feature contains an or-feature group indicating the types of help that a user could ask in a particular product, i.e., *Resolution*, *Diagnosis* and *Hint* features.

The *Social* feature can be optionally included in an gamified ITS system based on this feature model. It has an or-feature group indicating that within this feature a product could have at least one of the *Links to Social Network*, *Friends Management* and *Message Management* features. The *Report* feature may be optionally included in ITS products. It has an or-feature group representing two types of reports that could be selected: *Teacher Reports* and *Student Reports*.

Once a reference feature model for gamified ITSs is defined, this variability model may be used to aid the definition of different configurations of these systems. For instance, the green features of the Figure 4.2 represent the selected features of a particular configuration of the *Meu Tutor* gamified ITS for the ENEM (Brazilian high-school national exam) domain.

4.2 OntoSPL: an ontology-based feature modeling approach

In this section, we present an ontology-based feature modeling approach that we have proposed [Dermeval et al., 2015a, Tenório et al., 2014], called OntoSPL[1]. As previously explained, our intention with this approach is formalizing feature models in a way that such models could be automatically reasoned by machine at runtime. Note that, although we are also contributing to software engineering research by presenting a new approach for representing feature models which can be used at different domains (e.g., ubiquitous computing, autonomic systems and context-aware computing), our ultimate goal is to use this approach to enable automatic analysis of gamified ITS features, which is represented by the feature model presented in the previous section. As a result, it would be possible to automatically reason on such model for managing the reconfiguration of gamified ITS according to preferences received as input from teachers using an authoring tool, as will be further explained in Chapter 6. The gamified ITS feature model represented in this approach might be also used to integrate our authoring solution to third-party gamified ITS platforms, such as MeuTutor.

[1] Available at `http://surveys.nees.com.br/ontologies/OntoSPL.owl`

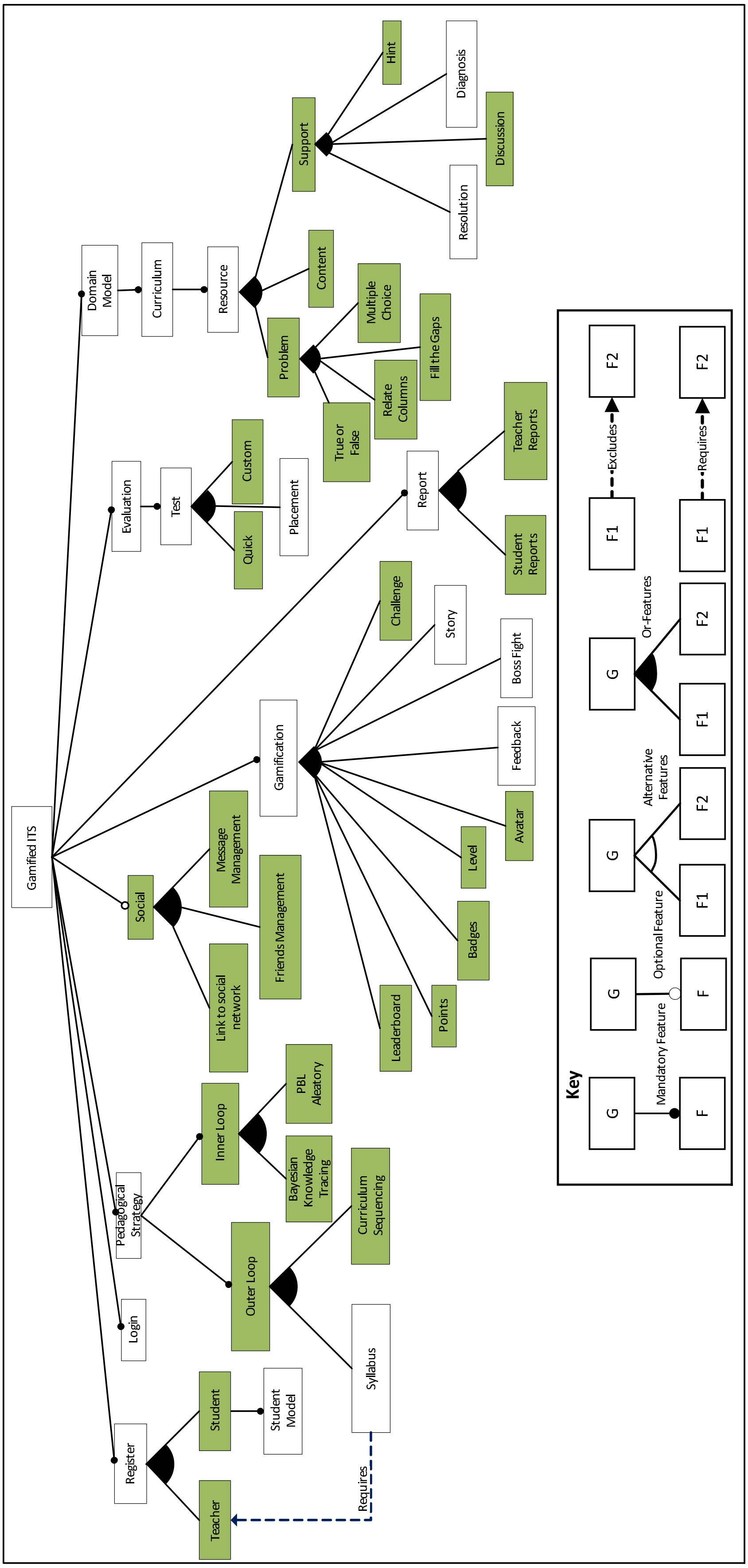


Figure 4.2: Configuration of the gamified ITS feature model for the MeuTutor ENEM

4.2.1 OntoSPL description

As described in Dermeval et al. [2015a], Tenório et al. [2014], OntoSPL describes the concepts of SPL based on a feature diagram. An SPL has a name, a description and contains a feature diagram. A feature diagram has a name, a set of root features and a set of feature constraints. As explained in Chapter 2, a feature is a resource available to the system. It has a name and can be classified as Mandatory, Optional or Alternative. Features are organized like a tree, hence it has a parent (when it is not the root) and may have some children. Moreover, an alternative feature has a set of alternative features with itself and an exclusive property. In addition, a feature constraint has a name and can be classified as Depend (Require), Exclude or Group. The Depend constraint has a name, a set of source features and a set of target features. It means that if all source features are selected in a product, all the target features must be selected too, in the same product derived from an SPL. The Exclude constraint has exactly the same properties of the Depend one. It has only a semantic difference, since if all source features are selected in a product then any target features may not be selected in such a product. Finally, the Group constraint has a name, a set of features and a constraint type that indicates a type of the constraint on the group. Figure 4.3 illustrates the hierarchy of classes of the OntoSPL ontology.

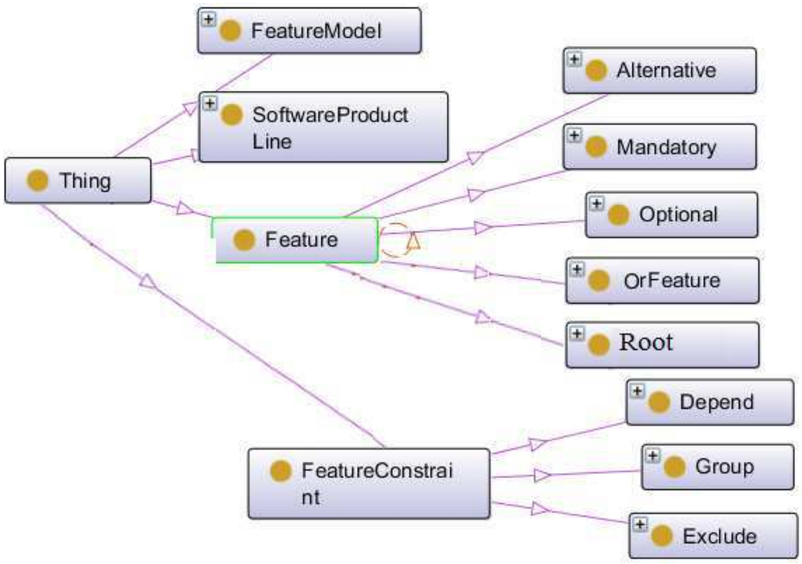


Figure 4.3: OntoSPL classes hierarchy. Extracted from Tenório et al. [2014]

Hereafter, the classes, properties and concepts of this ontology are presented:

- *SoftwareProductLine (name, description, FeatureModel)*: this class represents an arbitrary Software Product Line. It has primitive elements such as name and description. Moreover, a SPL contains a Feature Model;
- *FeatureModel (name, Feature, FeatureConstraint)*: this class describes a Feature Model that represents the hierarchical organization of the features of an SPL. It has a set of features and a set of feature constraints;
- *Feature (name)*: this class represents a resource available in the software product line. It may be classified into Mandatory, Optional or Alternative:
  - *Mandatory (name)*: this class represents a mandatory resource of the SPL, i.e., it must be present in all products;
  - *Optional (name)*: this class represents an optional resource of the SPL, i.e., it is optionally present in any product;
  - *Alternative (name, exclusive, AlternativeFeature)*: this class represents an alternative resource of the SPL. An alternative resource specifies that two or more resources may not co-exist.
- *FeatureConstraint (name)*: this class represents a constraint in the feature model. It may be classified into Depend, Exclude or Group:
  - *Depend (name, SourceFeature, TargetFeature)*: this class represents a constraint of the Depend type. As mentioned above, it has a set of source features and a set of target features;
  - *Exclude (name, SourceFeature, TargetFeature)*: this class represents a constraint of the Exclude type. As mentioned above, it has a set of source features and a set of target features;
  - *Group (name, SetFeatures, typeConstraint)*: this class represents a constraint of the Group type. It has a set of features and a typeConstraint that indicates the type of the constraint. It can be: (i) zero-or-one feature exactly (0 or 1), (ii) At-least-one feature (1 or more), (iii) Exactly-one feature (1), (iv) Any feature (0 or more), or (v) All features (n).

The following relationships[2] are represented in the ontology:

- *hasRootFeatures (FeatureModel, Feature)*: specifies that a FeatureModel contains a set of root features (which may not be empty);
- *hasSetOfAlternativeFeatures (Alternative, Alternative)*: specifies that an alternative feature must have at least one feature alternative. It is a symmetric property;
- *hasSetOfConstraints (FeatureModel, FeatureConstraint)*: specifies that a FeatureModel contains a set of feature constraints;
- *hasSetOfFeatures (Group, Feature)*: specifies that a Group constraint contains a set of features (which may not be empty);
- *requires (Feature, Feature)*: specifies a that a feature requires the selection of other feature;
- *isRequiredBy (Feature, Feature)*: specifies a that a feature is required by the selection of other feature. It is the inverse property of requires;
- *excludes (Feature, Feature)*: specifies a that a feature excludes the selection of other feature;
- *isExcludedBy (Feature, Feature)*: specifies a that a feature is excluded by the selection of other feature. It is the inverse property of excludes;
- *isBasedOn (SoftwareProductLine, FeatureModel)*: specifies that a SPL is based on exactly one FeatureModel. It is a functional property;
- *isChildOf (Feature, Feature)*: specifies that a feature is the child of exactly one another feature. It is a functional property and it is also the inverse property of isParentOf;
- *isParentOf (Feature, Feature)*: specifies that a feature contains a set of children features. It is the inverse property of isChildOf.

[2]Note that four new properties (*requires*, *isRequiredBy*, *excludes*, and *isExcludedBy*) that are presented in the original publication by Tenório et al. [2014] are presented in this thesis. This is an improvement of our previous conceptualization

The classes and relationships described above express a taxonomy of the OntoSPL ontology. In order to describe it in a detailed and formal way, it must be governed with axioms. All axioms of the OntoSPL are defined in description logics (DL) and the ontology is implemented in OWL.

OntoSPL supports the instantiation of products based on the SPL in order to facilitate the reconfiguration of the product when it is necessary [Tenório et al., 2014]. In this sense, the property $current\_state$ of the Feature class indicates whether the feature belongs or not to a particular product. This property presents the following range of values: $\{"eliminated" : string, "selected" : string\}$. Such a property can only receive the values: selected, case the feature must be in the product, or eliminated, case the feature must not be in the product. Hence, a software (e.g., authoring tool) can reason in the ontology to perform dynamic reconfiguration in an arbitrary product (e.g., gamified ITS). After defining the features that may be present in the product to be created, there is only necessary to set the property $current\_state$ for each feature instantiated in a product.

4.2.2 Empirical evaluation in changing scenarios

There are basically two modeling styles that could be used to represent feature models through the use of OWL [Dermeval et al., 2015a]. The first one is based on OWL classes and the second one is based on OWL instances/individuals. However, considering changing scenarios (i.e., situations on which an operation with some feature of the model is demanded), it would be important to compare these styles to select which one is more amenable to deal with changes, especially at runtime.

In this way, in order to evaluate OntoSPL in changing scenarios, we conducted a controlled experiment [Dermeval et al., 2015a] that compares OntoSPL (which is based on OWL instances/individuals) and the ontology proposed by Wang et al. [2007] (based on OWL classes and properties) in several changing scenarios[3] (i.e., in fact we consider fourteen operations such as adding/removing mandatory features, optional features, and so on).

Our empirical comparison takes into account metrics such as, time to perform a change, flexibility for changing (measured by the structural impact of a change in the ontology), and

[3]Note that we chose the approach proposed by Wang et al. [2007] because it is one of the first approaches that use ontology in feature modeling and it is also published in a high reputation venue on Semantic Web field.

correctness for performing a change. As presented in Dermeval et al. [2015a], the execution of this experiment included ten participants in academic settings. The data gathered from this experiment was analyzed using descriptive and inferential statistics.

Table 4.1 presents the summary of statistics of both ontologies with respect to the three response variables we investigated and Table 4.2 presents the results of the hypotheses tests application. As shown in Table 4.2, there is statistical significance for two metrics: time for changing and impact of a change. In this way, our results might indicate, with 95%, that: (i) the time for performing change on the ontology of Wang et al. [2007] is higher than on OntoSPL, (ii) the structural impact of changes on the ontology of Wang et al. [2007] is higher than on OntoSPL (which may suggest that OntoSPL is more flexible than the other one), and (iii) there is no statistical difference between the ontologies regarding the correctness of changes. For more details about this experiment, including experiment design and execution, threats to validity, and so on, please see the work published by Dermeval et al. [2015a].

Table 4.1: Summary of statistics of the metrics evaluated (O1 = Ontology by Wang et al. [2007], O2 = OntoSPL). Time is measured in milliseconds, impact is measured by the total number of ontology elements changed, and the correctness is a ratio between the number of correct steps performed from participants and the total number of correct steps. Adapted from Dermeval et al. [2015a]

Metric	Ontology	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Time	O1	8969	44250	89000	192700	241100	1437000
	O2	4001	39160	96900	147000	182100	782200
Impact	O1	37	47	49	49.32	52	62
	O2	23	23	23	23	23	23
Correctness	O1	0	0.6667	1	0.7882	1	1
	O2	0	0.7292	1	0.7938	1	1

The results of this experiment indicate that using OWL individuals is more flexible and demands less time for changing than the one based on OWL classes and properties. Based on these results, we moved forward to use this approach in the context of our gamified ITS feature model. Our intention is to rely on the flexibility capabilities of this approach to enable

Table 4.2: P-values after applying Wilcoxon tests (O1 = Ontology by Wang et al. [2007], O2 = OntoSPL). Adapted from Dermeval et al. [2015a]

Metric	Null Hypothesis	Alternative Hypothesis	p-value	Decision (95%)
Time (T)	*H1-0* : $\mu_T(O1) = \mu_T(O2)$	*H1-1*:$\mu_T(O1) \neq \mu_T(O2)$	0.0006058	Reject
Impact (I)	*H2-0* : $\mu_I(O1) = \mu_I(O2)$	*H2-1*:$\mu_I(O1) \neq \mu_I(O2)$	$2.2e^{-16}$	Reject
Correctness (C)	*H3-0* : $\mu_C(O1) = \mu_C(O2)$	*H3-1*:$\mu_C(O1) \neq \mu_C(O2)$	0.7366	Fail to reject

the automatic analysis of the variability of these systems.

4.3 Gamified ITS ontology-based feature modeling

After defining the reference feature model of gamified ITS in Section 4.1, in this section we specify the feature model using OntoSPL. Figure 4.4 presents an overview of the dependency relations between the OWL files that represent the gamified ITS feature model and OntoSPL. We also present how particular configurations of these systems are related to the OWL file that represents the gamified ITS feature model.

To use OntoSPL for specifying our gamified ITS feature model, we must import the OntoSPL.owl file in a new OWL file and create a set of OWL individuals to represent the feature model of gamified ITSs. In this way, as shown in Figure 4.4, the OWL file that represents the gamified ITS feature model (GITS-PL.owl)[4] imports the OntoSPL ontology (OntoSPL.owl)[5]. As such, the new file that represents the gamified ITS feature model (Figure 4.1) is updated with OWL individuals.

In an analogous way to relational-based database modeling, OntoSPL.owl may play a role similar to the scheme of a table in the relational model, whereas the OWL file that represents the gamified ITS feature model would be equivalent to the instances within the relational model, based on the scheme defined by OntoSPL.

Figure 4.5 illustrates the ontology which contains the gamified ITS feature model previously defined. As shown in the figure, all gamified ITS features are represented as OWL individuals in the ontology. We also exemplify the use of GITS-PL.owl by creating

[4]Available at `http://surveys.nees.com.br/ontologies/GITS-PL.owl`

[5]Available at `http://surveys.nees.com.br/ontologies/OntoSPL.owl`

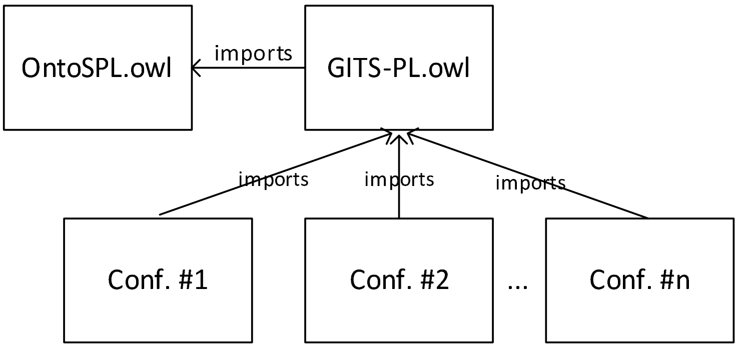


Figure 4.4: Relationship between the OWL files of OntoSPL, gamified ITS (GITS) feature model, and arbitrary configurations of gamified ITS

an OWL file[6] that represents the selected features for the MeuTutor-ENEM, as previously explained. For the MeuTutor-Enem, all green features highlighted in the Figure 4.2 receive a "selected" value with respect to their *currentState* dataproperties in the ontology, whereas, for the other features, they receive a "Eliminated" value in their dataproperty. Listing 4.1 presents an excerpt of the the MeuTutor-ENEM OWL File illustrating the selection of the *Badges* feature and the elimination of the *Story* feature for the MeuTutor-ENEM gamified ITS configuration.

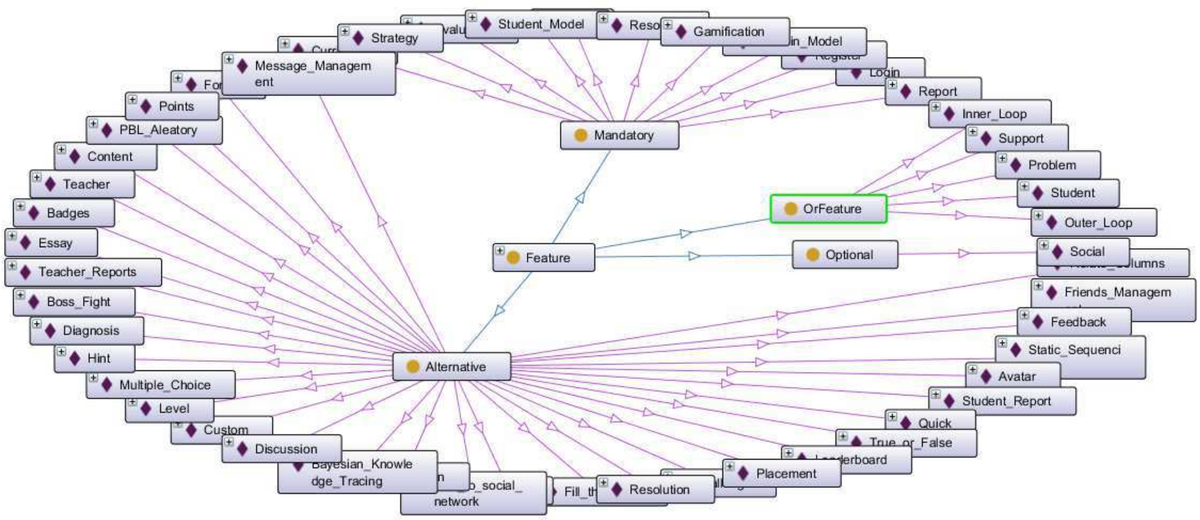


Figure 4.5: Ontology individuals of the gamified ITS feature model represented in GITS-PL.owl

[6]Available at `http://surveys.nees.com.br/ontologies/MeuTutor-Enem.owl`

Listing 4.1: Excerpt of the MeuTutor-ENEM.owl

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://surveys.nees.com.br/MeuTutor-Enem"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     ontologyIRI="http://surveys.nees.com.br/MeuTutor-Enem">
    <Prefix name="" IRI="http://surveys.nees.com.br/MeuTutor-Enem"/>
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <Prefix name="xml" IRI="http://www.w3.org/XML/1998/namespace"/>
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
    <Prefix name="gits-pl"
        IRI="http://surveys.nees.com.br/ontologies/GITS-PL.owl#"/>
    <Import>http://surveys.nees.com.br/ontologies/GITS–PL.owl</Import>
    <DataPropertyAssertion>
        <DataProperty IRI="gits-pl#currentState"/>
        <NamedIndividual IRI="gits-pl#Badges"/>
        <Literal datatypeIRI="rdf#PlainLiteral">Selected</Literal>
    </DataPropertyAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="gits-pl#currentState"/>
        <NamedIndividual IRI="gits-pl#GITS-PL.owl#Story"/>
        <Literal datatypeIRI=rdf#PlainLiteral">Eliminated</Literal>
    </DataPropertyAssertion>
    ...
</Ontology>
```

In this way, once a particular configuration of gamified ITS is specified in the ontology-based feature model defined, the variability of these systems could be automatically reasoned by third-party softwares to retrieve such configurations. Moreover, although not in the scope of this thesis, the knowledge representation gamified ITS configurations might also support further reconfigurations of gamified ITS features according to specific reasons such as learners' performance and motivation.

4.4 Concluding remarks

In this chapter we presented the reference feature model of gamified ITS that we specified and the ontology-based feature modeling approach (OntoSPL) proposed to support automatic analysis of such feature model at runtime. We described the features included in the model, besides describing the OntoSPL ontology as well as explaining how we have evaluated this ontology in comparison to other approach from the literature. We also described how we used OntoSPL to represent the gamified ITS feature model using ontologies, illustrating a configuration of MeuTutor-ENEM.

Note that the gamified ITS feature model defined in this chapter is still not sufficiently considering theoretical aspects of gamification and design practices of gamification. In the next chapter, we present the gamified tutoring ontological model that represents the knowledge about theories and evidence-supported design practices of gamification in connection with ITS. This knowledge will be further used to constrain the design space of gamified ITS with the aim of better supporting authoring for teachers.

Chapter 5

Gamified tutoring ontology

In this chapter, we present an ontological model that connects gamification concepts and design principles to ITS concepts, i.e., from ITS components such as domain, student and pedagogical models. To conceptualize this model, we first analyze the literature to identify particular behaviors that studies report positive effects about the use of game design elements combinations (Section 5.1). Next, in Section 5.2, we formalize a gamification domain ontology that represents core concepts about gamification as well as concepts considering specific gamification theories and frameworks; we also conceptualize evidence-supported gamification design practices, identified through the analysis of the literature, in such ontology. In Section 5.3, we integrate the concepts formalized in the gamification domain ontology to ITS concepts defined in an existing ITS ontology to specify our ontological model for gamified tutoring. We also present, in Section 5.4, how we have evaluated the ontological model developed using an ontology evaluation method based on knowledge representation roles. Finally, in Section 5.5, we conclude this chapter summarizing the main contributions presented in this chapter.

5.1 Gamification target behaviors in e-learning context

Recall that we intend to constrain the gamification design space of gamified ITS (as shown in Figure 4.1) based on evidence-supported design practices. As such, in this section, we identify particular target behaviors based on empirical results reported by the literature in the educational context. Hence, we expect to provide better support for teachers by leveraging these practices when they customize the features of a gamified ITS, as will be further presented in Chapter 6

As previously mentioned, there are many game design elements (e.g., points, badges, levels, leaderboard, etc) that could be used along with educational systems. Researchers are increasingly investigating the effects of gamification at several application contexts, including education [Nacke and Deterding, 2017]. In fact, identifying which game design elements effectively benefit learning performance as well as motivation and engagement of students is still an open issue. For instance, several works included in systematic literature reviews [de Sousa Borges et al., 2014, Hamari et al., 2014b, Seaborn and Fels, 2015] present combinations of game design elements that might be more amenable to effectively achieve particular behaviors. As such, to identify which game design elements combinations might be effective for learners in the e-learning context, we analyze the empirical works that provide evidence for using particular combinations of game design elements to target specific behaviors in the e-learning domain.

To analyze the empirical works included in the reviews with respect to educational contexts, we use the framework proposed by Hamari et al. [2014b], which was described in Section 2.2. This framework conceptualizes gamification as a process which includes motivational affordances, psychological outcomes and behavioral outcomes. According to this conceptualization, gamification is defined as a process of enhancing services with (motivational) affordances in order to invoke gameful experiences (psychological outcomes) and further behavioral outcomes. Thus, for each paper that present empirical evidence on the effect of using game design elements (motivational affordances) to target behavioral outcomes (e.g., improving learning outcomes, increasing engagement, and so on etc.), we used Hamari's framework to classify it.

Based on the classification of game design elements and behavioral outcomes the elements help to achieve, we group the effects of these elements by behavioral outcomes. Thus, we identified five main behavioral outcomes achieved by the use of gamification in the studies: participation, performance, enjoyment, exploration, competition and effectiveness.

We summarize the target behaviors we identified along with the game design elements that might help to achieve them based on the works analyzed in Table 5.1. The mapping of behavioral outcomes and motivation affordances (i.e., game design elements) are used to constrain the design space of gamified ITS considering empirical studies on the topic as well as used in the conceptualization of a gamification domain ontology, as presented in the following sections. The behaviors are described on below:

Table 5.1: Summary of target behaviors and game design elements

Target Behavioral Outcome	Game Design elements
Participation	Story, Rewards, Badges, Levels, Challenge, Leaderboard, Points
Performance	Story, Feedback, Rewards, Badges, Levels, Challenge, Leaderboard, Points
Competition	Leaderboard, Points
Enjoyment	Story, Rewards, Badges, Avatars, Challenge, Points
Exploration	Levels, Challenge, Boss Fight
Effectiveness	Leaderboard, Badges, Points

- Participation: this behavior includes game design elements that are more amenable to increase the level of participation/engagement of students based on the results provided by Denny [2013], Domínguez et al. [2013], Fitz-Walter et al. [2012], Foster et al. [2012], Goehle [2013], Halan et al. [2010], Li et al. [2012], Snyder and Hartig [2013], Spence et al. [2012]. It may include the following elements: *Challenge*, *Levels*, *Leaderboard*, *Story*, *Badges*, *Rewards*, and *Points*;

- Performance: this behavior includes game design elements that were used by several works [Cheong et al., 2013, Domínguez et al., 2013, Hakulinen et al., 2013, Smith and Baker, 2011] suggesting the their use for increasing students' learning outcomes. It includes the following elements: *Story*, *Feedback*, *Rewards*, *Badges*, *Challenges*, *Leaderboard*, *Points*, and *Levels*;

- Enjoyment: this behavior encompasses the game design elements used in the empirical works that are amenable to increase students' enjoyment (i.e., fun) [Denny, 2013, Hernández Ibáñez and Barneche Naya, 2012, Landers and Callan, 2011, Li et al., 2012]. The following game design elements are included in this behavior: *Story*, *Rewards*, *Badges*, *Points*, *Avatar*, and *Challenges*;

- Exploration: this behavior is supported by some empirical works [Fitz-Walter et al., 2011, Spence et al., 2012] which suggest that using some game design elements could enhance the exploration of the educational system by students. The following elements are included within this category: *Levels*, *Challenge*, and *Boss fight*;

- Competition: this behavior is suggested by the results provided by Domínguez et al. [2013]. Using *Leaderboard* and *Points* may enhance competition between students, which we define as *Competition* behavior;

- Effectiveness: we also defined an additional target behavior based on Domínguez et al. [2013], which we call *Effectiveness* behavior. This behavior suggests that using *Leaderboard*, *Badges*, *Points* there might be an increase in students' effectiveness while they interact with the educational system.

5.2 GaDO: Gamification Domain Ontology

As previously mentioned, gamification is an emerging topic with several concepts, theories, and definitions. Thus, during our ontological model engineering process, we decided to represent core concepts (e.g., gamification definition, game design element, player model, and so on) regarding gamification domain and specific gamification concepts (e.g., gamification design framework, gamification design practices, specific player models, and so on) in two different ontologies in our model. In this way, as we are representing concepts concerning the gamification domain, we developed a domain ontology to represent these concepts.

In the following sections, we presented how we developed the Gamification Domain Ontology (GaDO) including the two sub-domain ontologies: GaDO-core and GaDO-full. Next, in Section 5.3, we present how we specified an additional ontology that indeed connects the concepts of these ontologies with ITS concepts, called Gamification Tutoring Ontology (GaTO). Figure 5.1 presents an overview of the ontologies illustrating how they are related to each other. In order to develop these ontologies, we used the METHONTOLOGY approach, which is an ontology engineering methodology that is divided into seven main phases [Fernández-López et al., 1997, Gómez-Pérez, 1996], as explained in Section 2.5.3. Our decision on such methodology was made since it is listed as one of the most mature ontology engineering methodologies existing in the literature. Moreover, it includes activities to support most activities of the ontology development life-cycle [Bautista-Zambrana, 2015, Corcho et al., 2003].

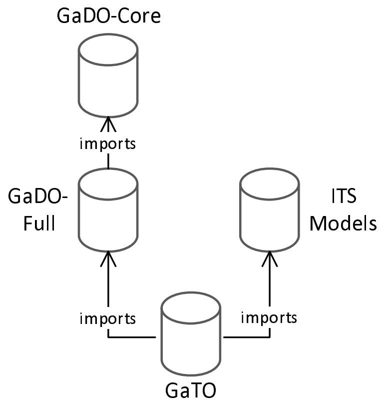


Figure 5.1: Ontological model illustrating the relationship between gamification and ITS ontologies

5.2.1 Gamification Domain Ontology (GaDO) – Core

As explained in Section 2.5.3, the first step of the METHONTOLOGY is defining the specification step. In the specification step for this ontology, we first defined its scope. It mainly considers core concepts regarding the gamification definition, which includes, for example, game design element and context. It involves players, player model and player type abstract concepts regarding a specific gamified context. We also specify concepts regarding abstract theories of motivation and needs that are supporting gamification.

Our main sources for knowledge acquisition include the works by Deterding et al. [2011], Hamari et al. [2014b], Werbach and Hunter [2012] in order to specify gamification concepts according to the definition provided by these authors. We also relied on three systematic literature reviews – i.e., de Sousa Borges et al. [2014], Hamari et al. [2014b], Seaborn and Fels [2015] – that, as previously mentioned, summarize a plethora of studies that use gamification in several contexts. For each systematic literature review, we consider the whole list of papers included in it as sources of knowledge to conceptualize our ontology. In addition, we also take into account the work by Challco et al. [2014] since it presents an ontology that conceptualizes gamification to be applied in a specific kind of educational system, i.e., computer-supported collaborative learning (CSCL).

Next, following the METHONTOLOGY process, we performed the conceptualization of our ontology. This phase includes defining the core concepts, a glossary of terms, a tree of concepts, and binary-relations between the concepts in the ontology. Based on our sources of knowledge, we defined the following core concepts: Gamification, Game Design Element, Context, Motivation and Need Theory, Player, Player Model, and Player Type.

The next phase includes integrating the conceptualization with existing ontologies on the topic. However, we could not find any other gamification domain ontology that could be reused in our ontological model. One potential ontology for reuse is the one presented by Challco et al. [2014], however, although that work has been considered a source of knowledge for our ontological model, it is particularly tied to the context of CSCL. Thus, we could not reuse such ontology in our domain ontology.

In the implementation phase, we implemented the GaDO-core ontology in an RDF/OWL file[1] with the aid of Protégé tool. Figure 5.2 presents an excerpt of our ontology as a UML conceptual model. In the sequel, we explain each of its concepts and relations.

Based on the gamification definition provided by the sources of knowledge we considered, we linked the concept of *Gamification* with several core concepts of this ontology. As seen in Figure 5.2, *Gamification* can rely on a set of *Motivational and Need Theories* in order to afford motivation. Following its definition, it is applied to a non-game context and also makes use of different types of *Game Design Elements*, which can be one of three types: *Dynamic*, *Mechanic*, and *Component*. According to Werbach and Hunter [2012], each one of these types can be specialized in several other elements; they are suppressed from Figure 5.2. *Dynamic* can be one of the following types: *Constraints*, *Emotions*, *Narrative*, *Progression* and, *Relationships*. In turn, *Mechanics* can be *Challenges*, *Chances*, *Competition*, *Cooperation*, *Feedback*, *Resource Acquisition*, *Rewards*, *Status*, *Story*, *Theme*, *Transactions*, *Turns*, and *Win States*. The *Component* type can also be sub-specialized in several types: *Achievements*, *Avatars*, *Badges*, *Boss Fights*, *Collections*, *Combat*, *Content Unlocking*, *Gifting*, *Leaderboard*, *Levels*, *Points*, *Quests*, *Social Graph*, *Team*, *Time Constraint* and *Virtual Goods*. Another important concept of this ontology is the *Player*, which interacts in a particular context that can be *Game* or *Non-Game*. A *Player* is

[1]The implementation of this ontology is available at `http://surveys.nees.com.br/ontologies/gado_core.owl`

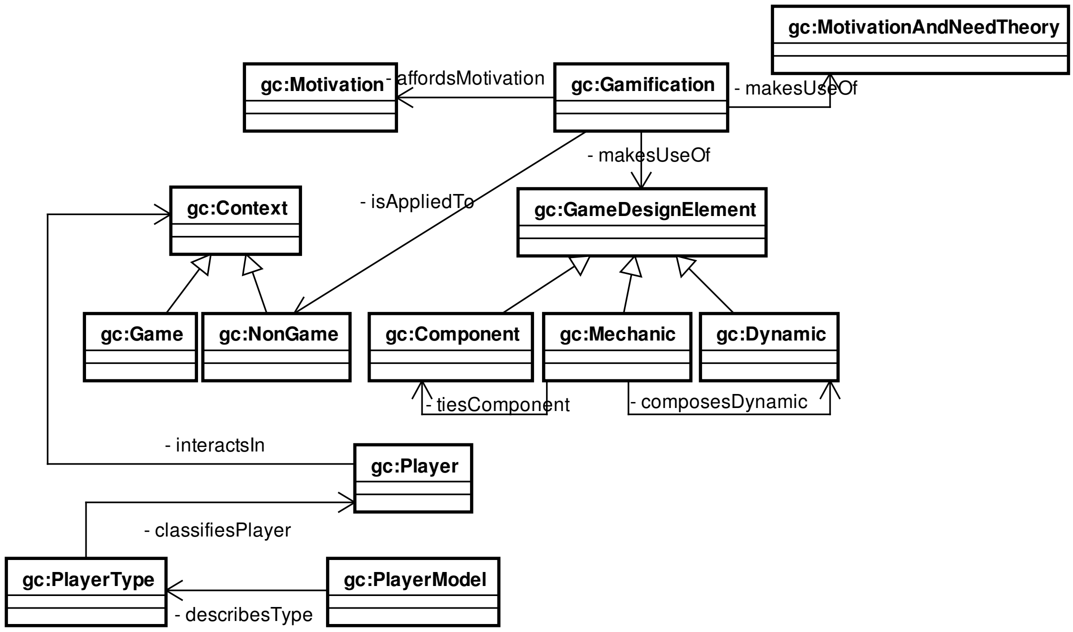


Figure 5.2: Excerpt of the Gamification Domain Ontology (GaDO) – Core. For the sake of clarity, we suppress the specialization for the Game Design Element types and some axioms. We use the prefix "gc" to refer to the concepts of this ontology

classified by a *Player Type*, whereas a *Player Type* is described by a *Player Model*.

The documentation of this ontology[2] was produced throughout the execution of all previous phases. Finally, in the last phase, we evaluate the generated ontology. However, as METHONTOLOGY does not explicitly define how to evaluate ontologies generated using such methodology, we choose our own strategy according to the existing works on ontologies evaluation. We explain how we evaluate this ontology as well as the other ontologies presented in this chapter in Section 5.4.

5.2.2 Gamification Domain Ontology (GaDO) – Full

In a similar way to the development of the GaDO-core ontology, we followed the METHONTOLOGY process steps for conceptualizing the GaDO-full ontology. As such, we first defined the scope of GaDO-full, which mainly considers a particular theory of motivation (i.e., Self- Determination Theory), a player model (i.e., BrainHex), and a

[2] Available at `http://surveys.nees.com.br/ontologies/documentation/GaDO-core.pdf`

gamification design framework (e.g., 6D framework) as well as how these concepts are linked to GaDO-core concepts. We also consider in the scope of this ontology the idea of gamification design practice, which is a pre-designed set of gamification elements linked to specific target behaviors that could be further used to aid the design of gamified ITS, following the mapping explained in Section 5.1.

Our main sources for knowledge acquisition include the works by Werbach and Hunter [2012] and Deci and Ryan [2010] in order to link the Self-determination theory concepts to GaDO-core concepts. We also relied on the work by Nacke et al. [2014] to specify the BrainHex player model along with its seven-player types. As gamification design framework, we chose the 6D framework since it is based on the Self-Determination Theory [Werbach and Hunter, 2012] and is the more comprehensive available gamification framework in the literature [Mora et al., 2015]. Thus, these references were also used as sources of knowledge to link the 6D framework to GaDO-core concepts. Additionally, we also relied on the systematic literature reviews (de Sousa Borges et al. [2014], Hamari et al. [2014b], Seaborn and Fels [2015]) as well as on the empirical papers listed in the reviews on the use of gamification in education to specify the concept of gamification design practice for the education context. As previously described, this concept is further used in the gamified tutoring ontology to constrain the gamification design space linking target behaviors to particular sets of game design elements based on the pieces of evidence provided by the empirical studies.

Next, we performed the conceptualization of this ontology. Likewise GaDO-core conceptualization, in this phase we define the core concepts, a glossary of terms, a tree of concepts, and binary-relations between the concepts in the ontology. Based on our sources of knowledge, we defined the following core concepts: Self-Determination Theory, Activity Loop, Engagement Loop, Motivational Affordance, Feedback, Target Behavior, Metric, Design Practice, and BrainHex Model.

As previously explained, the GaDO-full ontology makes use of the GaDO-core ontology to specialize particular concepts we are considering. In this way, in the integration phase of this ontology, we import the GaDO-core ontology in order to integrate this ontology's concepts to GaDO-core concepts. We could not find any other ontology that could be integrated to our ontology.

In the implementation phase, we also implemented the GaDO-full ontology in an RDF/OWL file[3] with the aid of Protégé tool . Figure 5.3 presents an excerpt of this ontology integrated with GaDO-core in a UML conceptual model – the blue classes represent the concepts of the GaDO-full ontology. In the sequel, we explain each of its concepts and relations as well as how they are integrated with GaDO-Core ontology.

The main concepts of GaDO-full ontology are related to the 6D framework components and how they are connected to GaDO-core ontology. This framework is supported by the *Self-Determination Theory*, which is represented in this ontology as a specialization of the *Motivation and Need Theory*, as shown in Figure 5.3. As described in Section 2.2.1, Werbach and Hunter [2012] establish that this framework has six steps: (i) Define business objectives; (ii) Delineate target behavior; (iii) Describe your players; (iv) Devise activity loops; (v) Don't forget the fun; and (vi) Deploy appropriate tools. Recall that our ultimate goal (which is not necessarily in the scope of this thesis) is to apply gamification to intelligent tutoring systems in order to increase engagement and motivation of students of these systems, expecting to increase their learning performance. Hence, this is the main general objective of this work. Indeed, only steps (ii), (iii) and (iv) are in the scope of this ontology conceptualization, since, the last two steps – i.e., (v) and (vi) may be only satisfied through the implementation of gamified intelligent tutoring systems. For instance, to not forget the fun it might be needed to investigate several aspects of the gamification design (components, mechanics, and dynamics game design elements).

As seen in Figure 5.3, a *Target Behavior* has a category (*TargetBehaviorCategory* class) and a success *Metric*. A *Target Behavior Category* can be one of the following types that we identified in Section 5.1: *Performance*, *Participation*, *Exploration*, *Enjoyment*, *Effectiveness*, and *Competition*. Although not explicitly presented in Figure 5.3, since the specializations of *Component* and *Mechanic* game design elements are suppressed for simplicity purpose, the design elements summarized in Table 5.1 are directly related (using object properties) to their correspondent target behavior category in the ontology.

Regarding activity loops (*ActivityLoop* class), its implementation intends to lead to particular target behaviors and they can be of two types: *Engagement Loop* and *Progressive*

[3]The implementation of this ontology is available at `http://surveys.nees.com.br/ontologies/gado_full.owl`

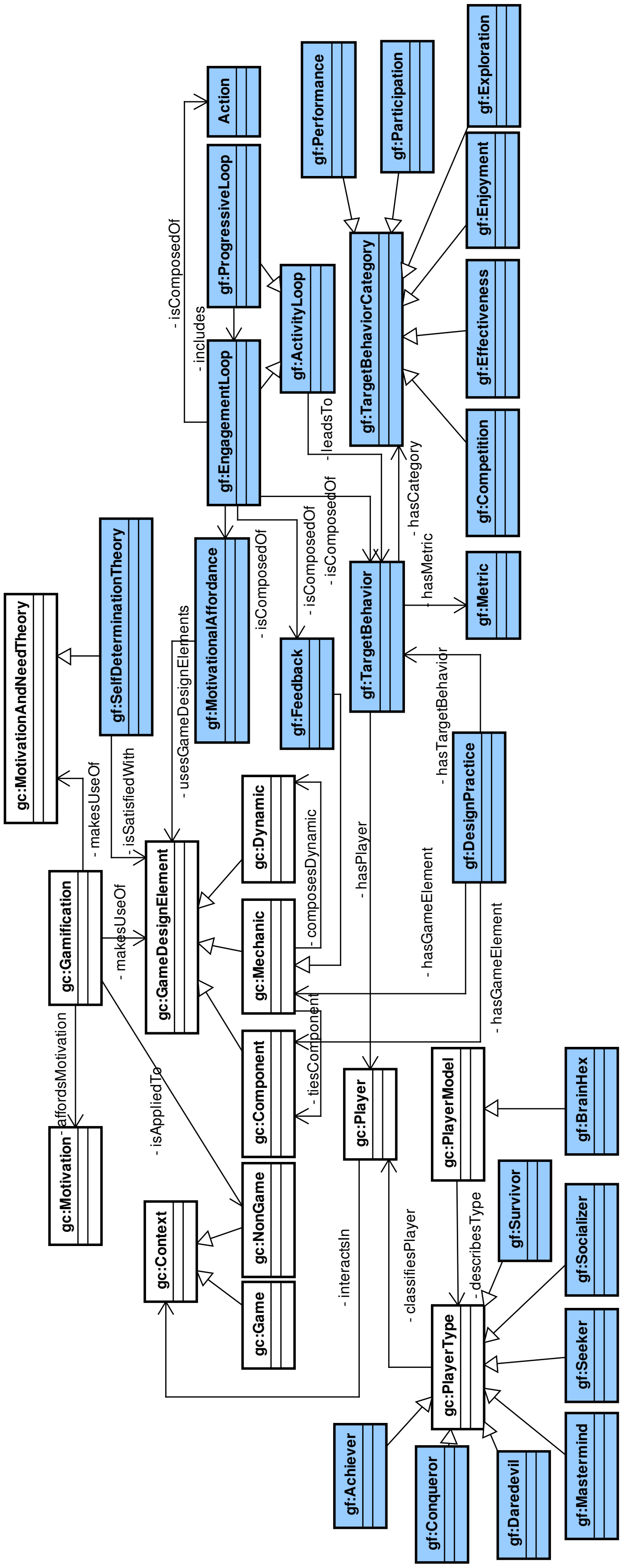


Figure 5.3: Excerpt of the Gamification Domain Ontology (GaDO) – full. Some classes and relations are omitted for clarity. We use the prefix "gf" to refer to the concepts of this ontology.

Loop. According to Werbach and Hunter [2012], an *Engagement Loop* is composed of three components: motivation, action, and feedback. In our conceptualization, motivation is represented by the use of *Motivational Affordances*, which are related to *Game Design Elements*, whereas *Feedback* is a *Mechanic* game design element. The *Action* component is connected to ITS concepts, since the interaction of the student in the tutor will occur with resources provided by it, as will be further explained in the GaTO ontology (Section 5.3). Furthermore, an *Engagement Loop* is also related to a *Target Behaviour*, which in turn is related to a particular *Player*. Moreover, a *Progressive Loop* includes the gamification design to drive different levels of gamification, thus, in our conceptualization we consider that it includes a set of *Engagement Loops* for each level. We also specify the *BrainHex* player model as a specialization of *Player Model* as well as its *Player Types*: *Achiever*, *Conqueror*, *Daredevil*, *Mastermind*, *Seeker*, *Socializer* and *Survivor* [Nacke et al., 2014].

Likewise GaDO-Core ontology, the documentation of this ontology[4] was produced throughout the execution of all previous phases. Finally, in the last phase, we evaluate the generated ontology, as will be further explained in Section 5.4.

5.3 GaTO: Gamified Tutoring Ontological Model

The main purpose of this ontology is connecting gamification and intelligent tutoring systems concepts. It includes representing ITS components – i.e., domain model, student model and pedagogical model – as well as their relationship with gamification concepts.

Our main sources for knowledge acquisition include the works considered in the gamification ontologies and theoretical works about ITS – i.e., the works of Du Boulay and Luckin [2001], Self [1998], de Barros Costa et al. [1998], Dillenbourg and Self [1992] and Self [1990]. In fact, for the sake of making use of existing work, these works are the theoretical background of the work proposed by Bittencourt et al. [2009], which presents an integrated ITS ontology that conceptualizes ITS components according to such works.

For conceptualizing this ontology, we also define the core concepts, a glossary of terms, a tree of concepts, and binary-relations between the concepts in the ontology. Based on

[4]Available at `http://surveys.nees.com.br/ontologies/documentation/GaDO-full.pdf`

our sources of knowledge, we explicitly defined the following core concepts: Gamified ITS, Domain Model, Student Model, Pedagogical Model and Gamification Model.

In the integration phase of this ontology, we import the GaDO-core and GaDO-full ontologies as well as the ITS ontology provided by Bittencourt et al. [2009] . Moreover, we also rely on existing RDF vocabularies – i.e., FOAF to represent personal data about students in the ontological model. We also implemented the GaTO ontology in an RDF/OWL file with the aid of Protégé tool[5]. Figure 5.4 presents an excerpt of this ontology integrated with GaDO-core, GaDO-full, and ITS ontologies in a UML conceptual model – the red classes represent concepts reused from the ITS ontology and the green classes represent the concepts of GaTO ontology. In the sequel, we explain each of its concepts and relations as well as how they are integrated with other ontologies.

The concepts of GaTO ontology represent the core concepts involved in a gamified intelligent tutoring system. As seen in Figure 5.4, besides including the three main ITS components – i.e., Student Model, Domain Model, and Pedagogical Model – a *Gamified ITS* also has a *Gamification Model*. The *Student Model* is connected to the ITS ontology through the *Behavioral Knowledge* concept, which is the representation of how a student behaves in the tutor, according to Dillenbourg and Self [1992]. It is also connected to the *Player* concept of the GaDO-core ontology to include students' behaviors as players. The *Pedagogical Model* is connected to the *Instructional Plan* ([Du Boulay and Luckin, 2001]) concept to represent the tutoring strategies that could be used in the tutor. The *Domain Model* is, actually, a concept from the ITS ontology provided by Bittencourt et al. [2009][6] and is related to the *Curriculum* concept. In turn, a *Curriculum* has a set of *Resources*, also referred as learning objects. Despite been suppressed in Figure 5.4 for clarity purposes, these resources can be of several types, for instance, Problem, Content, Concept, Question, Essay and so on. The *Gamification Model* is connected to the *Activity Loops* designed for that gamified tutor. Furthermore, the *Action* concept from the GaDO-full ontology, which is part of a particular *Engagement Loop*, makes use of *Resources* from the ITS ontology. This relationship enables that a specific *Engagement Loop* design considers the interaction with

[5]The implementation of this ontology is available at `http://surveys.nees.com.br/ontologies/gato.owl`

[6]Available at `http://surveys.nees.com.br/ontologies/its/its.pedagogical.owl`

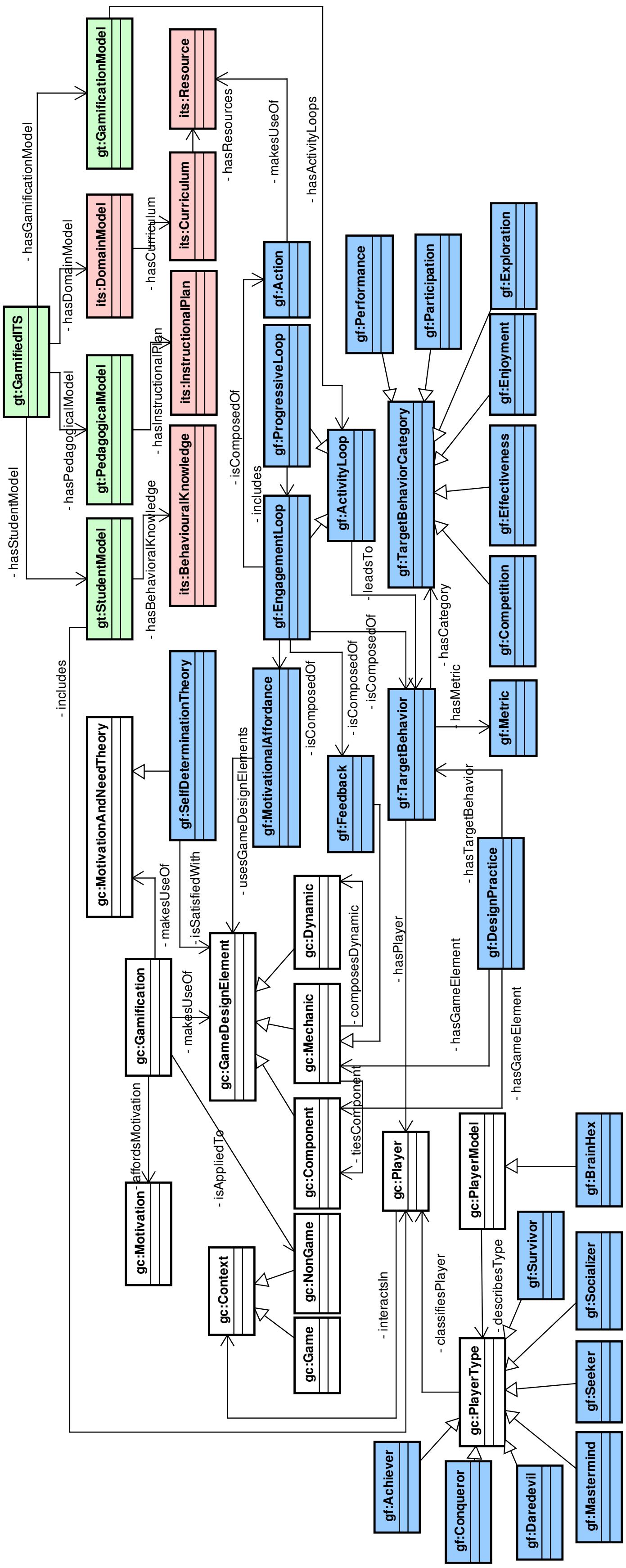


Figure 5.4: Excerpt of the Gamification Tutoring Ontology (GaTO). Some classes and relations are omitted for clarity. We use the prefix “gt” to refer to the concepts within GaTO ontology and “its” to refer to concepts from Bittencourt’s ontology.

Resources from an ITS *Domain Model*.

The documentation of this ontology was also produced throughout the execution of all previous phases[7]. In the next section, we also describe how we evaluate this ontology.

5.4 Evaluation of the ontologies

As previously mentioned, the METHONTOLOGY does not explicitly describe how to evaluate ontologies specified by following its steps. To evaluate our ontologies, we conduct a quantitative and qualitative evaluation with experts for each ontologies within our model.

5.4.1 Method

We used the FOCA methodology [Bandeira et al., 2016] to evaluate our ontology model. Our choice for such methodology was due because, in comparison to other ontologies evaluation strategies reported in the literature [Gangemi et al., 1996, Gómez-Pérez, 1996, Gruber, 1995, Obrst et al., 2007, Staab and Studer, 2013], this evaluation method strongly relies on the knowledge representation principles [Davis et al., 1993] as well as on constructs of other evaluation strategies to define a set of objective criteria to evaluate ontologies. The GQM (Goal-Questions-Metric) framework [Basili, 1992] is used to aid the evaluation process through a set of questions that are mapped to particular metrics. The output of the evaluation is an overall quality score as well as partial scores concerned to particular knowledge representation principles, for each evaluator.

According to Bandeira et al. [2016], the ontology evaluation is performed in three steps: (1) verifying ontology's type; (2) verifying questions and metrics, and (3) computing ontology's scores. In the first step, evaluators assign the type of the ontology that is evaluated. Table 5.2 presents the goals, questions, and metrics that are used to ascertain ontologies' evaluation (Step 2) using the FOCA methodology. It might be worth noting that the type of the ontology enables or disables some questions of the FOCA methodology. As explained by Bandeira et al. [2016], if an ontology's type is a domain or task one, the question 4 (Q4) must not be considered for the evaluation, whereas, if it is an application type, the question 5 (Q5) is not taken into account. The goals are inspired from the five knowledge representation

[7] Available at `http://surveys.nees.com.br/ontologies/documentation/GaTO.pdf`

Table 5.2: Goals, questions and metrics (along with a range of possible scores) of the FOCA methodology. Adapted from Bandeira et al. [2016].

Goal	Question	Metric	Range of scores
Substitute	Q1 – Are the ontology's competences defined?	Completeness	
	Q1.1 – Is there a description of the ontology's objective in the documentation?		0, 25, 50, 75, 100
	Q1.2 – Is there a description of the ontology's target public in the documentation?		0, 25, 50, 75, 100
	Q1.3 – Are there use scenarios in the documentation?		0, 25, 50, 75, 100
	Q2 – Is the ontology addressing the defined competences?		0, 25, 50, 75, 100
Ontological commitment	Q3 – Does the ontology reuse other ontologies?	Adaptability	0, 100
	Q4 – Does the ontology require a minimal knowledge commitment?	Conciseness	0, 25, 50, 75, 100
	Q5 – Does the ontology require a maximum knowledge commitment?	Conciseness	0, 25, 50, 75, 100
	Q6 – Are the ontology's properties coherent with the domain?	Consistency	0, 25, 50, 75, 100
Intelligent reasoning	Q7 – Are there contradictory axioms?	Consistency	0, 25, 50, 75, 100
	Q8 – Are there redundant axioms?	Conciseness	0, 25, 50, 75, 100
Computational efficiency	Q9 – Does the reasoner present modeling errors?	Computational efficiency	0, 25, 50, 75, 100
	Q10 – Does the reasoner run in a fast way?	Computational efficiency	0, 25, 50, 75, 100
Human expression	Q11 – Is documentation consistent with the modeling?	Clarity	
	Q11.1 – Are the terms presented in the ontology's documentation consistent with ontology's modeling?		0, 25, 50, 75, 100
	Q11.2 – Is there rationale and explanation of the terms presented in the ontology's documentation?		0, 25, 50, 75, 100
	Q12 – Are the concepts well-written?		0, 25, 50, 75, 100
	Q13 – Are there annotations in the ontologies defining the concepts?		0, 25, 50, 75, 100

roles described by Davis et al. [1993]. For each goal, a set of questions is defined in order to match goals to a quantifiable metric, which are used to compute the overall score (Step 3) of the ontology evaluation.

The overall score for an evaluator i is calculated by the Equation 5.1 on below. This same equation may be also used to calculate the partial score regarding each one of the coefficients related to the goals, for instance, to compute the score regarding the substitute goal (Cov_S), it is only necessary to use the equation on below canceling the other coefficients (Cov_{Oc}, Cov_{IR} and, Cov_{Ce}).

$$Score_i = \frac{e^{\{-0.44+0.03(Cov_S)_i+0.02(Cov_{Oc})_i+0.01(Cov_{Ir})_i+0.02(Cov_{Ce})_i-0.66GExp_i\}}}{1 \quad + \quad e^{\{-0.44+0.03(Cov_S)_i+0.02(Cov_{Oc})_i+0.01(Cov_{Ir})_i+0.02(Cov_{Ce})_i-0.66GExp_i\}}} \quad (5.1)$$

Where:

- Cov_S is the average score for the Substitute goal's questions, including sub-questions;

- Cov_{Oc} is the average score for the Ontological commitment goal's questions – note that the ontology's type modifies the computation of this variable. If it is a task or domain ontology this variable does not take into account Q4, whereas if it is an application one, Q5 is not considered for this score;
- Cov_{IR} is the average score for the Intelligent reasoning goal's questions;
- Cov_{Ce} is the average score for the Computation efficiency goal's questions;
- $GExp$ indicates the evaluator experience with the use of ontologies, if the experience is greater than 3 years, it receives 1, whereas it receives 0.

5.4.2 Procedure and participants

As suggested by the FOCA methodology, the evaluation should involve the participation of human agents. Five people with experience in the use of ontologies as well as on the ontologies' domain topics – i.e., gamification and intelligent tutoring systems were selected. Among these people, four of them are from academic settings. One is an undergraduate student in Computer Science, one is a Ms.C in Computer science (which has a master thesis in the ontology topic), the last one is a Ph.D. Student – which works with gamification and ontologies in the context of computers and education, and the last one is a Ph.D. professor that has as research interests gamification, ITS, and ontologies topics. Moreover, one other participant comes from industry, and has a Ms.C in Computer Science, his thesis involved computers and education, ontology and gamification topics. All participants had prior knowledge on ontology and prior experience with the Protégé tool.

Each of the ontologies presented in this chapter used the same participants, and Table 5.3 shows their experience information in the topics of the ontologies as well as the settings on which the participants are inserted.

To instrument our ontological model's evaluation, for each ontology of our model (i.e., GaDO-core[8], GaDO-full[9], and GaTO[10]), participants were introduced to the ontologies along with their documentation through a survey. The Steps 1 and 2 of the evaluation are included

[8]Evaluation form available at `https://goo.gl/forms/bjKuhVp4ChCEo2ih2`
[9]Evaluation form available at `https://goo.gl/forms/UiF5DxJ9baCAnMLo2`
[10]Evaluation form available at `https://goo.gl/forms/eZYkaDYobickC7m92`

Table 5.3: Participant experience per each topic and settings

Participant	Exp. in Ontologies	Exp. in Gamification	Exp. in ITS	Settings
P1	> 3 years	< 1 year	< 1 year	Academic
P2	> 3 years	< 1 year	< 1 year	Industrial
P3	>1 and < 3 years	>1 and < 3 years	> 3 years	Academic
P4	> 3 years	> 3 years	> 3 years	Industrial
P5	> 3 years	> 3 years	> 3 years	Industrial

in the three surveys, asking participants to assign which is the type of each ontology as well as to answer the questions presented in Table 5.2. We also collect from the participants their experience with ontologies, gamification, and intelligent tutoring systems as well as qualitative data about the positive and negative aspects of our ontologies.

5.4.3 Results

This section presents the analysis of the data collected in the evaluation with participants. The collected data as well as the scripts and spreadsheets used in the experimental analysis are available at `http://surveys.nees.com.br/ontologies/documentation/Analysis.rar`. In the following section, we present the descriptive statistics of our results.

Descriptive statistics

The collected data contains the participants' answers to the questions shown in Table 5.2 in each one of the three ontologies. Based on those answers, we compute the ontologies' overall score as well as the score regarding the four representation knowledge goals presented in Equation 5.1. Thus, we conduct a descriptive analysis of the data, by analyzing histograms and boxplots of the ontologies' scores. Figure 5.5 presents the boxplots for each score evaluated comparing the results for the three ontologies. We also summarize the statistics of each one of the ontologies over the five scores. Table 5.4 presents the summary of statistics for the Cov_S, Cov_{Oc}, Cov_{Ir}, Cov_{Ce}, and $Score$ metrics per each ontology evaluated.

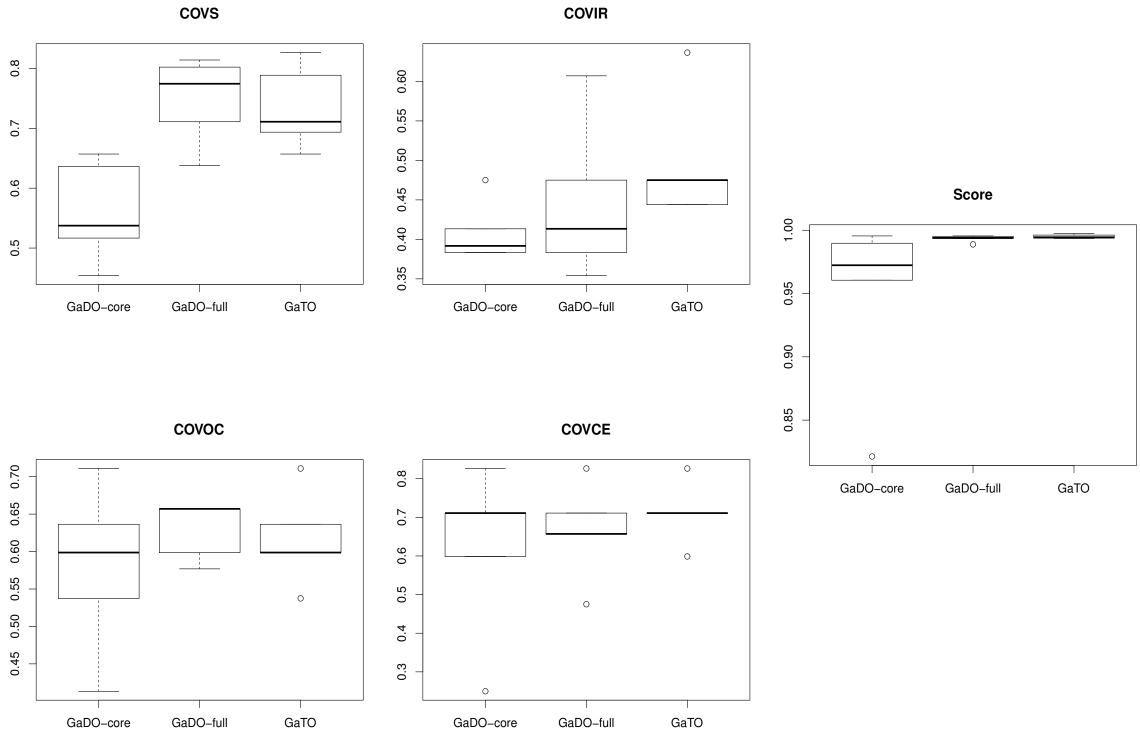


Figure 5.5: Boxplots comparing the five scores for the three ontologies

Table 5.4: Summary of statistics of the five metrics per each ontology evaluated

Goal	Ontology	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Sd.
	GaDO-core	0.4543	0.5167	0.5374	0.5604	0.6365	0.657	0.084877
Substitute (Cov_S)	GaDO-full	0.638	0.7109	0.7744	0.7479	0.8022	0.8141	0.073267
	GaTO	0.657	0.6935	0.7109	0.7353	0.7886	0.8264	0.070003
	GaDO-core	0.4134	0.5374	0.5987	0.5794	0.6365	0.7109	0.112089
Ontological Commitment (Cov_{Oc})	GaDO-full	0.5769	0.5987	0.657	0.6293	0.657	0.657	0.038691
	GaTO	0.5374	0.5987	0.5987	0.6164	0.6365	0.7109	0.06365
	GaDO-core	0.3834	0.3834	0.3917	0.4094	0.4134	0.475	0.038673
Intelligent Reasoning (Cov_{Ir})	GaDO-full	0.3543	0.3834	0.4134	0.4466	0.475	0.6071	0.100202
	GaTO	0.444	0.444	0.475	0.4949	0.475	0.6365	0.080641
	GaDO-core	0.2497	0.5987	0.7109	0.6193	0.7109	0.8264	0.221737
Computational Efficiency (Cov_{Ce})	GaDO-full	0.475	0.657	0.657	0.6653	0.7109	0.8264	0.126852
	GaTO	0.5987	0.7109	0.7109	0.7116	0.7109	0.8264	0.080497
	GaDO-core	0.8213	0.9605	0.9723	0.9479	0.9897	0.9955	0.072121
Overall Score ($Score$)	GaDO-full	0.9888	0.9935	0.994	0.9934	0.9951	0.9955	0.002672
	GaTO	0.9935	0.9937	0.9945	0.995	0.9962	0.9973	0.001637

Assumptions verification and inferential statistics

The statistics presented are very useful to understand the overall behavior of the data regarding the scores. However, we can also analyze it to discover if there are statistically significant differences between the ontologies regarding those scores. In this way, although our intention is not discovering which ontology is better with respect to the aforementioned metrics, we compare ontologies with each other to understand if the specified ontologies have similar scores according to the FOCA methodology. Hence, we applied non-parametric tests to compare the ontologies alternatives considering the hypotheses presented in Table 5.5.

Table 5.5: Hypotheses of the evaluation

H1-0: The substitute role of the ontologies is equal
H1-1: The substitute role of the ontologies is different
H2-0: The ontological commitment of the ontologies is equal
H2-1: The ontological commitment of the ontologies is different
H3-0: The intelligent reasoning of the ontologies is equal
H3-1: The intelligent reasoning of the ontologies is different
H4-0: The computational efficiency of the ontologies is equal
H4-1: The computational efficiency of the ontologies is different
H5-0: The overall scores of the ontologies are equal
H5-1: The overall scores of the ontologies are different

To verify how the ontologies' scores are compared with each other, statistical tests are applied for each one of the scores. The data of all five scores (at least for one of the ontologies) are not normal (i.e., the Shapiro–Wilk and Anderson Darling test were applied). As such, we apply a Kruskal-Wallis test to compare all three ontologies' scores and, then, we apply the Wilcoxon Test to compare the ontologies in pairs.

Table 5.6 presents the results of the hypotheses tests application. As shown in Table 5.6, the first column describes which metric is tested, the second one presents the p-values of the Kruskal-Wallis Test – considering as the null hypothesis that the values on all three ontologies are equal (Table 5.5). The third, fourth and fifth columns present the p-values of the ontologies' comparison, in pairs. As seen in the table, the null hypothesis for the group comparison is only rejected for the Substitute and Overall Score, respectively, with 5% and 10% of significance. Moreover, regarding the Substitute score, the null hypotheses for the comparison between GaDO-core and GaDO-full as well as between GaDO-core and GaTO are both rejected, with 5% of significance. Indeed, our results showed that the Substitute score for the GaDO-core is lower (with statistical significance) than the scores for GaDO-full and GaTO ontologies. With respect to the Intelligent Reasoning score, the null hypothesis for comparing GaDO-core and GaTO is rejected with 10% of significance, showing that the score for the GaTO ontology is better than for GaDO-core. Our tests also suggest that the null hypothesis for the comparison between the Overall Score of the GaDO-core and GaTO is rejected with 10% of significance. Concerning the Ontological Commitment and Computation Efficiency scores, our results showed that there's no statistical difference in all comparisons.

Table 5.6: P-values after applying Kruskal-Wallis and Wilcoxon tests (O1 = GaDO-core, O2 = GaDO-full, O3 = GaTO). 90% and 95% confidence levels are represented, respectively, by * and **

Goal	$\mu_{O1} = \mu_{O2} = \mu_{O3}$	$\mu_{O1} = \mu_{O2}$	$\mu_{O2} = \mu_{O3}$	$\mu_{O2} = \mu_{O3}$
Substitute (Cov_S)	0.01557**	0.01587**	0.01597**	0.9166
Ontological Commitment (Cov_{Oc})	0.6671	0.4578	0.7488	0.6684
Intelligent Reasoning (Cov_{Ir})	0.1552	0.8315	0.05547*	0.2888
Computational Efficiency (Cov_{Ce})	0.7453	1	0.6536	0.5152
Overall Score ($Score$)	0.0977*	0.1732	0.05556*	0.4633

5.4.4 Analysis and discussion

In our evaluation, we also collect from the participants their comments about positive and negative aspects of our ontologies. By analyzing these comments, we can better understand what are the main reasons for the results that we have found. As previously explained, our results are only statistically significant for the Substitute and Overall Scores. Hence, we mainly focus on analyzing participants' comments aiming to explain these results.

Regarding the Substitute Score, the GaDO-core ontology received the slighter score in comparison to the other two ontologies. One participant mentioned the following statement: "I've missed some rdfs:comment in some properties in the ontology, data and object properties.". Other two participants also mentioned that there was a lack of explanation in ontologies' properties. Another participant also states that some terms used in the ontology's descriptions are not consistent with the presented description. Two participants also commented that the ontology is not reusing any other ontology. All these comments might impact on the Substitute score since they are related to the questions Q1 and Q3. By analyzing the comments for the other two ontologies, we can observe that the comments regarding this role are less frequent. However, participants also describe a lack of annotations, been more frequent in the comment to the GaDO-core ontology.

With respect to the Overall Score, we may note that the number of participants' comments might have impacted it. Among the five participants that evaluated the GaDO-core ontology, four mentioned that there is a lack of annotations on some classes and/or properties. Two of them stated that there are problems in the definition of some classes, whereas the same number of participants also mention some confusion in the relation between some classes, for instance, between Game Design Element and Motivation and Need Theories. Moreover, two participants complained about the lack of reuse – one of them suggested to use the foaf ontology in the Player class. Finally, one participant mentioned that some terms are not consistent with classes' descriptions, and there was also one comment about problems using the reasoner. Among the participants that evaluated the GaDO-full ontology, there were two comments mentioning domain consistency problems (e.g., conceptualization using sub-classes in the Self-Determination Theory class instead of using object properties). Two participants also commented about the lack of annotations in some classes and properties, whereas there were also two comments about problems using the reasoner. Concerning the GaTO-ontology, there were also two comments mentioning the lack of annotations, one comment complaining about the lack of class definition, and one comment suggesting to improve the ontology's documentation in a general way.

Although the comments presenting some drawbacks for our ontologies, participants have also mentioned several positive aspects of them. In the GaDO-core evaluation, participants emphasized that it is easy to understand the ontology (two participants), the terms are well-written (1 participant), there is a good abstraction of the domain (1 participant), the ontology is well-designed (1 participant), and the documentation is providing a good explanation of the ontology. The comments regarding the GaDO-full include the following positive aspects: the terms are clear and well-written (2 participants), the ontology is complete (2 participants), the ontology is suitable to be applied in an educational context (1 participant), there's a good abstraction of the domain (1 participant), and there is reuse of other ontologies (1 participant). Finally, in the evaluation of the GaTO ontology, some aspects were also stressed: the terms are also well-written (1 participant), the ontology is concise (1 participant), there is a good abstraction of the domain (1 participant), there is a good level of completeness regarding the domain (1 participant) and the purpose of ontology is satisfied by connecting gamification and ITS concepts (1 participant).

Afterwards, all the aforementioned comments provided by experts were used to improve our ontologies conceptualizations

5.4.5 Threats to Validity

This section describes concerns that must be improved in future replications of this study and other aspects that must be taken into account in order to generalize the results of the evaluation performed in this chapter. In general, the design of the evaluation aimed at minimizing a lot of the threats discussed in this section by using an objective evaluation method for ontologies (i.e., FOCA methodology). However, there are threats that should be considered. To organize this section, the threats to validity were classified using the Internal, External, Construct and Conclusion categories [Wohlin et al., 2012].

Internal

As the experiment involves the active participation of humans, it was prone to a number of internal threats, such as (i) history – it is possible that the moment at which the experiment occurred may have affected the results, however, this threat was minimized by letting participants evaluating the ontologies at anytime they preferred; and (ii) maturation – since the participants took around 45 minutes to finish all the tasks of the evaluation, it is possible that they were bored or tired during the last tasks.

Construct

The threats to the validity with respect to the construct category are closely related to the evaluation method used in the evaluation. Thus, we could not identify additional threats beyond the threats within the FOCA methodology evaluation method. However, we might be confident of this evaluation method since FOCA methodology is based on the roles for knowledge representation and all questions were validated with experts.

External

The sample of the evaluation is representative to the academic and industrial contexts. However, the academic context is only represented by two participants and the industrial context considers only our industrial partner (i.e., MeuTutor company), thus there might be an interaction of setting and treatment threat. In fact, it is difficult to generalize the results of the experiment to other evaluators. The setting of the evaluation must be broadened to other academic and industrial settings to obtain more generic results.

Conclusion

Furthermore, due to some restrictions, for instance, this evaluation demands participant experience in several topics (i.e., ontologies, gamification, and ITS), the sample size of the experiment was 5 participants (repetitions), thus, there might be insufficient statistical power on the effects of the evaluation. Finally, it is possible that random irrelevancies have occurred in the settings on which the participants evaluated the ontologies, e.g., noise, distractions and so on.

5.5 Concluding Remarks

Connecting gamification and ITS theories as well as providing design practices for applying gamification in ITS can contribute to the effective design of gamified ITS that take into account both learning performance and motivation of students. In this work, we connect some of these theories and define design practices for using gamification based on the literature by formally representing such concepts with the use of ontologies. Our ontological model is composed of three ontologies (i.e., GaDO-core, GaDO-full, and GaTO) and was developed following the guidelines of an ontology engineering methodology (i.e., METHONTOLOGY).

To empirically evaluate our ontological model, we used the FOCA methodology that is based on the five roles of knowledge representation. Evaluators are experts on ontologies as well as on gamification and ITS topics. The qualitative results of our ontologies' evaluation suggest that they provide a good abstraction of the domain. In addition, the results obtained with the quantitative evaluation allowed us to state: (i) there is significance on the effects of the ontology factor in the Substitute score and in the Overall score; (ii) the Substitute score of the GaDO-core ontology is lower than the scores of GaDO-full and GaTO ontologies; (iii) the Intelligent Reasoning score of the GaDO-core ontology is lower than the score of the GaTO; (iv) the Overall Score of the GaDO-core ontology is lower than the score of the GaTO ontology. (v) there is no significance on the effects of the ontology factor in the Ontological Commitment and Computational Efficiency scores; (vi) there is no statistical difference between the GaDO-full and GaTO ontologies regarding the Substitute score as well as in the Overall Score.

The results shown in this chapter can be used to continually improve our ontological model in order to indeed support the development of authoring tools for creating gamified ITSs. In the next chapter, we propose the development of an authoring solution that rely both on the artificial intelligence techniques to model students' behavior and motivation, to reason on the domain knowledge, to individualize tutoring for students, and so on; as well as on the human intelligence of teachers to customize gamified ITS that take into account the context on which the tutor will be executed and teachers' preferences.

Chapter 6

AGITS: an authoring solution for designing gamified intelligent tutoring systems

In this chapter, we present an authoring solution to aid teachers designing gamified ITS. Our solution makes use of the gamified ITS ontology-based feature model to automatically manage the variability of gamified ITS that can be produced using the authoring tool. It relies on the ontological model conceptualization that connects gamification concepts and design practices to ITS concepts in order to constrain the variability design space and to better support the authoring process for teachers. In Section 6.1 we present the authoring process we propose in this thesis considering the traditional ITS components and the inclusion of a gamification model aiming to guide the development of the authoring computational solution. Next, in Section 6.2, we describe how we developed the authoring solution, depicting the software engineering phases conducted in the development of the solution. In Section 6.3 we present the first empirical study that we conducted to evaluate the prototypes of our authoring solution in lab settings to explore some features of the solution and, in Section 6.4, we describe a second experiment conducted exclusively with teachers to investigate how the participants perceive our authoring solutions with respect to several metrics (e.g., usability, complexity, and so on). Finally, in Section 6.5, we conclude this chapter by summarizing the main contributions presented.

6.1 Gamified ITS development process

In this section we present a process designed to aid teachers in the authoring process of gamified ITS. Before presenting the details of this process, we present a general gamified ITS development process that is related to the authoring process.

A high-level gamified ITS development process was specified considering the four classic ITS components (i.e., domain, student, pedagogical and interface models) as well as a gamification model and extra ITS features. Figure 6.1 shows a big picture of the gamified ITS development process considering these components. Note that each activity is intertwined with the previous one and, before finishing the development, it is possible to return to each one of the previous activities to enable evolution and management of changes. Furthermore, this process might be a bottom-up strategy for developing gamified ITS since it starts with the three cornerstone ITS components (i.e., domain, student and pedagogical model) and ends in the interface model development. However, other top-down strategies that start with interface development and then focus on the other components or processes that enhance parallelism could also be used.

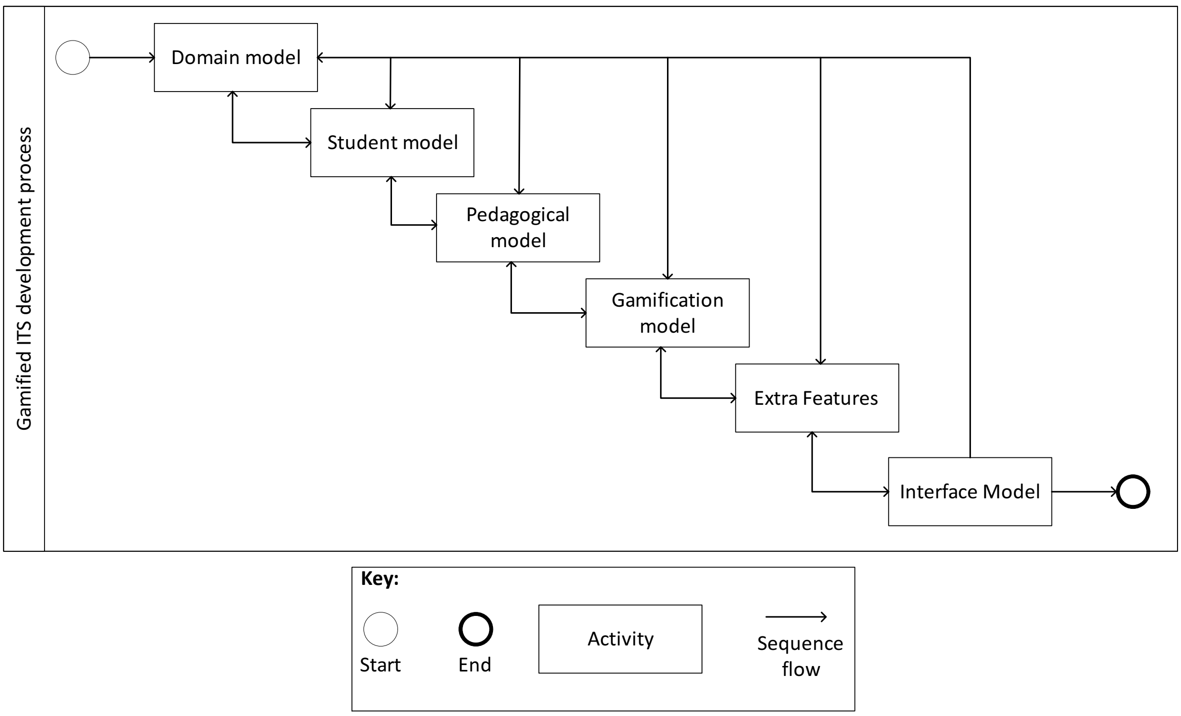


Figure 6.1: General gamified ITS development process

As shown in Figure 6.1, the activities are abstract enough to allow gamified ITS designers to use whatever sub-activities they need to develop their systems. This flexibility might be important since there is no agreement in the literature regarding the types of ITS, features to consider, and technologies to use in the development of ITS, as discussed in Sections B.5, B.6, and B.7, respectively. Note that, once each activity involves the development of software modules, traditional software engineering phases (i.e., requirements engineering, architectural design, implementation, and tests) must be followed inside each activity of the process. The four classic ITS components (domain, student, pedagogical and interface models) are explained in Section 2.1, in the following, we explain the extra activities that we are considering in this process: gamification model and general features.

The gamification model should consider all the features related to the inclusion of gamification in the ITS. For instance, game design elements (i.e., dynamic, mechanic and components) to include in a gamified ITS and how these elements are connected (e.g., gamification design, i.e., activity loops) to the learning contents, instruction and student knowledge behavior in the domain. Moreover, this model might also take into account strategies for personalizing the gamification and/or the tutor according to student characteristics (e.g., player type).

The *Extra Features* activity involves the development of additional features that could be included in a gamified ITS. For instance, features that enable teachers to manage the tutor defining a syllabus in the tutor. Teachers may also be able to check the performance of their students in the gamified tutor viewing reports. Moreover, collaborative features might also be enhanced in the tutor to enable students to interact with other students using social networks. In summary, this module may include any additional features that a team intends to include in the gamified ITS.

The authoring computational solution presented in this work relies on the general process for developing gamified ITS presented above. However, our solution does not support authoring for all the activities of this process. As previously discussed (Section B.4), the results of our systematic literature review on ITS authoring tools suggest that these tools could be used to design all four main classic ITS components. However, no paper targeted all ITS components in the same authoring tool, which may indicate that enabling authoring for all these components at the same time is not interesting. In fact, each component has its own function and unique properties which may be more or less amenable to authoring depending on several aspects, i.e., type of ITS, technologies used, needed pedagogical expertise, trade-off choices between usability and flexibility, and so on [Dağ et al., 2014, Murray, 2004, Sottilare, 2015]. For instance, if an ITS authoring tool allows authoring of all four ITS components, it might provide a high flexibility, but this would come at the expense of higher complexity and decrease in usability. On the other side, if an ITS authoring tool only provides authoring of few ITS components, it might have high usability and low flexibility levels.

With this in mind and considering we are proposing an authoring solution for teachers, our proposal must deal with the trade-off between flexibility and usability in its design. Our goal is not overloading teachers with many authoring activities and, at the same time, not constraining too much the authoring options for them, keeping the authoring process simple and usable. Hence, as shown in Figure 6.2, our solution does not target authoring for all activities presented the in gamified ITS development process, but for only four activities of the process.

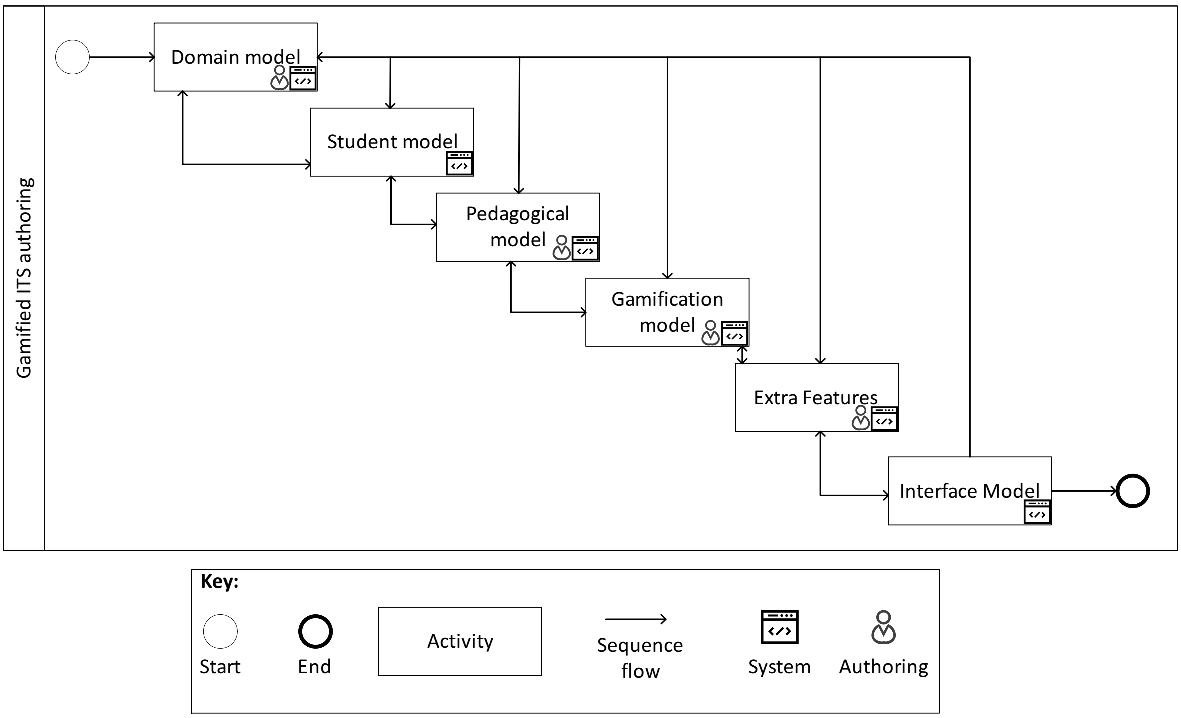


Figure 6.2: Gamified ITS authoring

As seen in the figure above, our authoring solution supports teachers in the *Domain model*, *Pedagogical model*, *Gamification model*, and *Extra Features* activities of the gamified

ITS development process. Note that we are not intending to support authoring for the *Student model* and *Interface model* activities. Although we found some works that enable *Student model* authoring (e.g., to configure student modeling rules [Chakraborty et al., 2010]), this activity strongly relies on the artificial intelligence features of tutors to automatic represent and update student models based on learner's actions. Hence, to not overload teachers with more authoring options, we decided to take more advantage of the artificial intelligence instead of human intelligence of teachers in this activity. Moreover, we also found some technological limitations that do not favor simple and usable authoring of the *Interface model*, thus we also decided to not support authoring for such activity. In the next section, we describe how authoring takes place in each one of the activities supported by our solution.

It is worth explaining how our authoring computational solution could be integrated, indeed, to a third-party gamified intelligent tutoring system. As shown in Figure 6.3, the authoring and gamified ITS modules can be interoperated by using the ontologies (i.e., GITS-PL.owl and GaTO.owl) specified in the previous chapters of this thesis. Our intention with this architecture is to providing a generic authoring solution that is independent from any particular gamified ITS platform. In fact, the GITS-PL ontology works as a contract between the software modules since it provides a shared and reasonable way to interoperate these systems. As previously explained, the reference feature model conceptualized in this ontology represents the design space of gamified ITS and the selections made by teachers are represented in a OWL file based on such ontology. On the other hand, a gamified ITS system must be able to deal with the configured ontology, which represents the desired configuration of a teacher, and self-reconfigure itself to such configuration of features. Note that this would require from third-party gamified ITS to implement some mechanism to manage the variability of its features at runtime by using some platform for reuse such as software product lines.

As previously mentioned, the GaTO ontology connects some gamification theories, frameworks and design practices to ITS concepts. This ontology aids the customization of gamified ITS by constraining the design space for the teachers in order to make the authoring process simpler and more usable. Additionally the role of the GaTO ontology for interoperating our authoring solutions and a gamified intelligent tutoring systems is twofold. First, it contains the knowledge about the domain model created by the teacher, which can

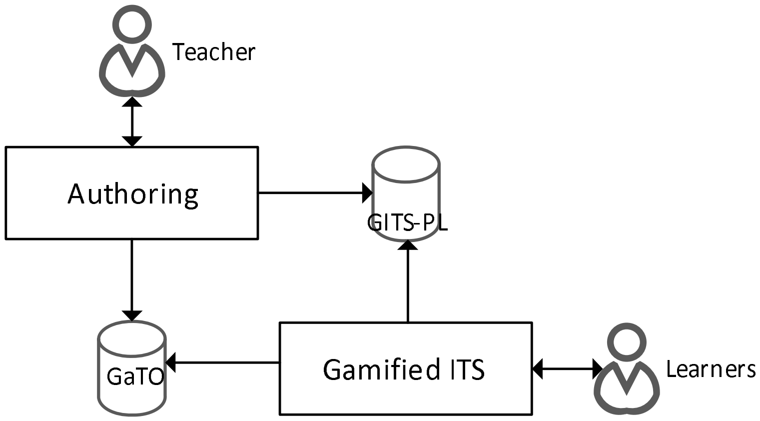


Figure 6.3: Illustration on how the ontologies are used to interoperate the authoring solutions and a third-party gamified ITS

be reasoned by a gamified ITS. Second, it also contains the decision on which gamification target behavior is selected by teacher which is used by a gamified ITS to activate several activity loops in the system.

Although we intend to propose a generic authoring solution for customizing gamified ITS, our computation solution might be constrained to particular types of ITS. As discussed in Section B.5, there are several types of ITS (e.g., example-tracing, model-tracing, cognitive tutors, content and problem-based tutors, and so on). However, the use of this authoring tool is constrained to the features represented in the feature model developed and particular types of ITS demand specific features we are might not considering in the design space. For example, for designing a example-tracing tutor [Aleven et al., 2016], it would be necessary to provide authoring for a behavior graph used in the tutor. Our computational solution is more amenable to provide authoring for content and problem-based tutors since we rely on problem-based solving pedagogical strategies (as presented in Section 4.1).

In the following section we describe the software engineering activities conducted for developing the authoring solution, depicting how the authoring process makes use of the ontologies previously presented.

6.2 Authoring computational solution

In this section we describe the software engineering activities performed to develop the authoring computation solution presented in this thesis (Figure 6.4). In the interest of clarity, the activities are sequentially presented, like a waterfall model. However, in fact it was conceived as an iterative and incremental process, which demanded lots of interaction between the activities. In the following sections, we describe each one of these phases.

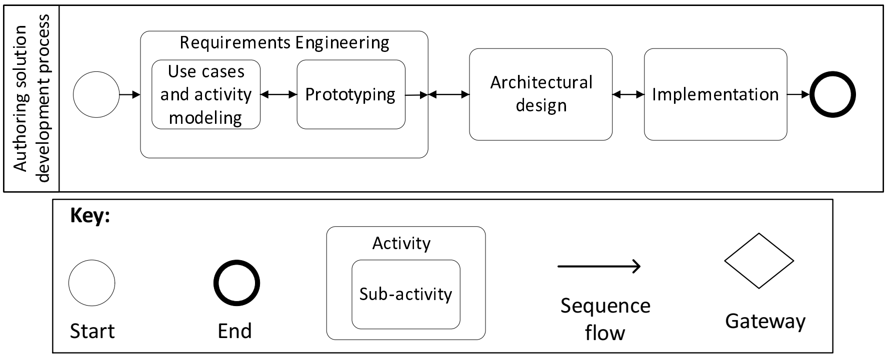


Figure 6.4: Authoring solution development activities

6.2.1 Requirements engineering

The authoring process supported by our solution aids teachers to author two main aspects of gamified ITS. First, it supports the configuration of gamified ITS features according to the reference feature model formalized in Section 4.1. Second, it supports authoring for educational resources (e.g., problems, content, and so on). Note that the functional requirements of this authoring solution are constrained by the design space defined in the reference feature models.

Use case and activities modeling

The main requirements of our solutions are presented in Figure 6.5. As shown in the use cases diagram, after a teacher logs in, he/she may create a tutor, edit a tutor or create educational resources. To create a tutor, a teacher can configure a tutor from scratch (*Configure tutor*) or apply a configuration template (*Apply tutor configuration template*). If a teacher decides to configure a tutor, he/she must define the curriculum of the tutor,

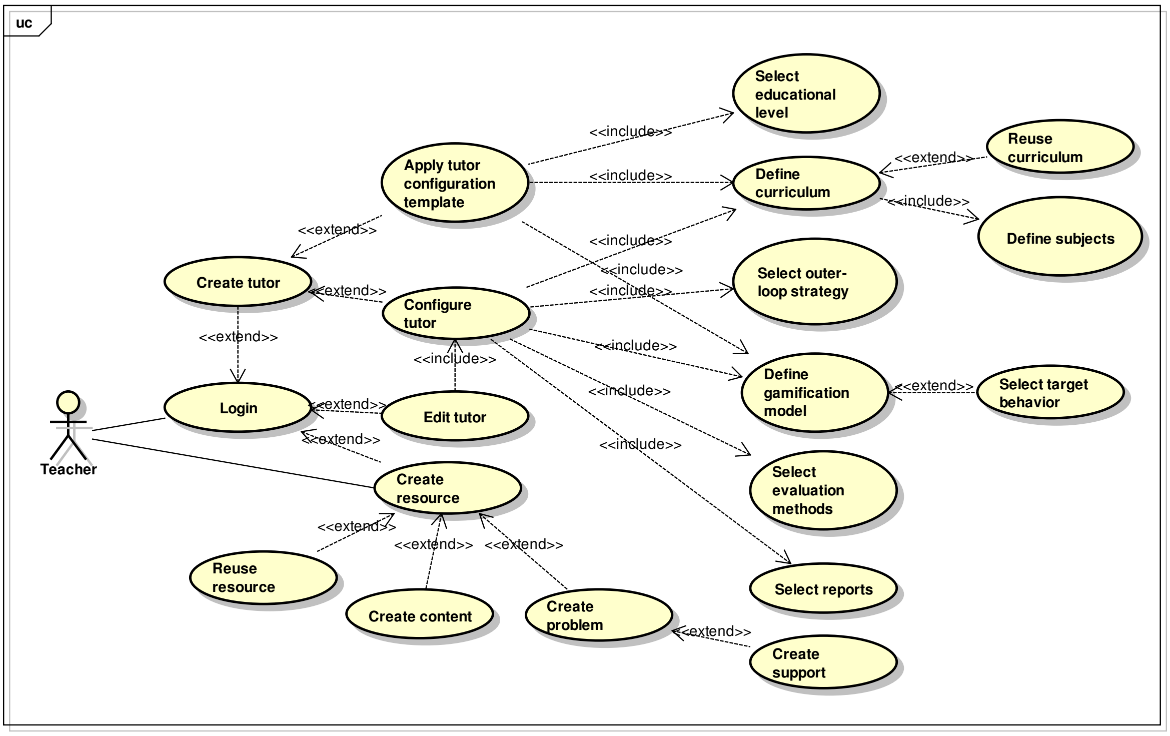


Figure 6.5: Authoring solution use cases

select an outer-loop strategy, define the gamification model, select evaluation methods, and select reports. When defining a curriculum, a teacher must define its subjects and can reuse an existing curriculum of the tutor. To define a gamification model, our solution enables teachers to select an expected target behavior or to select specific game design elements. If a teacher decides to apply a template, it is needed to select the educational level of the tutor (i.e., which is pre-configured with some specific features), to define the curriculum, and to define the gamification model. To edit a course, teachers must perform the same actions of the *Configure tutor* use case. In addition, a teacher may also create different types of educational resources in the tutor. As such, teachers can reuse existing resources previously created by others teachers in the authoring solution as well as create content and problems. When creating problems, teachers may also create support (e.g., hint and discussion).

As our target users are teachers, it is of utmost importance to the success of the system to be simple and with high usability. However, at the same time, teachers should feel they are in the control with respect to the gamified ITS they are authoring. Hence, it is also necessary to provide a fair level of flexibility in the authoring process. In this way, the design of this authoring solution must also consider the trade-off between usability and flexibility non-functional requirements, as discussed by Dağ et al. [2014], Murray [2004], Sottilare [2015]. Moreover, interoperability is also an important non-functional requirement to consider in the design of this system since we are providing a generic authoring solution which might provide decisions that must be reasoned by third-party gamified ITSs.

Once use cases do not define the sequence of activities of the system flow, Figure 6.6 presents an activity diagram illustrating the execution of the two main authoring flows of the solution. The top "lane" presents the flow of use cases with respect to the customization of gamified ITS features. As shown in the figure, there are two alternative flows to customize features, one creating a tutor from scratch and the other applying a configuration template. The "lane" on the bottom shows the flow for authoring educational resources. Note that after creating or reusing a resource, a teacher can create others resources, but this is one of the activities that demand more time to create, as such, we enable teachers to stop creating resources or continuing creating them according to their needs.

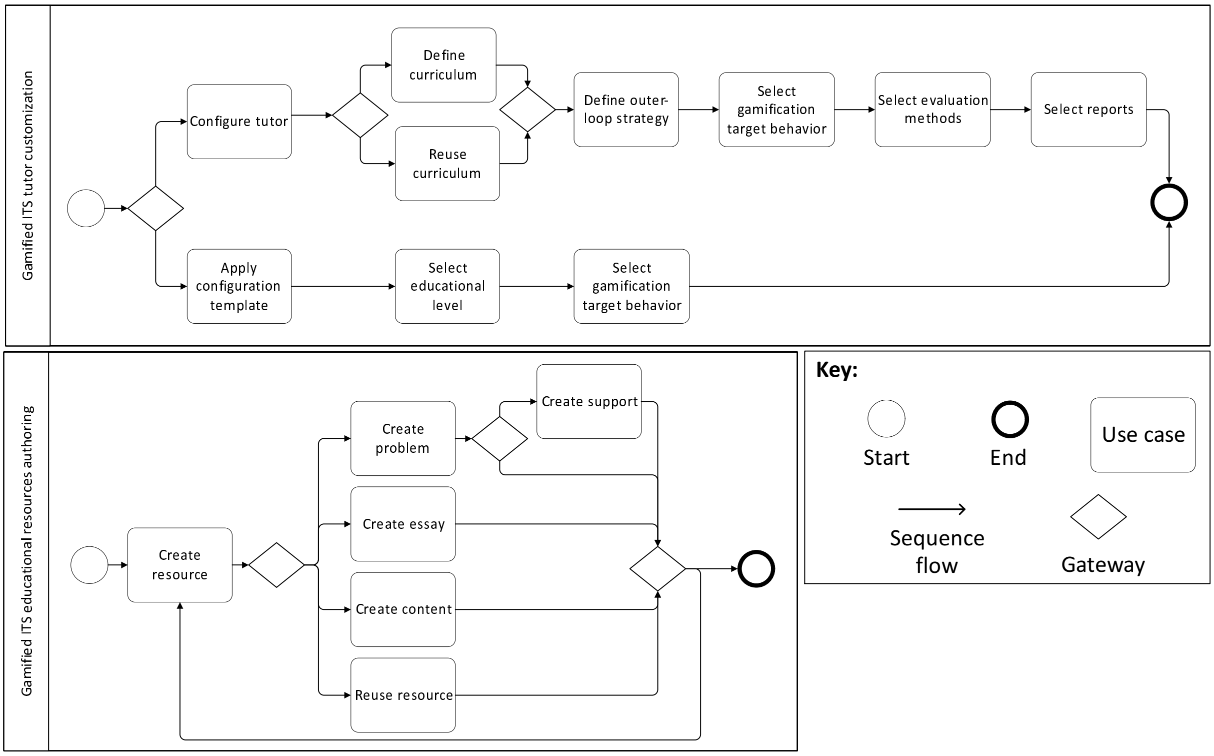
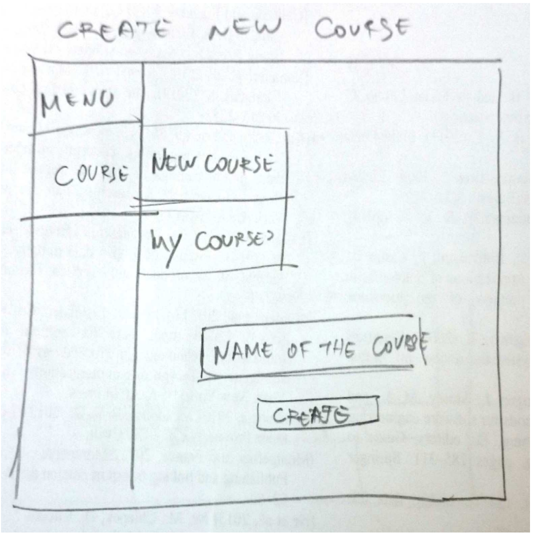


Figure 6.6: Authoring flow execution

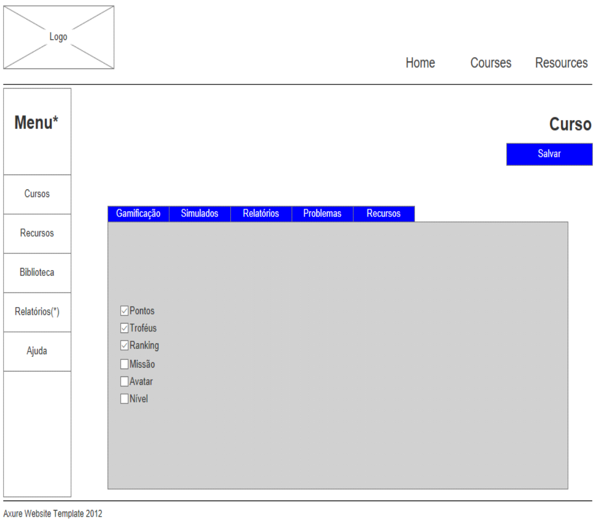
6.2.2 Authoring prototyping

Once users of this authoring solution are teachers, providing beautiful, usable and simple graphical interfaces is imperative. As such, we strongly rely on prototyping to increasingly improve the graphical interfaces of this solution.

Based on the functional and non-functional requirements identified, we defined several prototypes at different levels for supporting the development of the graphical interfaces of the authoring solution. First, we defined low level prototypes aiming to design the preliminary graphical interfaces and, then, medium-level prototypes (i.e., interactice) were developed with the aid of the Axure online software[1]. Figure 6.7 presents examples of two illustrations of both prototyping strategies.



(a) Low-level prototype for creating tutor



(b) Medium-level prototype for configuring a tutor

Figure 6.7: Example of prototypes specified for the authoring solution

Afterwards, considering the low and medium-level prototypes, we analyzed them with the aim of improving the graphical interfaces of the authoring solution for teachers. As such, we decided to redesign the authoring solution following the guidelines of the material design by google[2] since these guidelines are a standard way for developing web-based applications with great support of current web technologies (e.g., AngularJS[3]). In the redesign of

[1] `https://www.axure.com/`
[2] `https://material.io/guidelines/`
[3] `https://angularjs.org/`

graphical interfaces, we improved the prototypes with two features that we expect to aid teachers during the authoring decision-making. The first feature is the tunneling persuasive strategy [Fogg, 2002]. As discussed by Kraft et al. [2007], tunneling makes it easier to users to go through a process. The client enters the tunnel (i.e., starts the program) when they initiate the activity (i.e, customize a tutor in this case) attempt. By entering the tunnel they give away a certain level of self-determination in that information and activities are presented in a predetermined sequence. We also enforced the reuse features (e.g., apply template, reuse curriculum, and reuse educational resources) in order to decrease the effort required from teachers to author gamified ITS. Hence, the prototypes of the authoring solution were redesigned with these graphical interface capabilities. Figure 6.8 illustrates the first prototype on which teachers may choose if they want to configure a tutor from scratch or apply an existing template configuration in the system. The design elements of the tunneling strategy can be identified by the vertical line that guides the authoring process presenting the steps needed to follow.

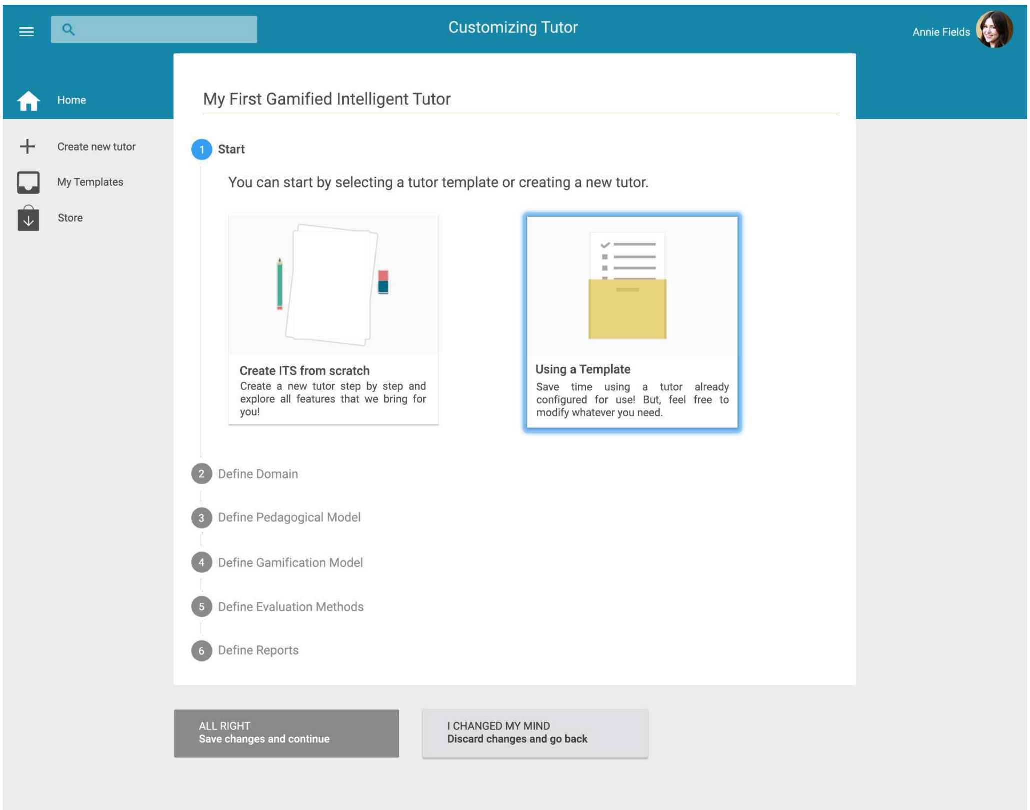


Figure 6.8: Prototype illustrating how the tunneling persuasive strategy and reuse capabilities are designed

Figure 6.9 shows the prototype for defining curriculum and subjects (domain model) of the flow for configuring a tutor from scratch, whereas, Figure 6.10 illustrates how a teacher may select a gamification target behavior during the authoring process.

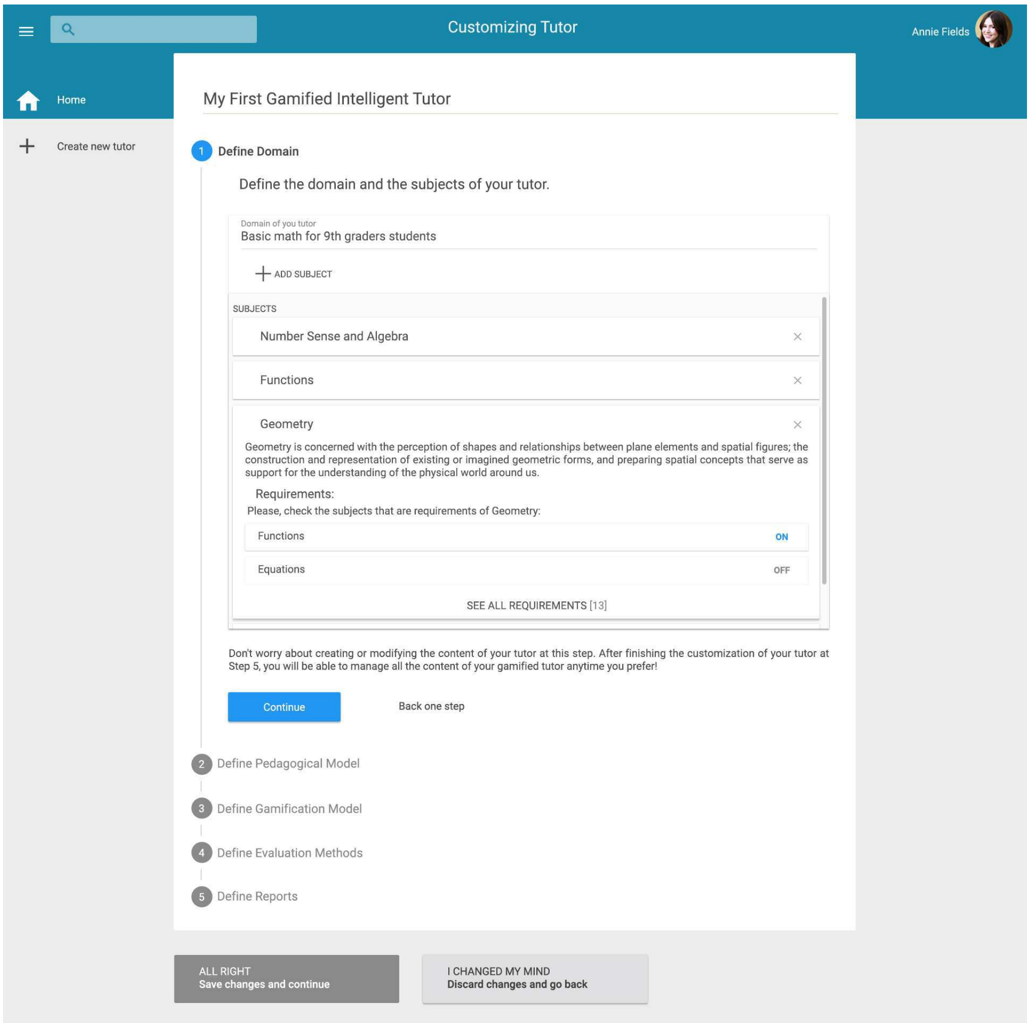


Figure 6.9: Prototype for defining curriculum and subjects

Moreover, Figures 6.11 and 6.12 illustrate two steps within application of a template to configure a tutor. The first figure shows the step on which teachers select the educational level of their tutor, whereas the second shows the last step on which teachers confirm the template application.

Moreover, Figure 6.13 shows two prototypes with respect to authoring of educational resources in the gamified tutor. The first prototype illustrates a teacher picking a tutor previously configured by him/her, and the second shows a teacher checking his/her previously created educational resources with options to create more resources of different

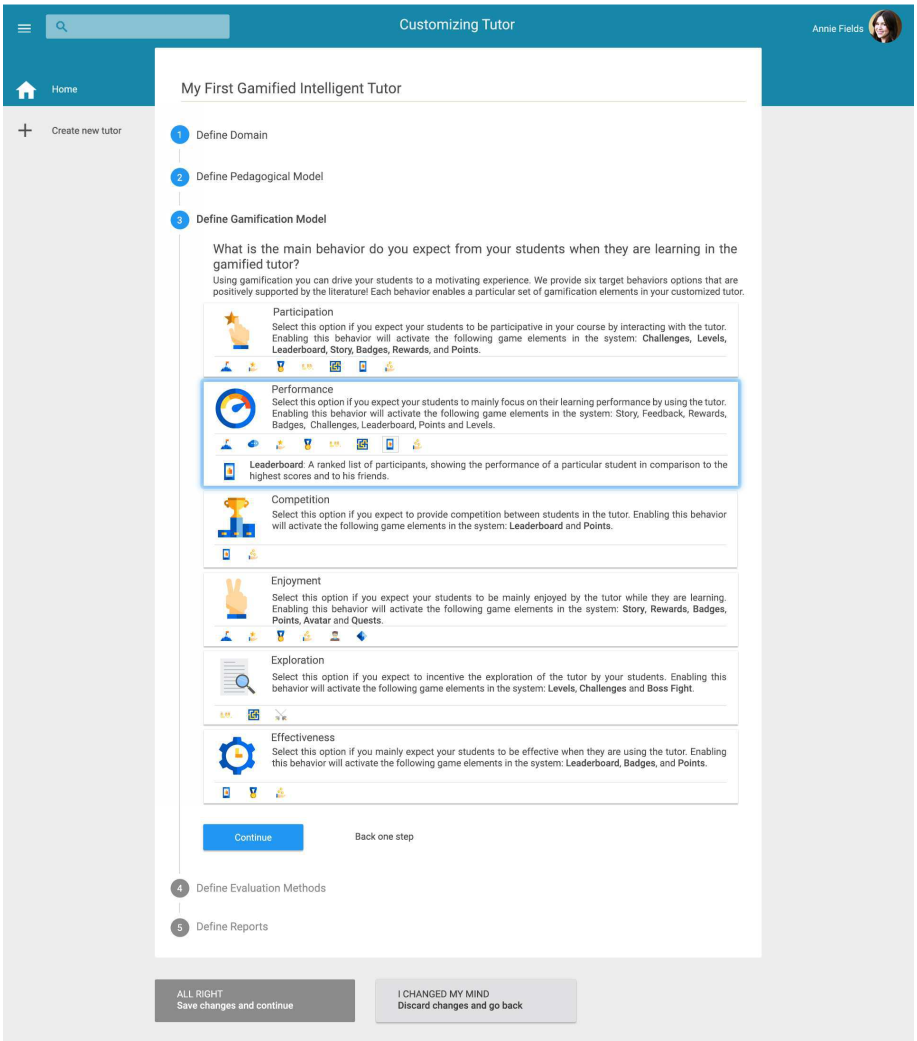


Figure 6.10: Prototype for selecting a gamification target behavior

types.

6.2.3 Architectural design and implementation

In this section we present the architectural models produced in the architectural design activity of the authoring computational solution development. First, we describe the architecture modules view along with the main design decisions that we made and, then, we describe the behavior of the architecture to explain how the authoring solution configure a new tutor and aids the creation of educational resources.

The modules view of our authoring computation solution architecture describes at a high level the main modules of the software and also illustrates how these modules are

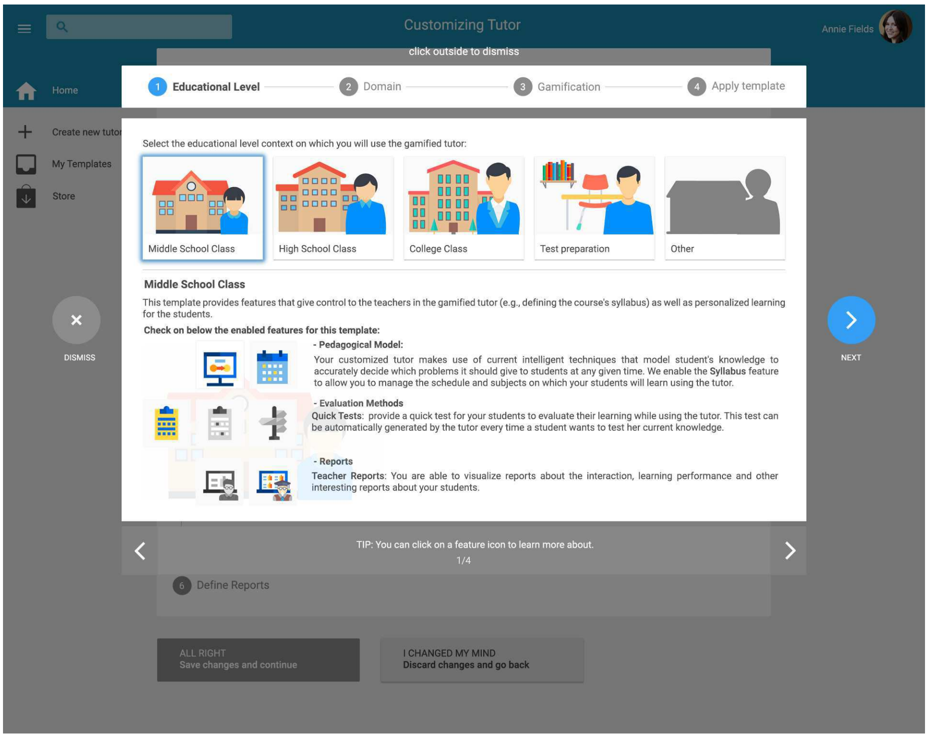


Figure 6.11: Prototype for selecting an educational level

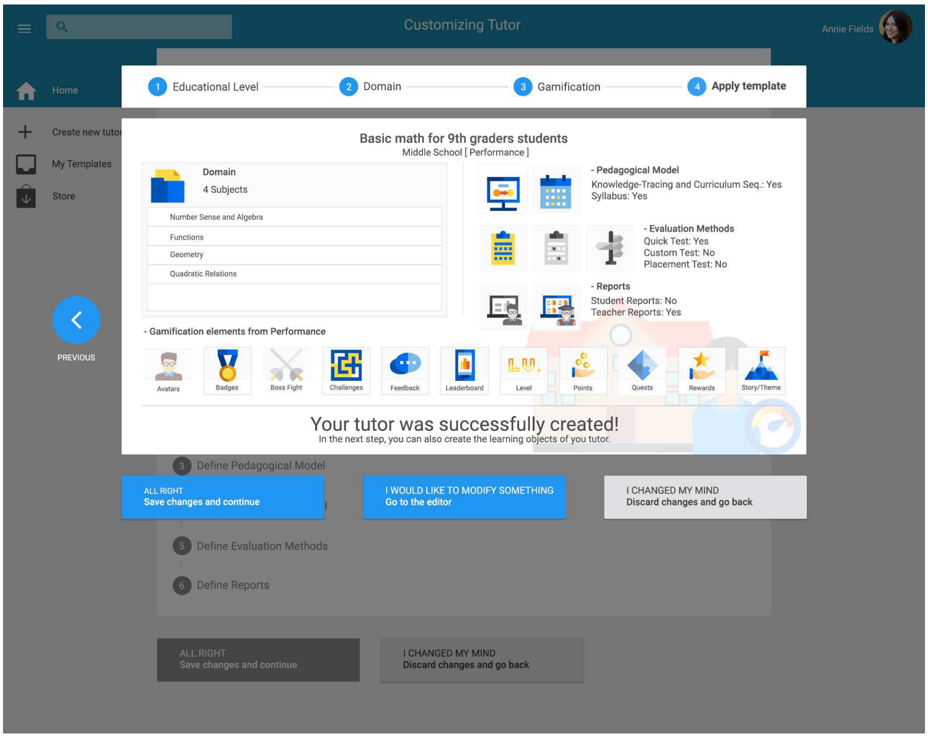
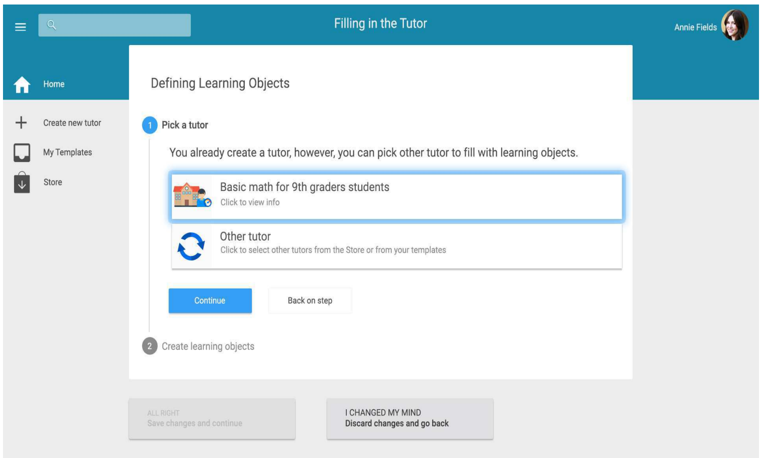
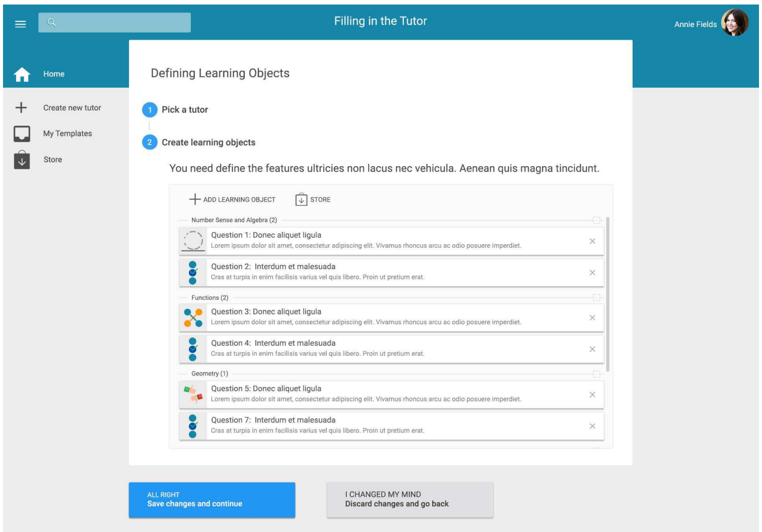


Figure 6.12: Prototype of the last step to apply a template



(a) Prototype for picking a tutor previously configured



(b) Prototype for creating educational resources

Figure 6.13: Prototypes for authoring educational resources

interconnected. This diagram contains the main architectural decisions that were made in order to satisfy the functional (see Figure 6.5) and non-functional requirements (i.e., usability and interoperability) identified in the previous activity. These decisions are: (i) use of the layer architectural style to manage the complexity of the system and to separate the concerns involved in the authoring process; (ii) use of the client/server style since we are building a web-based authoring system; (iii) use of rest services to enable the interoperability of frontend and backend components; (iv) use of the GITS-PL ontology to manage tutors configuration in order to enable interoperability with third-party gamified ITSs; (v) use of the GaTO ontology to represent the resources created in the tutor along with the decision related to the gamification target behavior; (vi) use of the Java Ontology Integrated Toolkit

[Holanda et al., 2013] to manage the persistence with the ontologies, and hence, Java as the backend programming language; (vii) use of the Spring framework[4] to support the Model-View-Controller architectural style; and (viii) use of AngularJS[5] as the frontend programming language since it provides built-in components for implementing the design guidelines we are using (i.e., material design by google).

As shown in Figure 6.14, the architecture contains four main layers: *Frontend*, *Backend*, and *Persistence*. The *Frontend* layer contains the views, controllers and services used to develop the graphical user interfaces of the authoring system. This layer is located in the client side of the architecture whereas the other two layers are located on the server side. Rest services intermediates the access to the Spring services provided by the *Backend* layer. These services make use of the *Ontology Management*, which deals with the knowledge access objects (KAOs) that are used to access the GITS-PL and GaTO ontologies (which are in *Persistence* layer). They also use the *Database Management* component to manage the data access objects (DAOs) related to the management of users in the database (*Persistence* layer).

Figure 6.15 presents an UML sequence diagram illustrating the behavior of the architecture to configure a new tutor receiving as input the choices made by a teacher. As seen in the figure, the configuration process may start when a teacher save a configuration a view of the *Frontend* layer, then the view calls an operation of the controller, which invokes itself the configuration services in same layer. Next, by using rest services, the front-end services call the Spring services, which invoke the *OntologyMgr* component. The *OntologyMgr* component makes use of operations from the KAO to update the gamified tutor instance owl file that represents which features are activated or deactivated according to the configuration saved. This component also updates the GaTO ontology to set the chosen gamification target behavior as well as the curriculum (and subjects) defined by a teacher.

To create an educational resource by using the authoring solution, the architecture works in a similar way to the configuration of tutor. Figure 6.16 shows the sequence diagram to create a generic problem, which can be of four types (e.g., multiple-choice and fill blanks). After receiving the data regarding the problem and the type of problem been created, the

[4]`https://projects.spring.io/spring-framework/`

[5]`https://angularjs.org/`

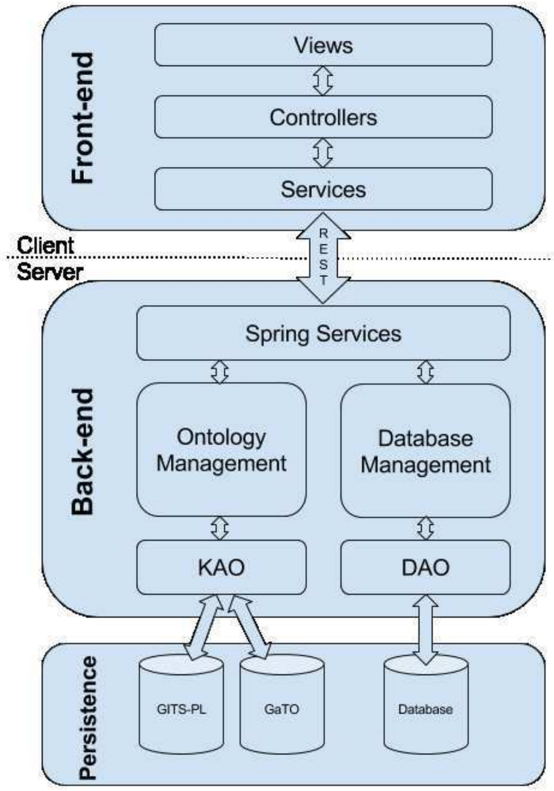


Figure 6.14: Authoring computation solution modules view

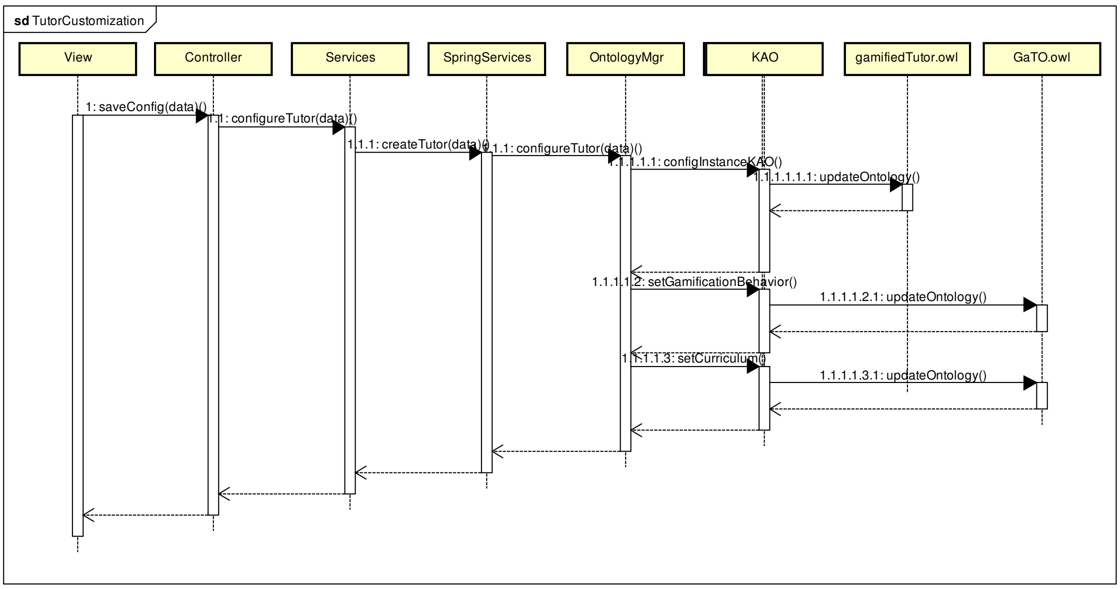


Figure 6.15: Sequence diagram showing the behavior of the architecture to configure a new tutor

OntologyMgr updates the GaTO ontology to create a new OWL individual representing this problem. This component also updates the gamified tutor instance owl file to activate the feature regarding the type of problem created according to the GITS-PL feature model.

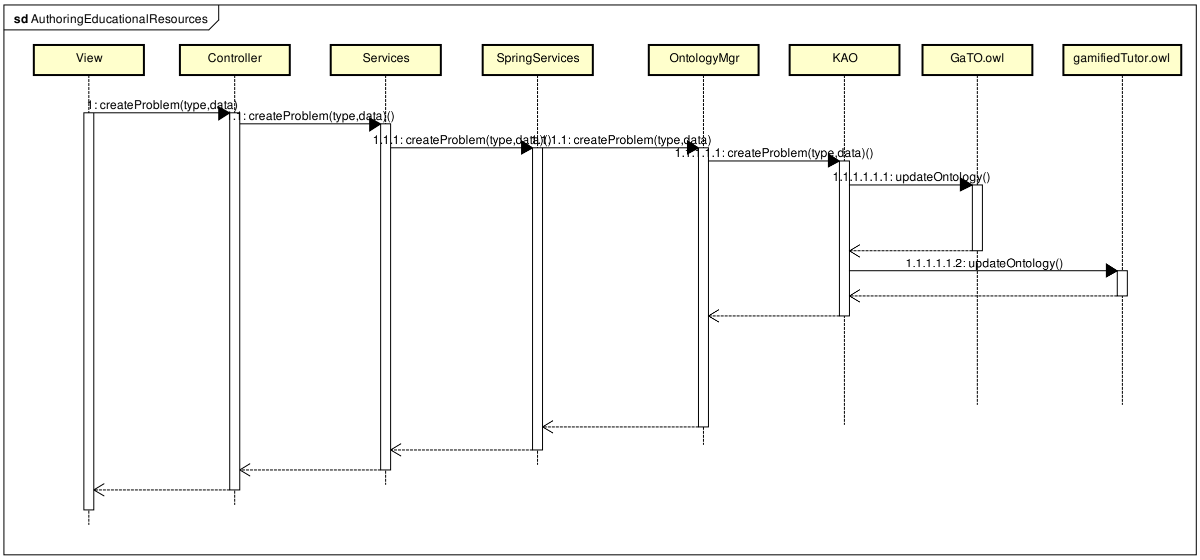


Figure 6.16: Sequence diagram showing the behavior of the architecture to create a new problem

After eliciting, analyzing, and representing the requirements and prototypes as well as designing the authoring computation solution presented in this work, we have implemented it using the aforementioned technologies. This implementation operationalizes an authoring solution that take advantage of the ontology that supports the management of gamified ITS variability and the ontology that connects gamification theories, framework and design practices to ITS concepts.

In the following sections, we describe how we empirically evaluated this authoring computational solution in a twofold way: (i) in laboratory settings with graduate students (Section 6.3); and (ii) with real teachers (Section 6.4).

6.3 Experiment #1: laboratory settings

As previously explained (Section 3.4), we could not find any related work enabling authoring or customization of gamified ITS features. As such, we don't have a basis for comparison with our proposal. Hence, in this experiment we compare different versions of our prototypes to gather feedback from users aiming to improve the designed prototypes.

Thus, this first experiment intends to analyze the designed prototypes of the authoring solution that combine the use of template and gamification authoring by selecting target behaviors evaluate them with respect to several metrics such as perceived ease of use, perceived usability, complexity, aesthetics, novelty, unity, intensity, attitude towards use, perceived system support, and credibility from the viewpoint of teachers in the context of graduate students and researches, from two research groups in Brazil and Canada, analyzing the prototypes and answering a survey.

In the following sections, we describe the materials and method used in this experiment, the procedure and participants, the results, analysis and discussion of the results as well as threats to the validity of our results.

6.3.1 Materials and methods

In this section, we describe the variables, experimental design and research hypotheses investigated in this experiment.

Variables

The independent variables of this experiment are defined as follows and the factor levels are summarized in Table 6.1.

- Gamified ITS configuration flow: this variable refers to the two alternative flows to customize a gamified tutor in the authoring solution, i.e., configuring from scratch or using a template.
- Gamification model authoring: this variable refers to the way teachers may select the game design elements to be included in the gamified ITS. As such, our authoring solution provides gamification authoring where teachers select a target behavior that is related to set of game design elements. In order to investigate the perception of this feature in comparison to gamification authoring by selecting individual game design elements, we include an alternative way (control variable) on which teachers may analyze prototypes for selecting game design elements individually.

Table 6.1: Factors levels

Factor	Levels
Gamified ITS configuration flow	Scratch – Configure tutor from scratch Template – Customize using template
Gamification model authoring	Individually – Activate game design elements individually Behavior – Select gamification target behavior

The effects (dependent variables) of the factors are overall analyzed with respect to several constructs investigated by some studies [Cho et al., 2009, Holden and Rada, 2011, Teo, 2011] that applied the technology acceptance model (TAM) method [Venkatesh and Davis, 2000] with teachers and/or users in the context of e-learning. We also rely on a study that presents constructs related to aesthetics which might be important to analyze our prototypes [Jiang et al., 2016]. These metrics are described on below.

- Perceived ease of use (PEU): This construct has to do with the extent to which a person thinks that using a system will be relatively free of effort [Holden and Rada, 2011, Teo, 2011];
- Perceived usability (PU): This construct is described as a system's capability to be used by humans effectively and easily [Holden and Rada, 2011, Shackel, 1991]. It includes five others sub-constructs: Understandability (U), Flexibility (F), Functionality (FU), Navigation (N) and Memorability (M). Understandability refers to the degree of users' perceived understanding of a given technology. Flexibility measures the degree of users' perceived flexibility of a given technology. Functionality refers to the satisfaction of the system's incorporated features. Navigation refers to the ease of operating the system intuitively and memorability refers to the ease of remembering how to use the system;
- Complexity (C): According to Jiang et al. [2016], the worth of an artwork depends on the number of different but interrelated components of the work – that is, complexity;
- Aesthetics (A): This construct refers to concepts and ideas that encompass the

orderliness and clarity of a design as well as to users' perceptions of the novelty and creativity of a website's design [Jiang et al., 2016];

- Unity (UT): The combination of components in a design must then be coherently connected together to create a sense of completeness – that is, unity [Jiang et al., 2016];
- Intensity (I): A good aesthetic object must have some marked quality – that is, intensity [Jiang et al., 2016];
- Novelty (NO): Novelty is the quality or state of being new and unusual, different from anything in prior existence. In the context of website design, it is manifested via the use of a new display menu style, the adoption of a new background or layout, the presentation of a customized interface, and so on [Jiang et al., 2016];
- Attitude towards use (ATU): This construct refers to attitude of users to be favorable in using a technology [Jiang et al., 2016];
- Perceived system support (PSS): Cho et al. [2009] define this construct as the perceived effectiveness of system support for a system;
- Credibility (CR): This variable captures the overall credibility of a prototype based on users' perceptions.

Experimental design

We used a 2x2 between-subjects design; participants were shown only one of the four possible versions of the prototypes. Each version presents graphical elements that combine the factor levels (see Table 6.1). Figure 6.17 shows the flow of the prototypes presented to participants per each version considered in the experimental design. In Version 1 participants analyze the prototypes for configuring a tutor by using a template and for authoring gamification by selecting a gamification target behavior. In Version 2, participants analyze the prototypes for configuring a tutor from scratch and authoring gamification by selecting a behavior, as well. Version 3 only differs from Version 1 when presenting the prototype for authoring gamification; in this version participants evaluate the way that a teacher selects game design elements individually. Similarly, Version 4 only differs from Version 2 in the way to author gamification, which is made by selecting game design elements individually. Note that the version on which a participant evaluates the prototypes is randomly allocated to him.

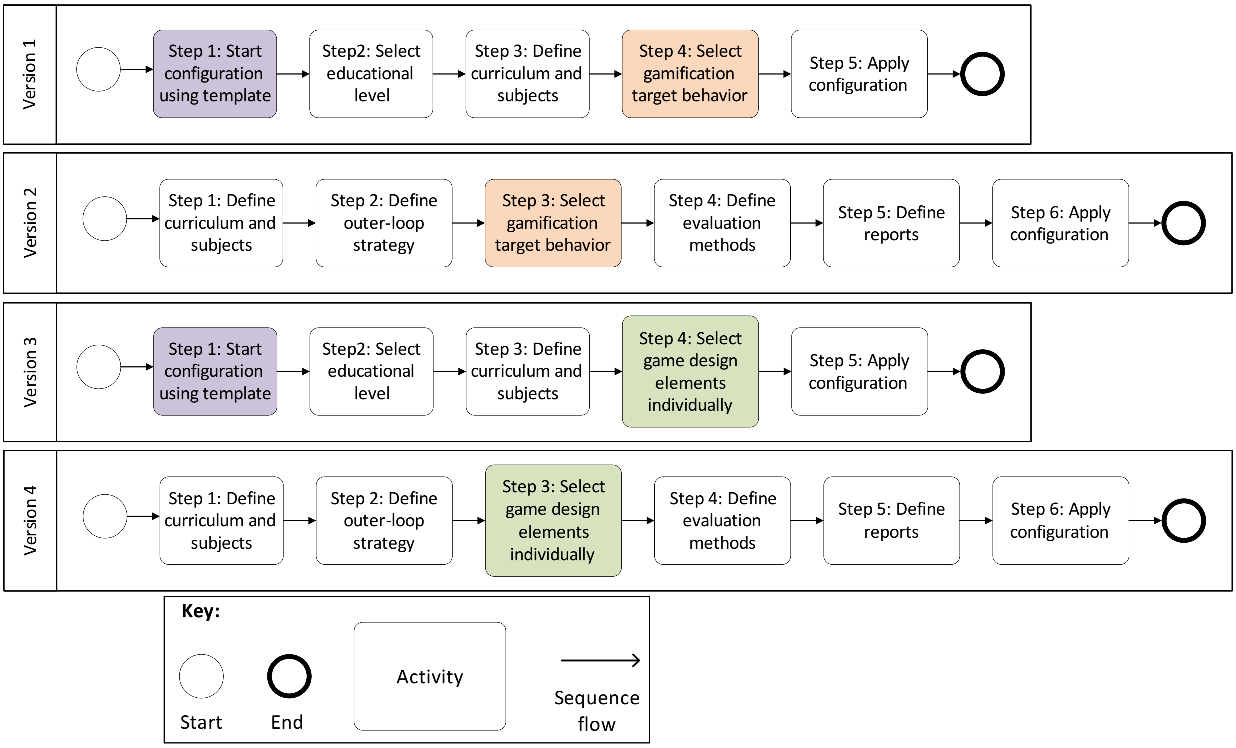


Figure 6.17: Trials definition illustrating the flow of steps for each version. Steps with same colors are highlighted to identify that flows are using the same treatment

Most the questions presented in Table 6.2 are answered after participants proceed through all prototypes steps, in the end of experiment. However, there are some questions (i.e., Understandability (U), C3, PSS1, PSS2, and PSS3) that are answered by participants to individually assess the effect of particular steps. Our choice for collecting these metrics per step was done because it can enable the individual analysis of steps against each other or to analyze steps that are common to more than one version. For example, Versions 1 and 2 present similar prototypes for selecting a gamification target behavior to author gamification. By contrast, versions 3 and 4 present similar prototypes for selecting game design elements individually to author gamification. In this way, we can jointly compare the metrics collected from versions 1 and 2 to the metrics collected from versions 3 and 4 to investigate the individual impact of this step regarding these constructs.

Table 6.2: Questions used to measure the constructs. The score of each construct is computed by the average of the questions using a Likert scale from 1 (completely disagree) to 7 (completely agree). Credibility is measured in a scale from 1 to 9.

Questionnaire
Perceived ease of use (PEU)
PEU1: Learning to use this system seems to be easy for me
PEU2: I think that would be easy to use this system to do what is needed to do
PEU3: The interaction with this system does not seem to require much effort
PEU4: I think that would be easy for me to become skilful at using this system
PEU5: I think that this system would be easy to use
Perceived usability (PU)
Understandability (U): This step seems to be clear and understandable
Flexibility (F): I think that this system would be flexible to interact with
Functionality (FU): The system seems to have good functionality (features)
Navigation (N): I feel that I would have an intuitive sense on how to operate the system
Memorability (M): I feel that it would be easy to remember how to perform tasks using the system
Complexity (C)
C1: This system seems to be very complex to use
C2: The extent to which the system employs diverse components and design styles seems to be very well designed
C3: The degree of information load on this step seems to be very well designed
Aesthetics (A)
A1: The system seems to be aesthetically appealing
A2: The system seems to be atractive
A3: The system seems to be beautiful
A4: The system seems to be lovely
A5: The system has a pleasant look and feel
Unity (UT)
UT1: The system design seems to be cohesive
UT2: The system design seems to be consistent
UT3: The system design seems to be harmonious
Intensity (I)
I1: The contrast of the graphics seems to be very well designed
I2: The intensity of the look and feel seems to be very well designed
I3: The brightness of how the system looks seems to be very well designed
Novelty (NO)
NO1: The system design seems to be original
NO2: The system design seems to be unique
Attitude towards use (ATU)
ATU1: Overall, the system seems to be good
ATU2: Overall, I have formed a favorable impression toward the system
ATU3: Overall, I have positive feelings about this system
Perceived system support (PSS)
PSS1: Step quality including help function and instructional support is good
PSS2: The step seems to provide personalized support (e.g., there are options which enable you to specify your preferences)
PSS3: Step's support for completing the task seems to be satisfactory
Credibility (CR): In general, what is the credibility of the authoring tool?

Note that, to compute the overall score of the metrics for the versions, we must calculate the average of the values per each metric and step, and calculate this average again with the overall metric. For instance, to compute the overall complexity for a version, first the average of the C3 construct must be computed separately and then must be put together with C1 and C2 metrics (which are answered in the end of the experiment) to calculate a new average that would represent the overall complexity score for one participant.

Research hypotheses

Based on the variables previously described, the research hypotheses presented in Table 6.3 are investigated in this experiment.

Furthermore, in Table 6.4, these research hypotheses are formally presented. As presented in Table 6.2, PEU, PU, C, A, UT, I, NO, ATU, PSS, and CR are functions that return, respectively, the value of perceived ease of use, perceived usability, complexity, aesthetics, unity, intensity, novelty, attitude towards use, perceived system support, and credibility on the versions V1 (template and behavior), V2 (scratch and behavior), V3 (template and individual selection of game design elements), and V3 (scratch and individual selection of game design elements). The functions U and C_3 return the value of understandability and complexity (with respect to the degree of information load), respectively, on the jointly responses for the prototype that a teacher selects a gamification target behavior (V1-Step4 and V2-Step3) in comparison to the responses for the prototype on which a teacher selects game design elements individually (V3-Step4 and V4-Step3).

6.3.2 Procedure and participants

This section describes how the experiment was executed. It depicts who the participants are (and how they were selected), which instruments were used and how the experiment was performed.

Participant Selection

The experiment involves the participation of human agents. Participants were researchers (i.e., undergraduate and graduate students as well as professors) from two research groups:

Table 6.3: Hypotheses of the first experiment

H1-0: The perceived ease of use (PEU) of the versions is equal
H1-1: The perceived ease of use (PEU) of the versions is different
H2-0: The perceived usability of the versions is equal
H2-1: The perceived usability of the versions is different
H3-0: The complexity of the versions is equal
H3-1: The complexity of the versions is different
H4-0: The aesthetics of the versions is equal
H4-1: The aesthetics of the versions is different
H5-0: The unity of the versions is equal
H5-1: The unity of the versions is different
H6-0: The intensity of the versions is equal
H6-1: The intensity of the versions is different
H7-0: The novelty of the versions is equal
H7-1: The novelty of the versions is different
H8-0: The attitude towards use of the versions is equal
H8-1: The attitude towards use of the versions is different
H9-0: The perceived system support of the versions is equal
H9-1: The perceived system support of the versions is different
H10-0: The credibility of the versions is equal
H10-1: The credibility of the versions is different
H11-0: The understandability for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is equal
H11-1: The understandability for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is different
H12-0: The degree of information load (C3) for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is equal
H12-1: The degree of information load (C3) for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is different
H13-0: The perceived system support (PSS) for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is equal
H13-1: The perceived system support (PSS) for authoring gamification by selecting a target behavior (versions 1 and 2) and by selecting game design elements individually (versions 3 and 4) is different

NEES[6] from the Federal University of Alagoas in Brazil and MADMUC[7] from the University of Saskatchewan in Canada. Participants from both research groups were invited by mailing lists.

[6] http://nees.com.br/en/
[7] http://madmuc.usask.ca/

Table 6.4: Formal definition of the research hypotheses

Hypothesis	Null Hypothesis	Alternative Hypothesis
H1	*H1-0* : $\mu_{PEU}(V1) = \mu_{PEU}(V2) = \mu_{PEU}(V3) = \mu_{PEU}(V4)$	*H1-1* : $\mu_{PEU}(V1) \neq \mu_{PEU}(V2) \neq \mu_{PEU}(V3) \neq \mu_{PEU}(V4)$
H2	*H2-0* : $\mu_{PU}(V1) = \mu_{PU}(V2) = \mu_{PU}(V3) = \mu_{PU}(V4)$	*H2-1* : $\mu_{PU}(V1) \neq \mu_{PU}(V2) \neq \mu_{PU}(V3) \neq \mu_{PU}(V4)$
H3	*H3-0* : $\mu_{C}(V1) = \mu_{C}(V2) = \mu_{C}(V3) = \mu_{C}(V4)$	*H3-1* : $\mu_{C}(V1) \neq \mu_{C}(V2) \neq \mu_{C}(V3) \neq \mu_{C}(V4)$
H4	*H4-0* : $\mu_{A}(V1) = \mu_{A}(V2) = \mu_{A}(V3) = \mu_{A}(V4)$	*H4-1* : $\mu_{A}(V1) \neq \mu_{A}(V2) \neq \mu_{A}(V3) \neq \mu_{A}(V4)$
H5	*H5-0* : $\mu_{UT}(V1) = \mu_{UT}(V2) = \mu_{UT}(V3) = \mu_{UT}(V4)$	*H5-1* : $\mu_{UT}(V1) \neq \mu_{UT}(V2) \neq \mu_{UT}(V3) \neq \mu_{UT}(V4)$
H6	*H6-0* : $\mu_{I}(V1) = \mu_{I}(V2) = \mu_{I}(V3) = \mu_{I}(V4)$	*H6-1* : $\mu_{I}(V1) \neq \mu_{I}(V2) \neq \mu_{I}(V3) \neq \mu_{I}(V4)$
H7	*H7-0* : $\mu_{NO}(V1) = \mu_{NO}(V2) = \mu_{NO}(V3) = \mu_{NO}(V4)$	*H7-1* : $\mu_{NO}(V1) \neq \mu_{NO}(V2) \neq \mu_{NO}(V3) \neq \mu_{NO}(V4)$
H8	*H8-0* : $\mu_{ATU}(V1) = \mu_{ATU}(V2) = \mu_{ATU}(V3) = \mu_{ATU}(V4)$	*H8-1* : $\mu_{ATU}(V1) \neq \mu_{ATU}(V2) \neq \mu_{ATU}(V3) \neq \mu_{ATU}(V4)$
H9	*H9-0* : $\mu_{PSS}(V1) = \mu_{PSS}(V2) = \mu_{PSS}(V3) = \mu_{PSS}(V4)$	*H9-1* : $\mu_{PSS}(V1) \neq \mu_{PSS}(V2) \neq \mu_{PSS}(V3) \neq \mu_{PSS}(V4)$
H10	*H10-0* : $\mu_{CR}(V1) = \mu_{CR}(V2) = \mu_{CR}(V3) = \mu_{CR}(V4)$	*H10-1* : $\mu_{CR}(V1) \neq \mu_{CR}(V2) \neq \mu_{CR}(V3) \neq \mu_{CR}(V4)$
H11	*H11-0* : $(\mu_U(V1_{S4}) + \mu_U(V2_{S3})) = (\mu_U(V3_{S4}) + \mu_U(V4_{S3}))$	*H11-1* : $(\mu_U(V1S4) + \mu_U(V2_{S3})) \neq (\mu_U(V3_{S4}) + \mu_U(V4_{S3}))$
H12	*H12-0* : $(\mu_{C_3}(V1_{S4}) + \mu_{C_3}(V2_{S3})) = (\mu_{C_3}(V3_{S4}) + \mu_{C_3}(V4_{S3}))$	*H12-1* : $(\mu_{C_3}(V1_{S4}) + \mu_{C_3}(V2_{S3})) \neq (\mu_{C_3}(V3_{S4}) + \mu_{C_3}(V4_{S3}))$
H13	*H13-0* : $(\mu_{PSS}(V1_{S4}) + \mu_{PSS}(V2_{S3})) = (\mu_{PSS}(V3_{S4}) + \mu_{PSS}(V4_{S3}))$	*H13-1* : $(\mu_{PSS}(V1_{S4}) + \mu_{PSS}(V2_{S3})) \neq (\mu_{PSS}(V3_{S4}) + \mu_{PSS}(V4_{S3}))$

Preparation and Instrumentation

The data collection was performed through the use of a survey (using a likert scale from 1 to 7) that includes our experimental design. After a participant agrees with the terms and answers demographic questions, he receives one of the four versions that are randomly allocated to him. For each step of the version, the participant answers some questions (U, C3, PSS1, PSS2, and PSS3) regarding this step and in the end he answers all the other questions with respect to the version in overall. The survey is available at `https://fluidsurveys.usask.ca/s/agits-survey/`. Note that, in this experiment, the prototypes are not interactive, participants just analyze the images containing the design related to steps and answer the questions.

6.3.3 Results

This section presents the analysis of the data collected in this experiment. The collected data, as well as the scripts used in the experimental analysis are available at `https://goo.gl/mtukuL`.

Before presenting the descriptive and inferential statistic results of this experiment, we depict the demographic statistics for the participants of this study (Table 6.5). As seen in the table, participants provided information about their gender, age, occupation, education level, and country. In the following section, we present the descriptive statistics for the results of this experiment.

Table 6.5: Participant demographics

Demographics	Version 1	Version 2	Version 3	Version 4
Size (n)	15	13	16	15
Gender				
Female	2 (13.33%)	5 (38.46%)	4 (25%)	6 (40%)
Male	13 (86.67%)	8 (61.54%)	12 (75%)	9 (60%)
Rather not say	0%	0%	0%	0%
Age				
16–25	3 (20%)	0%	6 (37.5%)	4 (26.66%)
26–40	10 (66.66%)	9 (69.23%)	8 (50%)	10 (66.66%)
41–65	2 (13.33%)	4 (30.76%)	2 (12.5%)	1 (6.66%)
Over 65	0%	0%	0%	0%
Rather not say	0%	0%	0%	0%
Occupation				
Student	7 (46.67%)	9 (69.23%)	9 (56.25%)	9 (60%)
Teacher	6 (40%)	3 (23.07%)	6 (37.5%)	4 (26.66%)
Other	2 (13.33%)	1 (7.69%)	1 (6.25%)	2 (13.33%)
Education level				
Junior High/Middle School	0%	0%	0%	0%
High School	0%	0%	1 (6.25%)	2 (13.33%)
Technical/trade school	1 (6.66%)	0%	1 (6.25%)	0%
Bachelor’s degree	3 (20%)	2 (15.38%)	2 (12.5%)	4 (26.66%)
Master’s degree	9 (60%)	9 (69.23%)	10 (62.5%)	6 (40%)
Doctorate degree	2 (13.33%)	2 (15.38%)	2 (12.5%)	3 (20%)
Other	0%	0%	0%	0%
Country				
Brazil	12 (80%)	8 (61.53%)	14 (87.5%)	13 (86.66%)
Canada	2 (13.33%)	1 (7.69%)	0%	1 (6.66%)
Ecuador	1 (6.67%)	0%	0%	0%
India	0%	0%	1 (6.25%)	0%
Iran	0%	1 (7.69%)	1 (6.25%)	0%
Nigeria	0%	1 (7.69%)	0%	1 (6.66%)
United States	0%	1 (7.69%)	0%	0%
Venezuela	0%	1 (7.69%)	0%	0%

Descriptive statistics and assumptions verification

The collected data contains the participants’ answers to the questions shown in Table 6.2 for each answer regarding the dependent variable. Note that, except for the Credibility (CR) which can receive a value from 1 to 9, all the other dependent variables are measured by the average of answers regarding each variable using a likert scale (from 1 to 7). Thus, to analyze these results, we first conduct a descriptive analysis of the data, by analyzing histograms and boxplots of the computed metrics.

In Table 6.6, we present the summary of statistics (e.g., median, mean, sd) and the results of the normality testes (e.g., shapiro-wilk and anderson-darling tests) we applied for the perceived ease of use (PEU), perceived usability (U), complexity (C), aesthetics (A), unity (U), intensity (I), novelty (NO), attitude towards use (ATU), perceived system support (PSS), and credibility (CR) metrics per each version analyzed in this experiment. We also present in Figure 6.18 the boxplots for the ten metrics comparing the four versions analyzed in this experiment.

In addition, in Table 6.7 we present the summary of statistics for the understandability (U), complexity3 (C3), and perceived system support (PSS) metrics with respect to the two alternative prototypes for authoring gamification: selecting a target behavior or selecting particular game design elements. We also present the boxplots, in Figure 6.19, of the jointly comparison between the versions of these two alternative prototypes.

Inferential statistics

As previously presented, we are investigating ten hypotheses to analyze the impact of four different versions of our prototypes with respect to ten constructs. We also investigate three hypotheses to verify the participants' perceptions with respect to the two alternative ways for authoring gamification.

To verify the hypotheses, statistical tests were applied for each one of the hypotheses formalized in Table 6.4. The hypotheses (H1 to H10) includes the comparison between four versions, hence, we apply hypotheses tests for factorial analysis (i.e., more than two-groups comparison). For the hypotheses H11 to H13, we apply two-group hypothesis tests. In order to decide which tests to apply, we first verified the normality of the data regarding the hypotheses (see the results in Table 6.6 and 6.7). Afterwards, for the normal distributions (i.e., when all factor levels are normal) we applied a parametric test (one-way anova for comparisons between more than two levels and t-test for two-group comparisons), whereas, for the non normal distributions, we applied a non-parametric test (Kruskal-Wallis for more than two-groups comparisons and Wilcox test for two-groups comparisons).

Table 6.8 presents the results of applying the tests for our hypotheses. We depict the hypotheses, the applied test, the p-value and the decision if the resultant p-value is enough

Table 6.6: Summary of statistics and normality tests for the ten metrics evaluated per version

	PEU	PU	C	A	UT	I	NO	ATU	PSS	CR
Version 1 (N=15)										
Min	4.8	5.2	5	5.2	4.6667	5	3	5.6667	5	7
Max	7	7	6.3333	7	7	7	7	7	7	9
Range	2.2	1.8	1.3333	1.8	2.3333	2	4	1.3333	2	2
Median	6	5.92	5.9333	6.2	6.3333	6	6	6.3333	5.9333	8
Mean	6.08	6.0853	5.8044	6.2	6.1111	5.9778	5.6	6.4	5.9167	8.2
St d. Dev.	0.627	0.4882	0.4004	0.5806	0.5296	0.6482	1.168	0.4748	0.5238	0.5606
Shap. Wilk (p-value)	0.6536	0.2038	0.3397	0.3206	0.036	0.2353	0.2131	0.0923	0.9108	6.00E-04
Anderson-Darling (p-value)	0.5992	0.1062	0.3607	0.377	0.0233	0.2953	0.3008	0.1372	0.6847	0
Normal?	1	1	1	1	0	1	1	1	1	0
Version 2 (N=13)										
Min	2.8	3.3	3.6667	2.8	3	4	2	2.6667	2.6778	3
Max	6.6	6.6333	6.3333	7	7	6.3333	6.5	6.6667	6.4778	9
Range	3.8	3.3333	2.6667	4.2	4	2.3333	4.5	4	3.8	6
Median	6	5.9	5.5	5.8	5.6667	5.6667	5.5	6	5.2	7
Mean	5.4923	5.4385	5.2308	5.3231	5.5385	5.5128	5	5.5128	5.0803	6.9231
St d. Dev.	1.1449	1.0259	0.9032	1.1417	1.005	0.728	1.4434	1.2518	0.9711	1.8913
Shap. Wilk (p-value)	0.0064	0.0045	0.0204	0.3711	0.1562	0.1888	0.0657	0.0011	0.0781	0.0027
Anderson-Darling (p-value)	0.0031	0.0021	0.0132	0.2726	0.1563	0.2144	0.0616	3.00E-04	0.0516	9.00E-04
Normal?	0	0	0	1	1	1	1	0	1	0
Version 3 (N=16)										
Min	5	4.84	4.4667	4.2	5.3333	4.3333	1	5.6667	4.35	5
Max	6.8	6.76	6.3333	6.4	6.3333	6.3333	6.5	7	7	9
Range	1.8	1.92	1.8667	2.2	1	2	5.5	1.3333	2.65	4
Median	6	5.98	5.8	5.5	6	5.6667	5	6	5.5167	8
Mean	6.05	5.96	5.6417	5.525	5.9792	5.6042	4.5625	6.1875	5.5312	7.625
St d. Dev.	0.5086	0.478	0.4856	0.6445	0.3096	0.5607	1.5152	0.3645	0.6246	0.9574
Shap. Wilk (p-value)	0.3122	0.774	0.0598	0.6348	0.0277	0.2806	0.0725	0.0011	0.6739	0.0064
Anderson-Darling (p-value)	0.2469	0.7045	0.0313	0.7827	0.0183	0.3427	0.0583	0	0.4896	0.0025
Normal?	1	1	1	1	0	1	1	0	1	0
Version 4 (N=15)										
Min	4.4	3.9667	4.3889	3.6	4	4	2	4.3333	3.4222	6
Max	7	7	6.9444	7	7	7	7	7	6.5444	9
Range	2.6	3.0333	2.5556	3.4	3	3	5	2.6667	3.1222	3
Median	6	5.9333	5.7778	6	6	6	6	6	5.9333	8
Mean	6.0933	5.9689	5.7148	6	6.0667	5.9778	5.3	6.1111	5.7289	7.8667
St d. Dev.	0.7959	0.9102	0.708	0.8718	0.7787	0.8495	1.347	0.7732	0.7971	1.1255
Shap. Wilk (p-value)	0.0373	0.133	0.9747	0.0314	0.0266	0.2792	0.0128	0.0802	0.0104	0.0124
Anderson-Darling (p-value)	0.0528	0.1978	0.9369	0.0837	0.0343	0.4595	0.0035	0.096	0.0419	0.012
Normal?	1	1	1	1	0	1	0	1	1	0

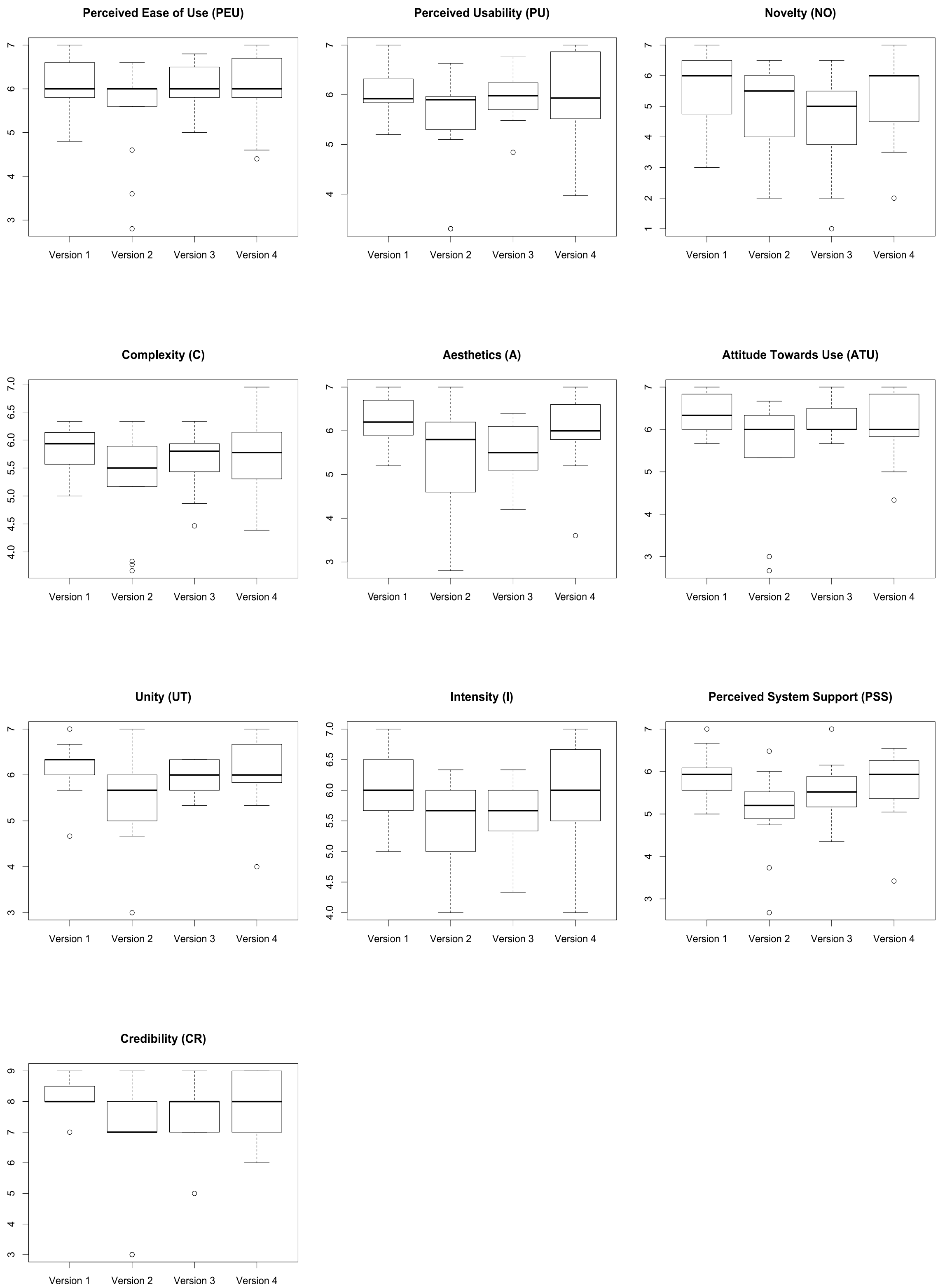


Figure 6.18: Boxplots comparing the four versions regarding perceived ease of use, perceived usability, novelty, complexity, aesthetics, attitude towards use, unity, intensity, perceived system support, and credibility.

Table 6.7: Summary of statistics and normality tests for the three metrics evaluated per steps with respect to the two ways for authoring gamification

	Understandability	Complexity_3	PSS
Target behavior (N= 28)			
Min	1	2	2.6667
Max	7	7	7
Range	6	5	4.3333
Median	6	6	6
Mean	5.4286	5.5357	5.7024
St d. Dev.	1.7518	1.4268	1.0747
Shap. Wilk (p-value)	2.00E-04	4.00E-04	3.00E-04
Anderson-Darling (p-value)	0	2.00E-04	3.00E-04
Normal?	0	0	0
Game design elements (N=31)			
Min	6	3	3.3333
Max	7	7	7
Range	1	4	3.6667
Median	6	6	6
Mean	6.4839	6.0323	5.9785
St d. Dev.	0.508	0.9826	0.8071
Shap. Wilk (p-value)	0	1.00E-04	9.00E-04
Anderson-Darling (p-value)	0	0	0.0038
Normal?	0	0	0

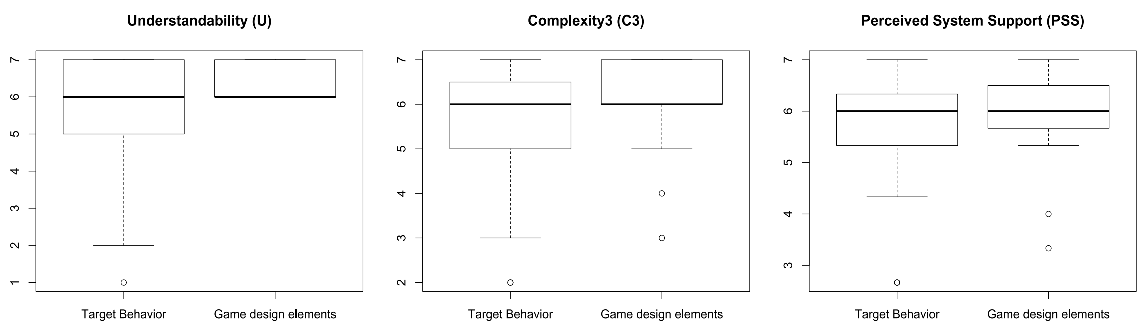


Figure 6.19: Boxplots comparing the understandability, complexity3, and perceived system support with respect to the two prototypes for authoring gamification

to reject (p-value < 0.05) or not the null hypothesis (i.e., there is no difference between the factors). As shown in the table, we have found statistical significance for the hypotheses H4 (aesthetics), H9 (perceived system support), and H11 (understandability). For the other hypotheses, our results suggest that there is no difference between the factor levels considered. Note that for the factorial analysis (i.e., more than two-group comparisons) that presented statistical significant (H4 and H9), we also applied a hypothesis test (i.e., TukeyHD) to verify which comparisons (in pairs) are also statistically significant. After applying this test, we have identified that both for aesthetics and perceived system support, version 1 (template and authoring by selecting behaviors) are better than version 2 (scratch and authoring by selecting behavior). For the others comparisons (e.g., version 1 x version 3, version 2 x version 3, and so on) we have not found significant results. In the following section we analyze and discuss these results.

Table 6.8: P-value results for the hypotheses of this experiment

Hypothesis	Test	p-value	Decision (95%)
$H1$	Kruskal-Wallis rank sum test	0.377139669	Fail to reject
$H2$	Kruskal-Wallis rank sum test	0.404490978	Fail to reject
$H3$	Kruskal-Wallis rank sum test	0.331467131	Fail to reject
$H4$	One-way ANOVA	0.021159169	Reject
$H5$	Kruskal-Wallis rank sum test	0.150977607	Fail to reject
$H6$	One-way ANOVA	0.166032869	Fail to reject
$H7$	Kruskal-Wallis rank sum test	0.18339061	Fail to reject
$H8$	Kruskal-Wallis rank sum test	0.104176058	Fail to reject
$H9$	One-way ANOVA	0.028247725	Reject
$H10$	Kruskal-Wallis rank sum test	0.09356651	Fail to reject
$H11$	Wilcoxon rank sum test with continuity correction	0.013880944	Reject
$H12$	Wilcoxon rank sum test with continuity correction	0.197600572	Fail to reject
$H13$	Wilcoxon rank sum test with continuity correction	0.348614264	Fail to reject

6.3.4 Analysis and discussion

Our results indicate that there is statistical difference with respect to aesthetics and perceived system support for the versions compared. After verifying the effects between the versions, we identified that there is statistical significance for stating that the aesthetics (adjusted p-value of 0.0333912) and perceived system support (adjusted p-value of 0.0208950) of the version 1 are better than of the version 2. These results might suggest that the prototypes that present customization by template and gamification authoring by selecting a behavior (version 1) may be more beautiful as well as give more support to aid performing the task required than version 2, which includes prototypes for customizing features from scratch and authoring gamification by selecting a target behavior. Thus, we might explain these results by analyzing some design elements of the versions that could help to enhance these effects. For instance, in version 1, the prototypes present a reduced number of steps for customizing gamified ITS features as well as make use of pre-configured design elements (e.g., selection an educational level, see Figure 6.11.

Furthermore, our results also indicate that there is statistical significance the ways of for authoring gamification (i.e., by selecting a target behavior or by selecting game design elements) are different with respect to understandability. After checking the summary of statistics (Table 6.7) and boxplot for this comparison (Figure 6.19) we can see that the prototypes that present a gamification authoring option by selecting game design elements are perceived to be more understandable than the option by selecting a target behavior. This result might be explained by the fact that the option which provides selection of a target behavior includes a longer explanation of each behavior as well as about the game design elements included per behavior (Figure 6.10, whereas for the other option there are only explanations about the game design elements. However, as seen in Table 6.7 and Figure 6.19, although both options have exactly the same median, there is an outlier that pushed the mean for the first option down affecting the effects. It is likely that without this outlier both options would present no difference with respect to understandability.

As previously mentioned, there is no statistical difference with regards to the following dependent variables for the comparison between the four versions: perceived ease of use, perceived usability, novelty, complexity, attitude towards use, unity, intensity, and credibility. There is also no statistical difference for the comparison between the two prototypes for authoring gamification with respect one dimension of complexity (C3) and perceived system support. However, although we could not identify statistical differences for the comparison between these versions, which present prototypes with different combinations of interface design elements, the scores received for all these response variables may be considered positive. As presented in Table 6.6, the median of all variables collected for the versions are above 5 (except for novelty in version 3), which might suggest that participants in general have a positive attitude towards the use our designed prototypes and somehow agreed that they may be ease to use, usable, simple, novel, unique and intense. Moreover, among the four versions, three versions (1, 3 and 4) present in terms of median a credibility with score 8, whereas version 2 presents a median credibility of 7. In addition, as shown in Table 6.7, both prototypes for authoring gamification have a 6 score as median for the understandability, complexity3 and perceived system support metrics, which also suggest that participants are likely to agree with the designed prototypes regarding these metrics.

Considering the aforementioned results, it is worth noting that all versions compared are part of the authoring solution. Hence, it is likely that the versions present similar results, particularly for metrics that are related to the interface design of the authoring (e.g., ease of use, usability, novelty, unity, and so on) since there is a standard design for the graphical interfaces.

6.3.5 Threats to the validity

Similarly to Section 5.4.5, this section describes the threats to the validity of this experiment. In general, the design of the experiment aimed at minimizing a lot of the threats discussed in this section by randomizing the versions on which participants evaluated. However, there are threats that should be considered, they are organized using the Internal, External, Construct and Conclusion categories [Wohlin et al., 2012].

Internal

As the experiment involves the active participation of humans, it was also prone to a number of internal threats, such as (i) history – it is possible that the moment at which the experiment occurred may have affected the results, however, this threat was minimized by letting participants participating of the evaluation anytime they preferred; (ii) maturation – since the participants took around 30 minutes to analyze the prototypes and answer all the questions, it is possible that they were bored or tired while answering the survey; and (iii) positive bias – as this experiment is not paired (i.e., subjects only analyze one treatment (e.g., version 1)), it is likely that participants did not have a basis for comparison with other authoring solutions. Hence, even versions with fewer features are positively evaluated by participants.

Construct

The threats of this category are mainly related to two aspects of our experiments. First, in this evaluation, participants analyzed non-interactive prototypes of our authoring computation solution. Thus, it is possible that participants could not have enough information to better analyze some constructs (e.g., ease of use, usability and complexity). However, in order to minimize the effect of the lack of interaction, we asked participants to answer questions on how they perceived these constructs, so our analysis can be only representative for perceptions on these constructs. The second aspect of our experiment that might be considered a threat is the choice of constructs used since it is possible that some constructs may not be measured by the questions. To minimize these threats we selected technology acceptance models (TAM) constructs validated with teachers or in the e-learning context.

External

The participants of the experiment are representative only for the academic context. In particular, as previously described, participants were students and professors from two research groups. In this way, we might not be able to generalize the results of this experiment to other contexts. The subjects of this evaluation must be broadened to other academic settings to obtain more generic results. Anyway, in order to amplify the external validity of our results, we selected participants from two countries (i.e., Brazil and Canada)

Conclusion

The sample size of the experiment was 59 participants, however, this number was randomly divided into four versions. Thus, there might be insufficient statistical power on the effects of the evaluation since the sample size for the versions 1, 2, 3, and 4 are, respectively, 15, 13, 16, and 15 participants. Moreover, we instrumented the survey tool to randomly allocate in a balanced way a participant to a version, however, many participants have not completed the survey. Thus, the samples are not balanced between the versions. In addition, it is also possible the the effect of the independent variables be spread since participants needed to analyze several steps.

6.4 Experiment #2: with teachers

After conducting the first study with researchers in laboratory settings, we obtained the first impressions on how our designed prototypes (non-interactive) for authoring gamified ITS are perceived by users.

However, recall that our ultimate goal is to provide a simple and usable authoring computational solution to aid teachers customizing gamified ITS features. Thus, in order to evaluate the interactive prototypes of our authoring solution only with teachers, we conduct a second experiment in a similar way to the first experiment. This new experiment explores the perceptions of teachers with respect to other constructs of the authoring process as well as with respect to prototypes of gamified ITSs graphical interfaces authored by them.

This experiment intends to analyze the interactive prototypes of the authoring solution by using template or scratch to evaluate them regarding perceived ease of use, complexity, usability, perceived utility, attitude towards use, behavioral intention to use, and perceived system support, credibility, and time to author metrics as well as representability, satisfactoriness and utility of authored gamified ITS protypes; from the viewpoint of teachers in the context of teachers in Brazil interacting with the prototypes and answering a survey about the authoring process and about the interfaces of the authored tutor.

In the following sections, we describe the materials and method used in this experiment, the procedure and participants, the results, analysis and discussion of the results as well as threats to the validity of our results.

6.4.1 Materials and methods

In this section, we describe the variables, experimental design and research hypotheses investigated in this experiment.

Variables

In this second experiment, we are considering only one independent variable, which is defined as follows and the factor levels are summarized in Table 6.9.

- Gamified ITS configuration flow: this variable refers to the two alternative flows to customize a gamified tutor in the authoring solution, i.e., configuring from scratch or using a template.

Table 6.9: Factors levels

Factor	Levels
Gamified ITS configuration flow	Configure tutor from scratch Customize using template

The effects (dependent variables) of the factors are overall analyzed with respect to similar constructs investigated in the previous experiment. However, we are not considering the aesthetics-related constructs (i.e., aesthetics, novelty, unity and intensity) since they are amenable to be effectively measured by the analysis of the non-interactive prototypes in the former experiment. Thus, beyond considering some dependent variables analyzed in the former experiment (i.e., perceived ease of use, usability, complexity, attitude towards use, perceived system support, and credibility), we also included some variables based on the work of Teo [2011] to evaluate the authoring tool prototypes: usefulness and behavioral intention to use. In addition, we measure the time spent by teachers to author a tutor using each version. We also defined three new dependent variables (i.e., representability, satisfaction, and utility) to verify the perception of teachers with respect to prototypes of authored gamified ITS configured by using the authoring solution. As we already explained some of these metrics in the previous experiment, we describe the new metrics below.

- Usefulness (USE): This construct refers to the degree to which a teacher believes that using technology would enhance his or her job performance [Teo, 2011];
- Behavioral intention to use (BIU): This construct refers to the degree of a teacher's willingness to use technology [Teo, 2011];

- Time (T): This metric measures the time spent by teachers to customize gamified ITS features using both versions;
- Representability (R): This variable refers to how the authored gamified ITS reflects the choices made by teachers during the authoring process;
- Satisfaction (S): This variable refers to the degree to which a teacher is satisfied with the authored gamified ITS prototype after configuring a tutor;
- Utility (UTI): This variable refers to the degree of a teacher's perception about the utility of the authored gamified ITS to his or her students.

Experimental design

We used a full-factorial design, on which participants were shown the two versions of the interactive prototypes. In the first version, authoring is performed by scratch, whereas, in the second one, authoring is performed using a template. Each participant interacts with both versions (the order of interaction is randomized in our experiment). Figure 6.20 shows the flow of the prototypes presented to participants according to the order of interaction.

Note that, by contrast to the previous experiment, we do not collect answers by each step of the authoring process, data is collected after participants use each version. As presented in the figure, participants are asked to answer demographic questions as well as about the authoring tool and the authored tutor (see questions in Table 6.10). Time is measured by the system designed to instrument the experiment and a likert scale from 1 (completely disagree) to 7 (completely agree) is used for most variables, except for credibility, representability, satisfaction, and utility, which use a scale from 1 (very bad) to 9 (very good). To compute the overall score of the metrics, for the variables that include more than one question, we must calculate the average of the variable according to the answer to each variable's question.

Moreover, to illustrate how teachers answer questions about the authored tutor, Figure 6.21 presents a prototype of an authored gamified ITS after performing the steps illustrated following one of the orders showed in Figure 6.20.

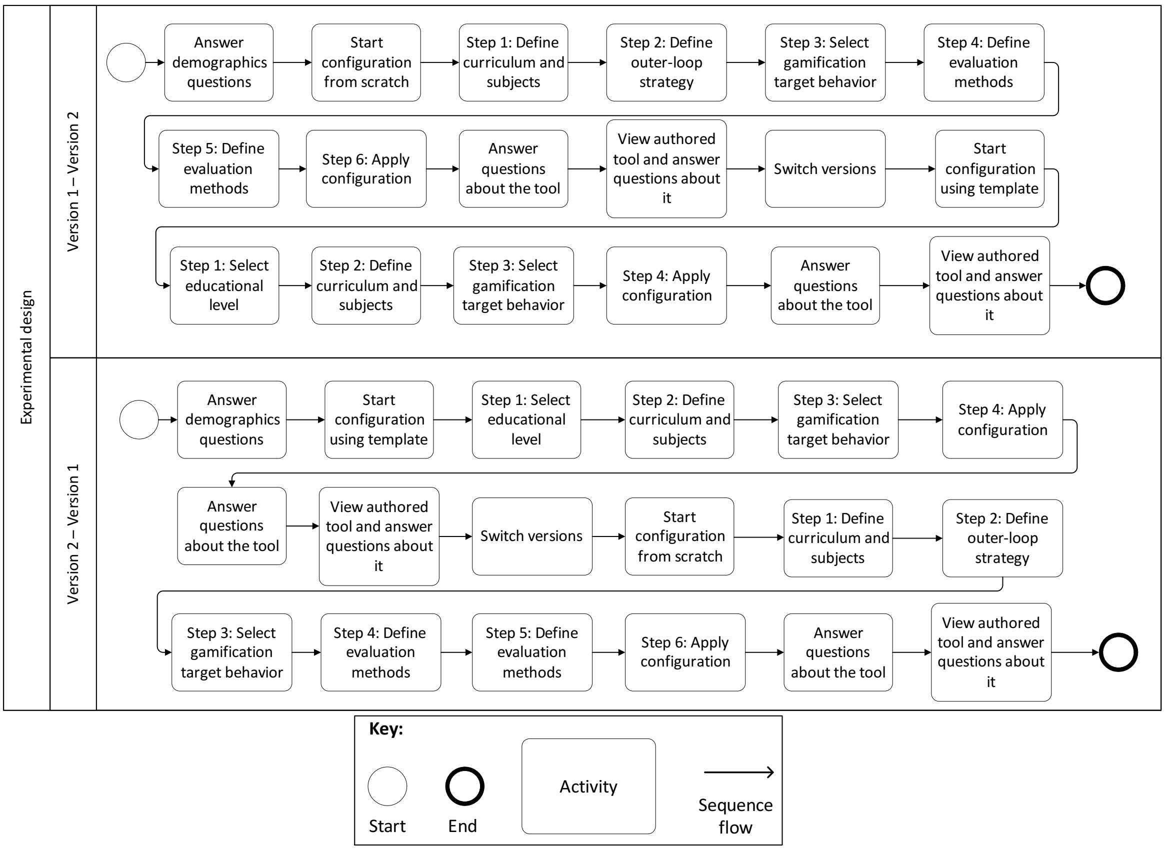


Figure 6.20: Trials definition of flow of tasks performed by teachers to participate of the second experiment

Figure 6.21: Example of a gamified ITS prototype authored that can be generated in the experiment according to teachers’ choices

Table 6.10: Questions used to measure the constructs.

Questionnaire
Perceived ease of use (PEU)
PEU1: Learning to author the educational system was easy
PEU2: I think that would be easy to use this authoring tool to do what is needed to do
PEU3: The interaction with this authoring tool does not require much effort
PEU4: I think that would be easy for me to become skilful at using this authoring tool
PEU5: I think that this authoring tool is easy to use
Usability (U)
Understandability (UN): This authoring tool is clear and understandable
Flexibility (F): I think that this authoring tool is flexible to interact with
Fuctionality (FU): The authoring tool have good functionality (features)
Navigation (N): I feel that I would have an intuitive sense to author an educational system using the tool
Memorability (M): I feel that it would be easy to remember how to perform tasks using the tool
Complexity (C)
C1: The authoring tool is simple
C2: The authoring tool is well designed to employ diverse components and design styles
C3: The degree of information load on this authoring tool is very well designed
Usefulness (USE)
USE1: Using this authoring tool enables me to perform task more slowly
USE2: Using this authoring tool decreases my performance
USE3: Using this authoring tool decreases my productivity
USE4: Using this authoring tool decreases my efficacy
Attitude towards use (ATU)
ATU1: Overall, configuring an educational system using the authoring tool is good
ATU2: Overall, I have formed a favorable impression about this authoring tool of educational systems
ATU3: Overall, I have positive feelings about this authoring tool
Behavioral intention to use (BIU)
BIU1: I would have interest to continue to use this authoring tool in the future
BIU2: I expect that I would use this authoring tool in the future
BIU3: If available, I plan to use this authoring tool in the future
Perceived system support (PSS)
PSS1: The authoring tool quality including help function and instructional support is good
PSS2: The authoring tool support for completing the task is good
PSS3: The authoring tool provides personalized support (e.g., there are options which enable me to specify my preferences)
Credibility (CR): In general, what is the credibility of the authoring tool?
Representability (R): How the system reflects my previous authoring choices?
Satisfaction (S): Your degree of satisfaction with the authored gamified educational system
Utility (UTI): How useful is providing the authored gamified educational system to your students?

Research hypotheses

Based on the variables described, the following research hypotheses are investigated in this second experiment:

Table 6.11: Hypotheses of the second experiment

H1-0: The complexity of the versions is equal
H1-1: The complexity of the versions is different
H2-0: The usefulness of the versions is equal
H2-1: The usefulness of the versions is different
H3-0: The authoring time using the versions is equal
H3-1: The authoring time using the versions is different
H4-0: The perceived ease of use (PEU) of the versions is equal
H4-1: The perceived ease of use (PEU) of the versions is different
H5-0: The usability of the versions is equal
H5-1: The usability of the versions is different
H6-0: The attitude towards use of the versions is equal
H6-1: The attitude towards use of the versions is different
H7-0: The behavioral intention to use of the versions is equal
H7-1: The behavioral intention to use of the versions is different
H8-0: The perceived system support of the versions is equal
H8-1: The perceived system support of the versions is different
H9-0: The credibility of the versions is equal
H9-1: The credibility of the versions is different
H10-0: The perceived representability of the tutor authored using the versions is equal
H10-1: The perceived representability of the tutor authored using the versions is different
H11-0: The perceived satisfaction of the tutor authored using the versions is equal
H11-1: The perceived satisfaction of the tutor authored using the versions is different
H12-0: The perceived utility of the tutor authored using the versions is equal
H12-1: The perceived utility of the tutor authored using the versions is different

In Table 6.12, these research hypotheses are formally presented. As presented in Table 6.10, C, USE, T, PEU, U, ATU, BIU, PSS, and CR, are functions that return, respectively, the value of complexity, usefullness, authoring time, perceived ease of use, perceived usability, attitude towards use, behavioral intention to use, perceived system support, and credibility of the versions 1 (scratch) and 2 (template).The functions R, S, and UTI return the value of representability, satisfaction and utility of the the authored gamified ITS after using versions 1 and 2.

6.4.2 Procedure and participants

This section describes how this experiment was executed. It describes who the participants are (and how they were selected), which instruments were used and how the experiment was performed.

Table 6.12: Formal definition of the research hypotheses' second experiment. V1 = scratch and V2 = template

Hypothesis	Null Hypothesis	Alternative Hypothesis
H1	*H1-0* : $\mu_C(V1) = \mu_C(V2)$	*H1-1* : $\mu_C(V1) \neq \mu_C(V2)$
H2	*H2-0* : $\mu_{USE}(V1) = \mu_{USE}(V2)$	*H2-1* : $\mu_{USE}(V1) \neq \mu_{USE}(V2)$
H3	*H3-0* : $\mu_{PEU}(V1) = \mu_{PEU}(V2)$	*H3-1* : $\mu_{PEU}(V1) \neq \mu_{PEU}(V2)$
H4	*H4-0* : $\mu_U(V1) = \mu_U(V2)$	*H4-1* : $\mu_U(V1) \neq \mu_U(V2)$
H5	*H5-0* : $\mu_T(V1) = \mu_T(V2)$	*H5-1* : $\mu_T(V1) \neq \mu_T(V2)$
H6	*H6-0* : $\mu_{ATU}(V1) = \mu_{ATU}(V2)$	*H6-1* : $\mu_{ATU}(V1) \neq \mu_{ATU}(V2)$
H7	*H7-0* : $\mu_{BIU}(V1) = \mu_{BIU}(V2)$	*H7-1* : $\mu_{BIU}(V1) \neq \mu_{BIU}(V2)$
H8	*H8-0* : $\mu_{PSS}(V1) = \mu_{PSS}(V2)$	*H8-1* : $\mu_{PSS}(V1) \neq \mu_{PSS}(V2)$
H9	*H9-0* : $\mu_{CR}(V1) = \mu_{CR}(V2)$	*H9-1* : $\mu_{CR}(V1) \neq \mu_{CR}(V2)$
H10	*H10-0* : $\mu_R(V1) = \mu_R(V2)$	*H10-1* : $\mu_R(V1) \neq \mu_R(V2)$
H11	*H11-0* : $\mu_S(V1) = \mu_S(V2)$	*H11-1* : $\mu_S(V1) \neq \mu_S(V2)$
H12	*H12-0* : $\mu_{UTI}(V1) = \mu_{UTI}(V2)$	*H12-1* : $\mu_{UTI}(V1) \neq \mu_{UTI}(V2)$

Participant selection

Similarly to previous study, the experiment involves the participation of human agents. Participants were teachers and professors, working at different educational levels, and were invited in one of the following ways: (i) by sending e-mail invitations to all professors of the Federal University of Alagoas; (ii) by sending email invitations to teachers registered in the Brazilian Conference on Computers and Education (2015); (iii) by sending invitation e-mails to the computers and education mailing list; and (iv) posting an invitation on the computers and education facebook group. It is worth noting that before sending these invitations, we conducted a pilot study in laboratory settings (i.e, in the NEES research group) to receive feedback and to adjust our instruments.

Preparation and instrumentation

The data collection was performed through the use of a survey that includes our experimental design. As showed in Figure 6.20, after a participant agrees with the terms and answers demographic questions, there is a randomized allocation of an authoring version to interact. After performing the steps for authoring, participants answer a questionnaire regarding the authoring process for using each version as well as visualize and evaluate a prototype of the authored gamified tutor. The system that instruments our experiment is available at http://surveys.nees.com.br/agits/.

6.4.3 Results

This section presents the analysis of the data collected in this experiment. The collected data, as well as the scripts used in the experimental analysis are available at https://goo.gl/7Tkr4I.

Similarly to the previous experiment, before presenting the descriptive and inferential statistic results of this second experiment, we depict the demographic statistics for the participants of this study (Table 6.13). As seen in figure, participants provided information about their gender, age, occupation, education level, country, educational level, informatics skills, whether they received or not prior training to use educational technologies, and whether they consider themselves able to use educational technologies. In the following section, we present the descriptive statistics for the results of this experiment.

Descriptive statistics and assumptions verification

The collected data contains the participants' answers to the questions shown in Table 6.10 for each answer regarding the dependent variable. Note that, except for the Credibility (CR), Representability (R), Satisfaction (S), and Utility (UTI) which can receive a value from 1 to 9 as well as time that is measured in minutes, all the other dependent variables are measured by the average of answers regarding each variable using a likert scale (from 1 to 7). Thus, to analyze these results, we first conduct a descriptive analysis of the data, by analyzing histograms and boxplots of the computed metrics.

In Table 6.14, we present the summary of statistics (e.g., median, mean, sd) and the results of the normality testes (e.g., shapiro-wilk and anderson-darling tests) we applied for the perceived ease of use (PEU), usability (U), complexity (C), usefulness (USE), attitude towards use (ATU), behavioral intention to use (BIU), perceived system support (PSS),

Table 6.13: Participant demographics of the second experiment

Demographics	Version 1	Version 2
Size (n)	36	41
Gender		
Female	20 (55.55%)	23 (56.1%)
Male	16 (44.45%)	18 (43.9%)
Rather not say	0%	0%
Age		
18–25	0%	0%
26–40	13 (36.11%)	18 (43.9%
40–65	21 (58.33%)	21 (51.2%)
Over 65	2 (5.55%)	2 (4.9%)
Rather not say	0%	0%
Occupation		
Student	0%	0%
Teacher	36 (100%)	41 (100%)
Other	0%	0%
Education level		
Junior High/Middle School	0%	0%
High School	0%	0%
Technical/trade school	0%	0%
Bachelor’s degree	9 (25%)	14 (34.14%)
Master’s degree	10 (27.77%)	11 (26.82%)
Doctorate degree	17 (47.22%)	16 (39.02%)
Other	0%	0%
Country		
Brazil	36 (100%)	41 (100%)
Skills		
Advanced	12 (33.33%)	12 (29.26%)
Beginner	1 (2.77%)	1 (2.43%)
Intermediate	23 (63.88%)	28 (68.29%)
Training		
No	25 (69.44%)	26 (63.41%)
Yes	11 (30.55%)	15 (36.58%)
Capability		
No	6 (16.66%)	7 (17.07%)
Yes	30 (83.33%)	34 (82.92%)

credibility (CR), time (T), representability (R), satisfaction (S), and utility (UTI) metrics per version analyzed in this experiment. We also present in Figure 6.22 the boxplots for these metrics comparing the two versions analyzed in this second experiment.

Table 6.14: Summary of statistics and normality tests for the ten metrics evaluated per version

	C	USE	Time (min)	PEU	U	ATU	BIU	PSS	CR	R	S	UTI
Version 1 (N=36)												
Min	2.333	1	1.31	3	2	1.667	1	1.333	2	3	3	2
Max	7	7	15.473	7	7	7	7	7	9	9	9	9
Range	4.667	6	14.163	4	5	5.333	6	5.667	7	6	6	7
Median	5.667	3	5.088	5.9	5.5	6	5.833	5.333	7	8	7	8
Mean	5.306	3.076	5.519	5.661	5.306	5.574	5.269	5.278	6.833	7.25	7	7.361
St d. Dev.	1.158	1.655	3.284	0.987	1.027	1.086	1.423	1.128	1.682	1.697	1.805	1.9
Shap. Wilk (p-value)	0.024	0.025	0.035	0.023	0.088	0.001	0	0.008	0.004	0	0.003	0
Anderson-Darling (p-value)	0.042	0.064	0.164	0.025	0.297	0.007	0	0.042	0.005	0	0.003	0
Normal?	0	1	1	0	1	0	0	0	0	0	0	0
Version 2 (N=41)												
Min	1.667	1	0.45	2	1.8	1	1	1.667	3	2	2	1
Max	7	7	14.74	7	7	7	7	7	9	9	9	9
Range	5.333	6	14.29	5	5.2	6	6	5.333	6	7	7	8
Median	5	3	2.607	5.6	5	5	5	5	7	7	7	7
Mean	5.049	3.006	3.367	5.537	5.151	5.154	5.057	5.008	6.829	6.78	6.537	6.805
St d. Dev.	1.284	1.266	2.727	1.185	1.15	1.401	1.52	1.248	1.611	1.93	1.818	2.076
Shap. Wilk (p-value)	0.075	0.081	0	0.009	0.058	0.003	0.007	0.034	0.009	0.001	0.028	0.001
Anderson-Darling (p-value)	0.114	0.23	0.001	0.053	0.114	0.007	0.03	0.032	0.006	0.002	0.033	0.001
Normal?	1	1	0	1	1	0	1	0	0	0	0	0

Inferential statistics

Recall that we are investigating twelve hypotheses to analyze the use of two different interactive ways to author gamified ITS considering nine metrics related to authoring tool and three metrics related to a prototype of gamified ITS that is produced based on the choices teachers make during the experiment.

To verify the hypotheses, statistical tests were applied for each one of the hypotheses formalized in Table 6.12. The hypotheses (H1 to H12) includes the comparison between two versions, hence, we apply two-group hypothesis tests. In order to decide which tests to apply, we first verified the normality of the data regarding the hypotheses (see the results in Table 6.14). For the normal distributions (i.e., when all factor levels are normal) we

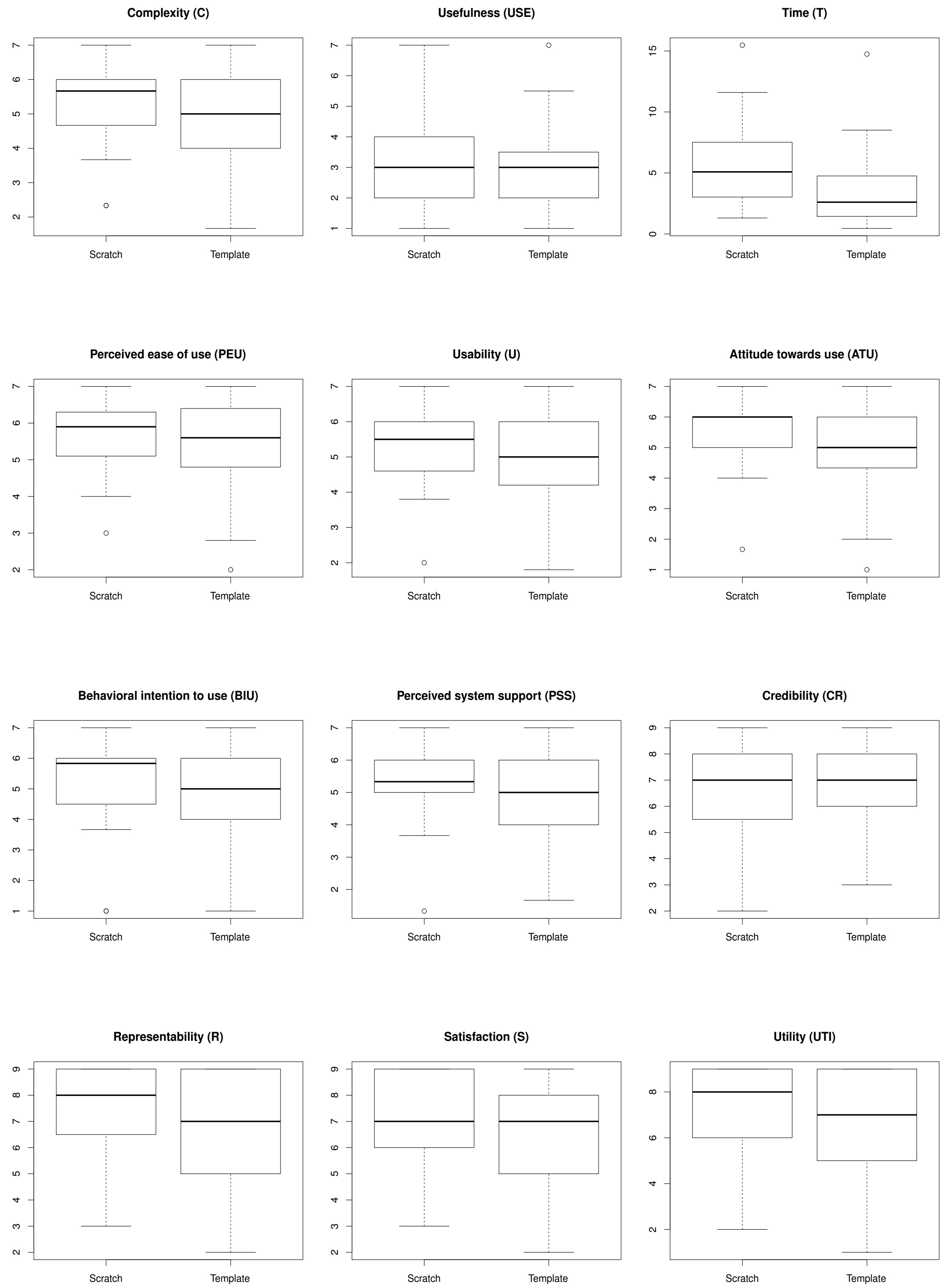


Figure 6.22: Boxplots comparing the two versions regarding complexity, usefulness, time, perceived ease of use, usability, attitude towards use, behavioral intention to use, perceived system support, credibility, representability, satisfaction, and utility

applied a parametric test (t-test), whereas, for the non normal distributions, we applied a non-parametric test (Wilcoxon test).

Table 6.15 presents the results after applying the tests for the hypotheses of this experiment. The hypotheses, applied test, p-value and the decision if the resultant p-value is enough to reject (p-value < 0.05) or not the null hypothesis (i.e., there is no difference between the factors) are presented in this table. As shown, we found statistical significance for the hypothesis H9 (time for authoring). For the other hypotheses, our results suggest that there is no statistical difference between the two versions considered. In the following section we analyze and discuss these results.

Table 6.15: P-value results for the hypotheses of the second experiment

Hypothesis	Metric	Method	p-value	Decision (95%)
$H1$	Complexity	Wilcoxon rank sum test with continuity correction	0.387900237	Fail to reject
$H2$	Usefulness	Welch Two Sample t-test	0.836533973	Fail to reject
$H3$	Time	Wilcoxon rank sum test	0.00081419	Reject
$H4$	Perceived ease of use	Wilcoxon rank sum test with continuity correction	0.643764588	Fail to reject
$H5$	Usability	Welch Two Sample t-test	0.535877832	Fail to reject
$H6$	Attitude towards use	Wilcoxon rank sum test with continuity correction	0.191883734	Fail to reject
$H7$	Behavioral intention to use	Wilcoxon rank sum test with continuity correction	0.459741939	Fail to reject
$H8$	Perceived system support	Wilcoxon rank sum test with continuity correction	0.286745514	Fail to reject
$H9$	Credibility	Wilcoxon rank sum test with continuity correction	0.950243262	Fail to reject
$H10$	Representability	Wilcoxon rank sum test with continuity correction	0.312598008	Fail to reject
$H11$	Satisfaction	Wilcoxon rank sum test with continuity correction	0.252061798	Fail to reject
$H12$	Utility	Wilcoxon rank sum test with continuity correction	0.191234576	Fail to reject

6.4.4 Analysis and discussion

As shown in Table 6.15, our results indicate that there is statistical difference (p-value of 00081419) with respect to the authoring time metric for both authoring versions. This result might suggest that using templates demand less time from teachers (see Table 6.14 and Figure 6.22) than authoring from scratch. Note that this result was expected, since we provide, in this version, pre-configured gamified ITS by educational levels that demand less choices by teachers. However, this is not obvious since we are investigating the use of additional features that may require some cognitive effort from teachers to understand.

Additionally, regarding the authoring tool process using both versions, there is no statistical difference with regards to the following dependent variables for the comparison between the two versions: perceived ease of use, usability, complexity, usefulness, attitude towards use, behavioral intention to use, perceived system support, and credibility. One might note that not finding significant differences between the versions is not necessarily a bad result. In fact, as presented in Table 6.14, the average scores received per metric for both versions may be considered positive. The perceived ease of use, usability, complexity, attitude towards use, and behavioral intention to use receive average scores above 5 (agree). Note that the complexity metric is measure with negative assertions, i.e., high scores in this metric indicates simpler prototypes. The usefulness metric also present negative assertions (as presented in Table 6.10), present average scores close to 3 (disagree). For the credibility score, the average score for both versions are close to 7 (in a scale from 1 to 9). With respect to the three metrics (representability, satisfaction, and utility) that are related to the gamified ITS prototypes generated after teachers have chosen the features of their tutors, we could not find statistical difference between the versions. However, we might also note that the average score for these metrics are also close to 7 (in a scale from 1 to 9).

Taken together, our results might suggest that in general teachers have positive perceptions regarding the two ways for customizing gamified ITS features. Thus, despite the version 1 requires less time to author gamified ITS than the version 2, one might say that participants somehow agreed that both of our authoring solution versions are ease to use, usable, and simple. Our results also indicate that teachers have a positive attitude towards the use of both versions of our authoring solution, moreover, it may also suggest that teachers behave with intention to use our solution and perceive that our versions provide support for completing the authoring task. Finally, teachers are also likely to agree that the authored gamified ITS prototypes are perceived to be representative of the choices made by them during the authoring process. Teachers are also amenable to be satisfied by the authored prototypes and find them helpful for their students.

6.4.5 Threats to the validity

Likewise the former experiment, this section describes the threats to the validity of this second experiment. In general, the design of the experiment aimed at minimizing a lot of the threats by using a randomized full-factorial experimental design. However, there are threats that should be considered, they are organized using the Internal, External, Construct and Conclusion categories [Wohlin et al., 2012].

Internal

In the same way of the first experiment, this experiment also involves humans, thus it was prone to several internal threats regarding subjects. They can be: (i) history – it is possible that the moment at which the experiment occurred may have affected the results, however, this threat was minimized by letting participants participating of the evaluation anytime they preferred; and (ii) maturation – since the participants took an average of almost 9 minutes to interact with both authoring solution versions and answer all the questions, it is possible that they were bored or tired while answering the questionnaire for a particular version. To alleviate the threats with respect to the subject that participated of this experiment, we randomized the order of treatments of the experiment.

Construct

In order to reduce one of the construct threats of our previous experiment, we conduct, in this new experiment, a study with interactive versions of our authoring solution. However, likewise the previous experiment, some threats that may also be applied to this experiment is our constructs choices. For instance, the perceived usability (PU) dependent variable measures the perceptions of users on the usability of the prototypes, hence, it might be considered a threat to the construct validity. To minimize these threats, most of our constructs are validated using the technology acceptance models (TAM) with teachers or in the e-learning context.

External

As showed in Table 6.13, teachers are only representative for Brazil . Moreover, most of the participants have advanced or intermediate skills to use information technologies. In this way, we might not be able to generalize the results of this experiment to other contexts. This experiment must be extended considering other countries and IT skills to support the generalization of our results. Anyway, aiming to amplify the external validity of our results, we invited the participation of teachers in heterogeneous contexts (i.e., university, middle-schools, and computers and education teachers/researchers).

Conclusion

The sample size of this second experiment was 36 for the version 1 and 41 for the version 2. We instrumented the experimental system to randomly order the version on which teachers interact with in a balanced way. However, some participants have not answered the questionnaires for one of the versions; this is the reason why our sample sizes are not balanced for both versions. Thus, there might be effects that have more statistical power for one version than the other.

6.5 Concluding remarks

In this chapter, we firstly presented how the authoring process of gamified ITS takes place in this thesis, depicting which components are authorable in this work. Moreover, we also described how we specified the requirements, developed the architecture, and implemented the authoring solution proposed in this thesis. We also presented how the system prototypes were defined at different prototyping levels. To empirically evaluate our authoring solution, we conducted two experiments.

The first experiment intended to analyze its non-interactive prototypes investigating the combination of two features (way for customizing features and way for authoring gamification) in the authoring process with respect to perceived ease of use, perceived usability, complexity, aesthetics, novelty, unity, intensity, attitude towards use, perceived system support, and credibility metrics in the context of graduate students and researchers from two research groups in Brazil and Canada.

The results of the first experiment allowed us to state that (i) there is statistical difference with respect to aesthetics and perceived system support between the version that uses template and that allows gamification authoring by selecting a behavior (version 1) is better than the version that do not use template and that allows the same way to author gamification; (ii) authoring gamification by selecting game design elements is more understandable than by selecting a behavior; (iii) there is no difference among the combinations of using or not the template feature and the two ways for authoring gamification with respect to the perceived ease of use, perceived usability, complexity, novelty, attitude towards use, unity, intensity, and credibility metrics; (iv) there is no difference between the two ways for authoring gamification with regards to the perceived system support and complexity3 (information load); (v) the average scores of all metrics for the four versions evaluated in a likert scale from 1 to 7 are greater than 5 (partially agree), for some versions (e.g., perceived ease of use, and attitude towards use of the version 1), greater than 6 (agree); and (vi) despite there is no statistical difference between the version with regards the credibility metric, one might note that considering the average score (scale from 1 to 9), version 1 (template and behavior) received the highest score of 8.2, followed by version 4 (no template and game design elements) with 7.886, version 3 (template and game design elements) with 7.625, and version 4 (no template and behavior) with 6.9231.

After conducting the first experiment, our authoring solution was improved based on the results collected from participants and a second experiment was performed. The second experiment intended to analyze the interactive prototypes of the authoring solution by using two ways of authoring (scratch or template) regarding perceived ease of use, complexity, usability, perceived utility, attitude towards use, behavioral intention to use, and perceived system support, credibility, and time to author metrics as well as representability, satisfactoriness and utility of authored gamified ITS prototypes generated. Participants were teachers in Brazil and they interacted with the authoring versions and answered a survey about the authoring system and about the interfaces of the authored tutor.

The results of the second experiment allowed us to state that (i) authoring gamified ITS by using template requires less time than from scratch; (ii) there is no difference in the use or not of template for authoring gamified ITS with regards to perceived ease of use, usability, complexity, usefulness, attitude towards use, behavioral intention to use, perceived system support, and credibility; (iii) there is no difference in the use or not of template in the authoring process with respect to the perceived representability and utility of the authored gamified ITS as well as teachers have similar satisfaction scores on the authored system after using a version with template and a version with no template; and (iv) the average scores of all metrics for the versions evaluated in a likert scale from 1 to 7 with positive assertions are greater than 5 (partially agree) and the average scores of the metric evaluated with negative assertions (i.e., usefulness) are close to 3 (partially disagree); (v) the credibility average score of the version that do not use template is higher (7.361) than the credibility of the version that uses template (6.805); (vi) the average scores regarding the representability, satisfaction, and utility of authored gamified ITS prototypes after using or not template are close to 7.

The results found in this chapter provide important insights in order to contribute to the active participation of teachers in the design of gamified intelligent tutoring systems in simple, usable, and fast ways. By using our authoring computational solution, teachers may take advantage of gamification and ITS theories and practices to customize gamified ITS according to their preferences. In addition, considering the full implementation of the technological infrastructure proposed in this work (i.e., which includes the use of an ontology-based feature model and a gamified tutoring ontology as well as third-party gamified ITS that can reason on such ontologies), it would be possible to combine the human intelligence of teachers with the artificial intelligence provided by ITS capabilities.

In the next chapter, we present the final remarks of this thesis, highlighting our contributions, describing our limitations, and pointing out future works.

Chapter 7

Conclusions and future works

This chapter summarizes what we proposed in this thesis to achieve our objectives. Next, we highlight the main contributions of this thesis, listing the publications related to this work and the papers that we submitted as well papers to be submitted as a result of this thesis. We also describe the limitations of this work, and, finally, we suggest several topics to explore in further researches.

7.1 Conclusions

In this thesis we presented a solution for authoring gamified intelligent tutoring systems (named AGITS). This solution makes use of an ontology-based feature model (specified based on the OntoSPL ontology proposed) to enable the management of the variability of gamified ITS features that were identified. The authoring solution also takes advantage of an integrated ontological model (GaTO) that connects gamification and ITS concepts as well as design principles in order to constrain the variability space of gamified ITS based on such theories and practices and to aid managing the design of gamified ITS in an intoperable way.

After describing the research problem and objectives of this thesis (Chapter 1), and presenting the theoretical background involved in the development of this work (Chapter 2), we described how we analyzed the literature related to our contributions. We conducted three systematic reviews for investigating the literature to identify related works. The first one identified 1 (one) related work that use software product line and/or feature modeling to deal with the high variability of gamified ITS features. The second SLR – complemented by a snow-ball searching – identified 10 (ten) existing works that use ontologies for representing feature models. The third one identified 33 (thirty-three) ITS authoring tools that could be considered related to our work. We also found 5 (five) works that apply gamification in intelligent tutoring systems. Our work was compared against all these related works, ascertaining the originality of the contributions presented in this thesis.

Next, based on the literature and on industrial gamified ITS (i.e., MeuTutor and Duolingo), we defined a reference feature model for representing the variability of gamified ITS. In order to make such feature model reasonable by machine, including by third-party software systems, we developed a generic ontology (OntoSPL) for representing feature models. This ontology was empirically evaluated in comparison to a well-know feature model ontology (i.e., [Wang et al., 2007]) with respect to changing scenarios to measure the reasoning flexibility, time and correctness of OntoSPL. After evaluating this ontology, we represented the reference feature model identified using the OntoSPL ontology. We also illustrated how a particular configuration of gamified ITS (i.e., the features included in MeuTutor-ENEM) is realized by using such ontology.

Afterwards, aiming to formally represent and connect theories and practices about gamification and ITS in order to further constrain the design space of gamified ITS and to aid the authoring process, we developed an integrated ontological model. To conceptualize such model, we first analyzed the literature and identified six behaviors along with a set of game design elements that were evidence-supported by empirical studies – from three systematic reviews on the use of gamification – in the e-learning domain. Using an ontology engineering methodology (i.e., METHONTOLOGY), we also formalized a gamification domain ontology (GaDO) that represents the core concepts about gamification as well as concepts of particular gamification theories (e.g., Self-Determination Theory) and a gamification design framework (6D framework). This ontology also considers the gamification design practices (i.e., behaviors along with theirs respective game design elements) in the conceptualization. Then, the ontological model connects concepts from the GaDO ontology and from existing ITS ontologies to provide an integrated ontological model of gamified tutoring (named GaTO). After specifying the ontologies, they were evaluated by 5 (five) experts using an ontology evaluation method (i.e., the FOCA methodoloy) that is based on knowledge representation roles. The results of the evaluation supported the improvement of out integrated ontological model to be used by the authoring solution.

Thus, in order to support teachers to configure gamified ITSs managing the high variability of these systems and at the same time leveraging theories and practices to aid the configuration process in simple and usable ways, we designed and implemented an authoring solution for teachers. First, considering the design trade-offs between usability and flexibility of ITS authoring tools, we decided which components we would allow authoring in the authoring solution based on a generic development process for gamified ITS. After that, we described how we specified the requirements and prototypes, designed the architecture, and implemented the authoring solution proposed in this thesis. As we did not have any basis for comparison, to empirically evaluate our solution, we conducted two controlled experiments varying some features of our authoring proposal. The first experiment intended to analyze non-interactive prototypes investigating four combinations of activated or deactivated features (authoring using or not template and gamification authoring by selecting target behaviors or game design elements) in the authoring process with respect to 9 (nine) constructs in the context of graduate students and researchers from two research groups in Brazil and Canada. The results of the first experiment were used to improve the authoring solution prototypes. After these improvements, a second experiment was performed to analyze the interactive prototypes of the authoring solution only with teachers in Brazil. In the second experiment, teachers use the two ways of authoring (template and no template (scratch)) and evaluate them regarding 9 (nine) metrics as well as evaluate 3 (three) metrics related to the authored gamified ITS prototypes generated after using both way for authoring.

7.2 Main contributions

As presented in this work, the main contributions of this thesis are targeting three main problems from the artificial intelligence in education research: (1) managing the high complexity and variability of designing gamified ITS; (2) applying gamification to ITS considering theories and design practices; and (3) providing simple and usable solutions to enable teachers customizing gamified ITSs. In following, we summarize the contributions of this thesis according to these research problems.

1. **Managing the high complexity and variability of designing gamified ITS:**

    - *OntoSPL ontology*: in order to deal with the high variability of gamified ITS features, we first contributed to the software engineering research by defining an ontology for representing feature models (named OntoSPL). This ontology is based on OWL individuals and supports the configuration of features in a way to favor reconfiguration when is needed. We evaluated this ontology [Dermeval et al., 2015a] in comparison to the ontology proposed by Wang et al. [2007] (based on OWL classes and properties) in several changing scenarios. Our results indicate that using OntoSPL is more flexible and demands less time for changing than the ontology proposed by Wang et al. [2007];

    - *Ontology-based feature modeling of gamified ITS*: to enable the representation of gamified ITS variability in a formal way, we first identified a gamified ITS feature model based on existing literature and on industrial gamified ITS. Thus, we take advantage of the OntoSPL ontology to specify the gamified ITS feature model identified according to such ontology. Our ontology-based feature modelling approach enables the management of gamified ITS features at runtime of specific gamified ITS configurations (that are made by the teachers using an authoring solution) as well as that would allow third-party tutors to reason on such ontology specification to be reconfigured according to authors' configurations.

2. **Applying gamification to ITS in a formal way considering theories and design practices:**

    - *Evidence-supported gamification target behaviors in e-learning*: one of our contributions is mapping and grouping – based on three systematic reviews on the literature about gamification in the context of e-learning – six student behaviors (along with a set of game design elements) that could be targeted by gamified tutors and that are supported by positive evidence found in the empirical studies provided by the reviews. The behaviors are used to constrain the design space of gamified ITS, represented in an ontology-based feature model, based on evidence reported by current literature which may aid the configuration of gamified tutors in a way that it could be more amenable to be effective for achieving such behavior by the use of gamification;

- *Gamification domain ontology (GaDO)*[1]: we developed the GaDO ontology to formalize the knowledge about gamification domain in a way that it can be used to aid the application of gamification to ITS considering theories and design practices. This ontology conceptualizes the concepts about gamification theories and frameworks considered in the scope of this thesis (e.g., Self-determination theory, BrainHex player model, and the 6D framework) as well as the design practices represented by the behaviors identified in our previous contribution. The GaDO ontology might be useful to support the application of gamification in the e-learning domain considering such theories and practices and was evaluated by experts on the topic. The results of the evaluation suggest that the ontology may be properly targeting its aim with respect to the knowledge representation principles;
- *Gamified tutoring ontology (GaTO)*[2]: this ontological model connects the concepts specified in the GaDO ontology to ITS concepts represented in existing ITS ontology. The GaTO ontology operationalizes the knowledge repository involved in the application of gamification to ITS. Thus, this integrated ontological model might be of great importance to support the design of gamified ITS considering theories and design practices from gamification and ITS. Based on the state of the art analysis, the GaTO ontology is the only one that formalize the knowledge about theories and practices regarding gamified ITS and was also positively evaluated by experts on the topic with regards to the knowledge representation. This ontology may aid the authoring process of gamified tutors (by leveraging such theories and practices) and represents the knowledge generated in the configuration process (e.g., domain model) performed by authors.

3. **Providing a simple and usable solution to enable teachers customizing gamified**

[1]GaDO-core is available at `http://surveys.nees.com.br/ontologies/gado_core.owl` and GaDO-full is available at `http://surveys.nees.com.br/ontologies/gado_full.owl`

[2]GaTO is available at `http://surveys.nees.com.br/ontologies/gato.owl`

ITSs:

- *A computation solution for Authoring Gamified ITS (AGITS)*: to allow teachers customizing gamified ITS taking advantage of gamification and ITS theories and design practices as well as dealing with the high variability and complexity for designing these systems, we developed an authoring computation solution for gamified ITS. This authoring solution was developed considering ITS authoring design trade-offs broadly reported by the literature (usability vs. flexibility). Our intention was to provide a new solution flexible enough to enable teachers to personalize gamified ITS according to their own preferences and, at the same time, being usable and simple to use by them. As such, we support authoring (fully or partially) of the domain model, gamification model, pedagogical model, and extra features of gamified ITS by teachers and we empirically investigated how different versions (exploring the use or not of a template feature combined with authoring gamification based on identified target behaviors) of our solution are perceived by teachers in terms of simplicity, usability, and several other constructs that could be related to their technology acceptance in the context of educational technologies. In general, our results allowed us to conclude that our designed authoring solution have been positively evaluated (with respect to ease of use and simplicity, among others constructs) by participants (students and teachers) of both studies conducted. Moreover, the results of the study only with teachers might also suggest that, although using template demands less authoring time, the ease of use, usability, complexity, usefulness, attitude towards use, behavioral intention to use, perceived system support, and credibility of both the authoring process by using a template or by scratch are perceived in a similar and positive way by teachers. The representability, satisfaction, and utility of gamified ITS prototypes generated after teachers have chosen the features of their tutors during the experiment are also positively supported by teachers.

The contributions described in this thesis may be of great importance to the AIED research because they showed that we can design a flexible and usable authoring solution for gamified ITS that deals with the high variability and complexity inherent to the design of gamified ITS and taking advantage of theories and design practices, allowing teachers to intelligently design gamified tutors (combining human and artificial intelligence) – in few minutes – using different features for support authoring.

7.2.1 List of publications, papers under evaluation and papers to submit

In this section we list the publications that resulted from the development of this thesis. We also present the submitted papers that are currently under evaluation by the editorial board of journals and the papers that we intend to submit as soon as possible. During the development of this thesis, two international journal papers, one national conference paper and one doctoral consortium paper were published. Moreover, two journal papers are under evaluation; one of them (IJAIED) received a major review and the author of this thesis already submitted a reviewed version of paper that is the second-round review. The other paper received a minor review and we are still reviewing the paper to address reviewers' comments. Hereafter, we summarize our publications, the papers that are under evaluation and the papers we intend to submit.

Dermeval et al. [2014] – Brazilian Symposium on Software Engineering (SBES): A systematic review on the use of ontologies in requirements engineering

Dermeval et al. [2015a] – Expert Systems with Applications (ESWA): Ontology-based feature modeling: An empirical study in changing scenarios.

Dermeval et al. [2015b] – Requirements Engineering Journal (REEN): Applications of ontologies in requirements engineering: a systematic review of the literature

Dermeval [2016] – User Modelling, Adaptation and Personalization (UMAP): [3] Intelligent authoring of gamified intelligent tutoring systems

Major review on the International Journal of Artificial Intelligence in Education (IJAIED):

[3]The participation in the doctoral consortium track of this conference was of utmost importance to define the evaluation strategies of our authoring solution. The author of this thesis had the opportunity to discuss this thesis with important researchers in the topics targeted in this work such as Paul de Bra and Judith Masthoff.

Authoring tools for designing intelligent tutoring systems: a systematic review of the literature

Minor review on the International Journal on Knowledge and Learning (IJKL): An ontology-driven software product line architecture for developing gamified intelligent tutoring systems

To submit to IEEE Transactions on Learning Technologies (TLT): Towards an ontological model to apply gamification in intelligent tutoring systems

To submit to International Journal of Artificial Intelligence in Education (IJAIED): Authoring gamified intelligent tutoring systems

7.3 Limitations

In the development of this thesis, we could identify some limitations that may applied to our work. Note that, as previously mentioned, for each empirical study that we conducted to evaluate our contributions, we presented and discussed some possible threats to the validity of our results (see Sections 5.4.5, 6.3.5, and 6.4.5) and how we tried to mitigate those threats.

The reference feature model specified in this thesis includes several variation points and variants of features that could be included in gamified ITS configurations that were identified by the analysis of the literature and industrial gamified ITSs. In this way, the design of gamified ITS in the context of this thesis is limited to the features identified in that feature model, which is constrained: to a particular type of ITS based on curriculum sequencing (i.e., based on existing ITS theories [de Barros Costa et al., 1998, Dillenbourg and Self, 1992, Self, 1990, 1998]) and problem-based learning, to specific game design elements (e.g., badges, level, avatar, etc), to specific educational resources, and so on.

Another limitation of this work may be related to our OntoSPL ontology. Although we presented an ontology for representing feature models that is more flexible and requires less time to change than a well-known ontology for feature models (i.e., Wang et al. [2007]), we did not present a strategy for detecting automatic inconsistency in the configuration of products based on the OntoSPL. Thus, despite relying on the capabilities of ontologies for automatic detecting inconsistency, our ontology is still limited to provide this functionality.

Furthermore, our approach for identifying the evidence-supported behaviors and their respective set of game design elements was based on the manual investigation of empirical works included in three systematic reviews on the use of gamification in the context of e-learning. Thus, despite considering evidence reported by empirical works in the topic to support, based on these practices, the application of gamification to ITS, the practices identified are limited in time and scope to the works analyzed by the reviews. Indeed, there are mixed results on the use of gamification and more studies are required to identify in which circumstances gamification may be applied, including the conduction of theory-driven empirical studies [Nacke and Deterding, 2017]. Nevertheless, the behaviors identified in this thesis may provide a starting point for constraining the design space of gamified ITS based on the evidence reported by the literature. In addition, our ontological model is flexible enough to support redesign of game design elements when is needed to reconfigure a particular gamified ITS.

Our gamified tutoring ontology imports a gamification domain ontology that conceptualizes core and extended concepts about gamification based on particular gamification theoretical background (i.e., self-determination theory, 6D framework, and BrainHex player model). Moreover, it is also based on the ITS theories previously mentioned. Thus, the GaTO ontological model is tied to specific gamification and ITS theories and practices. However, one might note that the way we represented these concepts in the ontologies might favor their extension to support other theories, particularly, for the gamification context.

The main users of interest to our authoring solution are teachers. As such, the authoring computational solution is designed to enable them to actively personalize the gamification model of ITS according to their preferences in simple and usable ways. However, although this authoring solution is based on a conceptualization (GaTO ontology) that considers gamification theories and design practices that might benefit students, the gamification authored by our authoring solution is not fully personalized for students. Nevertheless, we include in our ontological model, concepts (i.e., player types and activity loops) that could further support the personalization of gamification for students.

Finally, we could not empirically evaluate how teachers perceive our authoring solution with respect to the domain model authoring of content and problems, which is also a limitation of this work that could be targeted in future works.

Thus, new efforts must be performed to extend our contributions in order mitigate some of these limitations aiming to achieve a balanced and effective way of combining teachers and artificial intelligence in the design of gamified intelligent tutoring systems.

7.4 Future works

In this section we describe further researches that could be investigated from the contributions presented in this thesis:

- Conduct more empirical studies aiming to reproduce the results of the previous experiments, including considering fewer factors and dependent variables;
- Empirically evaluate the domain model with no reuse option of the authoring process proposed for this authoring solution;
- Develop an integrated infrastructure that includes the authoring solution proposed in this thesis and a gamified ITS system that may reason on teachers' decisions to be reconfigured. This infrastructure would take advantage of human (from teachers) and artificial intelligence (provided by AI techniques, e.g., ontologies and machine learning as well as advanced software engineering techniques, for instance, dynamic software product lines and/or autonomic computing) to also provide adaptation of gamification to learners perspective. For instance, this infrastructure could be used to generate a gamified ITS according to teachers' preferences and, then, the system created could be capable to model learner's motivational (including his player type) and performance levels at the time they are interacting with the gamified tutor and to reconfigure the system with a different combination of game design elements or tutoring strategies that could improve the engagement and performance of students;
- Empirically investigate the effects of authored gamified tutors by teachers as well as the individual impact of the behaviors identified in this work and individual game design elements on the performance and motivation of students;

- Empirically investigate different authoring processes to support other configuration options for teacher (e.g., interface authoring, gamification loops, and so on) in terms of flexibility and usability;

- Propose and investigate authoring solutions for the gamified ITS life-cycle. This solution would enable the active participation of teachers along with artificial intelligence throughout the gamified tutor life-cycle, from the beginning of an ITS design (pre-instruction) and at later stages of the execution of the tutor (i.e., during instruction and post-instruction);

- Investigate and evaluate the use of persuasive strategies in the authoring solution graphical interface (beyond tunneling) to aid teachers completing authoring tasks, for instance, using the persuasive design principles presented by Oinas-Kukkonen and Harjumaa [2009] (e.g., reduction, tailoring, and so on);

- Register the software developed in this thesis and investigate the potential of this solution as a technological and scientific innovation.

Bibliography

Muhammad Aeem Abbas, Wan Fatimah Wan Ahmad, and Khairul Shafee Kalid. Semantic Web Technologies for Pre-School Cognitive Skills Tutoring System. *Journal of information science and engineering*, 30(3):835–851, 2014. ISSN 1016-2364.

Philip Achimugu, Ali Selamat, Roliana Ibrahim, and Mohd Naz'ri Mahrin. A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(6):568–585, 2014. ISSN 0950-5849.

Maha Al-Yahya, Remya George, and Auhood Alfaries. Ontologies in e-learning: review of the literature. *Int. J. Softw. Eng. Appl*, 9(2):67–84, 2015.

Clayton P Alderfer. An empirical test of a new theory of human needs. *Organizational behavior and human performance*, 4(2):142–175, 1969.

Efthimios Alepis and Maria Virvou. Mobile Authoring in Educational Software. *Object-Oriented User Interfaces for Personalized Mobile Learning*, 64:31–46, 2014. ISSN 18684408.

V Aleven, B M McLaren, and J Sewall. Scaling Up Programming by Demonstration for Intelligent Tutoring Systems Development: An Open-Access Web Site for Middle School Mathematics Learning. *Learning Technologies, IEEE Transactions on*, 2(2):64–78, 2009a.

Vincent Aleven, Bruce M McLaren, Jonathan Sewall, and Kenneth R Koedinger. The cognitive tutor authoring tools (ctat): preliminary evaluation of efficiency gains. In *Intelligent Tutoring Systems*, pages 61–70. Springer, 2006.

Vincent Aleven, Bruce M Mclaren, Jonathan Sewall, and Kenneth R Koedinger. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2):105–154, 2009b.

Vincent Aleven, Bruce M. McLaren, Jonathan Sewall, Martin van Velsen, Octav Popescu, Sandra Demi, Michael Ringenberg, and Kenneth R. Koedinger. Example-tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education*, 26(1):224–269, 2016. ISSN 1560-4306.

John R Anderson. *The architecture of cognition*. Harvard University Press, 1983.

Fernando R. H. Andrade, Riichiro Mizoguchi, and Seiji Isotani. *The Bright and Dark Sides of Gamification*, pages 176–186. Springer International Publishing, Cham, 2016. ISBN 978-3-319-39583-8. doi: 10.1007/978-3-319-39583-8_17. URL http://dx.doi.org/10.1007/978-3-319-39583-8_17.

Phillip G. Armour. Software: Hard data. *Commun. ACM*, 49(9):15–17, September 2006. ISSN 0001-0782. doi: 10.1145/1151030.1151043. URL http://doi.acm.org/10.1145/1151030.1151043.

Ivon Arroyo, Kimberly Ferguson, Jeffrey Johns, Toby Dragon, Hasmik Meheranian, Don Fisher, Andrew Barto, Sridhar Mahadevan, and Beverly Park Woolf. Repairing disengagement with non-invasive interventions. In *AIED*, volume 2007, pages 195–202, 2007.

Mohsen Asadi, Dragan Gasevic, Yair Wand, and Marek Hatala. Deriving variability patterns in software product lines by ontological considerations. In Paolo Atzeni, David Cheung, and Sudha Ram, editors, *Conceptual Modeling*, volume 7532 of *Lecture Notes in Computer Science*, pages 397–408. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-34001-7. doi: 10.1007/978-3-642-34002-4_31. URL http://dx.doi.org/10.1007/978-3-642-34002-4_31.

Franz Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003.

Ebrahim Bagheri, Mohsen Asadi, Faezeh Ensan, Dragan Gašević, and Bardia Mohabbati. Bringing semantics to feature models with safmdl. In *Proceedings of the 2011 Conference* *of the Center for Advanced Studies on Collaborative Research*, CASCON '11, pages 287–300, Riverton, NJ, USA, 2011. IBM Corp.

Ryan S Baker. Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2):600–614, 2016.

Ryan SJd Baker, Sidney K D'Mello, Ma Mercedes T Rodrigo, and Arthur C Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4):223–241, 2010.

Judson Bandeira, Ig Ibert Bittencourt, Patricia Espinheira, and Seiji Isotani. Foca: A methodology for ontology evaluation. *arXiv preprint arXiv:1612.03353*, 2016.

Barron-Estrada, Maria Lucia, Ramon Zatarain-Cabada, Rosalio Zatarain-Cabada, Hector Barbosa-Leon, and Carlos A Reyes-Garcia. Building and Assessing Intelligent Tutoring Systems with an e-Learning 2 . 0 Authoring System. *Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, pages 1–9, 2010.

M.L.a Barrón-Estrada, R.a Zatarain-Cabada, P.a Tamayo, S.a Tamayo, and H.b Pérez-Espinoza. A learning social network with multi-modal affect. In *Proceedings of the 10th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence and Applications, MICAI 2011*, pages 163–168, 2011.

Richard Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *Journal of MUD research*, 1(1):19, 1996.

Victor R Basili. Software modeling and measurement: the goal/question/metric paradigm. 1992.

Chris Bateman and Richard Boon. *21st Century Game Design (Game Development Series)*. Charles River Media, Inc., 2005.

Chris Bateman, Rebecca Lowenhaupt, and Lennart E Nacke. Player typology in theory and practice. In *Proceedings of DiGRA*, pages 1–24, 2011.

María Rosario Bautista-Zambrana. Methodologies to build ontologies for terminological purposes. *Procedia-Social and Behavioral Sciences*, 173:264–269, 2015.

Joseph Beck, Mia Stern, and Erik Haugsjaa. Applications of ai in education. *Crossroads*, 3 (1):11–15, 1996.

Courtney Bell and Danielle S McNamara. Integrating istart into a high school curriculum. In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, pages 809–814. Cognitive Science Society Austin, TX, 2007.

David Benavides, Sergio Segura, and Antonio Ruiz-Cortês. Automated analysis of feature models 20 years later: A literature review. *Information Systems*, 35(6):615 – 636, 2010. ISSN 0306-4379. doi: http://dx.doi.org/10.1016/j.is.2010.01.001. URL `http://www.sciencedirect.com/science/article/pii/S0306437910000025`.

David Benavides, Alexander Felfernig, JoséA. Galindo, and Florian Reinfrank. Automated analysis in feature modelling and product configuration. In John Favaro and Maurizio Morisio, editors, *Safe and Secure Software Reuse*, volume 7925 of *Lecture Notes in Computer Science*, pages 160–175. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38976-4. doi: 10.1007/978-3-642-38977-1_11. URL `http://dx.doi.org/10.1007/978-3-642-38977-1\_11`.

Shlomo Berkovsky, Mac Coombe, Jill Freyne, Dipak Bhandari, and Nilufar Baghaei. Physical activity motivating games: Virtual rewards for real activity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 243–252, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753362. URL `http://doi.acm.org/10.1145/1753326.1753362`.

Ig Ibert Bittencourt, Evandro Costa, Marlos Silva, and Elvys Soares. A computational model for developing semantic web-based educational systems. *Knowledge-Based Systems*, 22 (4):302 – 315, 2009. ISSN 0950-7051. doi: http://dx.doi.org/10.1016/j.knosys.2009.02.012. URL `http://www.sciencedirect.com/science/article/pii/S0950705109000501`. Artificial Intelligence (AI) in Blended Learning(AI) in Blended Learning.

S B Blessing, S B Gilbert, S Ourada, and S Ritter. Authoring model-tracing cognitive tutors. *International Journal of Artificial Intelligence in Education*, 19(2):189–210, 2009.

Stephen B Blessing, Shrenik Devasani, Stephen B Gilbert, and Jivko Sinapov. Using conceptgrid as an easy authoring technique to check natural language responses. *International Journal of Learning Technology*, 10(1):50–70, 2015.

Marko Bošković, Ebrahim Bagheri, Dragan Gašević, Bardia Mohabbati, Nima Kaviani, and Marek Hatala. Automated staged configuration with semantic web technologies. *International Journal of Software Engineering and Knowledge Engineering*, 20(04): 459–484, 2010.

Keith W. Brawner. Rapid dialogue and branching tutors. In *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education, AIED 2015, Madrid, Spain, June 22 + 26, 2015.*, 2015.

Santi Caballé and Fatos Xhafa. Clpl: Providing software infrastructure for the systematic and effective construction of complex collaborative learning systems. *Journal of Systems and Software*, 83(11):2083–2097, 2010.

Coral Calero, Francisco Ruiz, and Mario Piattini. *Ontologies for software engineering and software technology*. Springer Science & Business Media, 2006.

S Chakraborty, D Roy, P Kumar Bhowmick, and A Basu. An authoring system for developing Intelligent Tutoring System. In *Proceedings of the IEE Students' Technology Symposium (TechSym)*, pages 196–205, 2010.

Geiser Chalco Challco, Dilvan Moreira, Riichiro Mizoguchi, and Seiji Isotani. *Towards an Ontology for Gamifying Collaborative Learning Scenarios*, pages 404–409. Springer International Publishing, Cham, 2014. ISBN 978-3-319-07221-0. doi: 10.1007/978-3-319-07221-0_50. URL `http://dx.doi.org/10.1007/978-3-319-07221-0_50`.

Lianipng Chen, Muhammad Ali Babar, and He Zhang. Towards an evidence-based understanding of electronic data sources. In *Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering*, EASE'10, pages

135–138, Swinton, UK, UK, 2010. British Computer Society. URL `http://dl.acm.org/citation.cfm?id=2227057.2227074`.

Christopher Cheong, France Cheong, and Justin Filippou. Quick quiz: A gamified approach for enhancing learning. In *PACIS*, page 206, 2013.

Vincent Cho, TC Edwin Cheng, and WM Jennifer Lai. The role of perceived user-interface design in continued usage intention of self-paced e-learning tools. *Computers & Education*, 53(2):216–227, 2009.

C Y Chou, B H Huang, and C J Lin. Complementary machine intelligence and human intelligence in virtual teaching assistant for tutoring program tracing. *Computers and Education*, 57(4):2303–2312, 2011.

Paul C. Clements and Linda Northrop. *Software Product Lines: Practices and Patterns*. SEI Series in Software Engineering. Addison-Wesley, August 2001.

Albert T Corbett and John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & knowledge engineering*, 46(1):41–64, 2003.

Krzysztof Czarnecki, Chang Hwan Peter Kim, and Karl Trygve Kalleberg. Feature models are views on ontologies. In *Proceedings of the 10th International on Software Product Line Conference*, SPLC '06, pages 41–51, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2599-7. URL `http://dl.acm.org/citation.cfm?id=1158337.1158678`.

Funda Dağ, Levent Durdu, and Serpil Gerdan. Evaluation of educational authoring tools for teachers stressing of perceived usability features. *Procedia-Social and Behavioral Sciences*, 116:888–901, 2014.

Danilo L Dalmon, Leônidas O Brandão, Anarosa AF Brandão, Seiji Isotani, et al. A domain engineering for interactive learning modules. *Journal of Research and Practice in Information Technology*, 44(3):309, 2012.

Robertas Damaševičius. Specification of learning content using feature diagrams. In *Information Systems Development*, pages 821–829. Springer, 2010.

Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI magazine*, 14(1):17, 1993.

Evandro de Barros Costa, Angelo Perkusich, and Edilson Ferneda. From a tridimensional view of domain knowledge to multi-agent tutoring system. In *Brazilian Symposium on Artificial Intelligence*, pages 61–72. Springer, 1998.

Sergio de Cesare, Guido L. Geerts, Grant Holland, Mark Lycett, and Chris Partridge. Ontology-driven software engineering. In *Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications*, OOPSLA '09, pages 723–724, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-768-4. doi: 10.1145/1639950.1639983. URL `http://doi.acm.org/10.1145/1639950.1639983`.

Simone de Sousa Borges, Vinicius HS Durelli, Helena Macedo Reis, and Seiji Isotani. A systematic mapping on gamification applied to education. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 216–222. ACM, 2014.

EL Deci and RM Ryan. Self-determination. the corsini encyclopedia of psychology hoboken, 2010.

Paul Denny. The effect of virtual achievements on student engagement. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 763–772. ACM, 2013.

Diego Dermeval. Intelligent authoring of gamified intelligent tutoring systems. *Late-breaking Results, Posters, Demos, Doctoral Consortium and Workshops Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalisation (UMAP)*, 2016.

Diego Dermeval, Jessyka Vilela, Ig Ibert Bittencourt, Jaelson Castro, Seiji Isotani, and Patrick Brito. A systematic review on the use of ontologies in requirements engineering. In *Brazilian Symposium on Software Engineering (SBES)*, pages 1–10. IEEE, 2014.

Diego Dermeval, Thyago Tenório, Ig Ibert Bittencourt, Alan Silva, Seiji Isotani, and Márcio Ribeiro. Ontology-based feature modeling: An empirical study in changing scenarios. *Expert Systems with Applications*, 42(11):4950 – 4964, 2015a. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2015.02.020. URL `http://www.sciencedirect.com/science/article/pii/S0957417415001190`.

Diego Dermeval, Jéssyka Vilela, Ig Ibert Bittencourt, Jaelson Castro, Seiji Isotani, Patrick Brito, and Alan Silva. Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering*, pages 1–33, 2015b.

Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15. ACM, 2011.

S.a Devasani, S.B.a Gilbert, and S.B.b Blessing. Evaluation of two intelligent tutoring system authoring tool paradigms graphical user interface-based and text-based. In *Proceedings of the 21st Annual Conference on Behavior Representation in Modeling and Simulation, BRiMS 2012*, pages 51–58, 2012.

David Díez, Alessio Malizia, Ignacio Aedo, Paloma Díaz, Camino Fernández, and Juan-Manuel Dodero. A methodological approach to encourage the service-oriented learning systems development. *Journal of Educational Technology & Society*, 12(4): 138–148, 2009.

Pierre Dillenbourg and John Self. A framework for learner modelling. *Interactive Learning Environments*, 2(2):111–137, 1992.

Wei Ding, Peng Liang, Antony Tang, and Hans van Vliet. Knowledge-based approaches in software documentation: A systematic literature review. *Information and Software Technology*, 56(6):545–567, 2014. ISSN 0950-5849.

Adrián Domínguez, Joseba Saenz-De-Navarrete, Luis De-Marcos, Luis Fernández-Sanz, Carmen Pagés, and José-Javier Martínez-HerráIz. Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63:380–392, 2013.

Benedict du Boulay. Recent meta-reviews and meta–analyses of aied systems. *International Journal of Artificial Intelligence in Education*, 26(1):536–537, 2016. ISSN 1560-4306.

Benedict Du Boulay and Rosemary Luckin. Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education*, 12(3): 235–256, 2001.

Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10):833–859, 2008. ISSN 0950-5849.

Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In Forrest Shull, Janice Singer, and DagI.K. Sjøberg, editors, *Guide to Advanced Empirical Software Engineering*, pages 285–311. Springer London, 2008. ISBN 978-1-84800-043-8.

Economist. Why video games are so expensive to develop. Disponível em http://www.economist.com/blogs/economist-explains/2014/09/economist-explains-15, September 2014. Acesso em 17 de Julho de 2015.

H Escudero and R Fuentes. Exchanging courses between different Intelligent Tutoring Systems: A generic course generation authoring tool. *Knowledge-Based Systems*, 23(8): 864–874, 2010.

Richard M Felder and Linda K Silverman. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.

Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. *AAAI-97 Spring Symposium Series*, 1997.

João Bosco Ferreira Filho, Olivier Barais, Benoit Baudry, Windson Viana, and Rossana M. C. Andrade. An approach for semantic enrichment of software product lines. In *Proceedings of the 16th International Software Product Line Conference - Volume 2*, SPLC '12, pages 188–195, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1095-6. doi:

10.1145/2364412.2364444. URL `http://doi.acm.org/10.1145/2364412.2364444`.

Zachary Fitz-Walter, Dian Tjondronegoro, and Peta Wyeth. Orientation passport: using gamification to engage university students. In *Proceedings of the 23rd Australian computer-human interaction conference*, pages 122–125. ACM, 2011.

Zachary Fitz-Walter, Dian Tjondronegoro, and Peta Wyeth. A gamified mobile application for engaging new students at university orientation. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, pages 138–141. ACM, 2012.

Brian J Fogg. Persuasive technologies. *Communications of the ACM*, 42(5):26–29, 1999.

Brian J Fogg. Persuasive technology: using computers to change what we think and do. *Ubiquity*, 2002(December):5, 2002.

Brian J Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, page 40. ACM, 2009.

Jason A Foster, Patricia Kristine Sheridan, Robert Irish, and Geoffrey Samuel Frost. Gamification as a strategy for promoting deeper investigation in a reverse engineering activity. In *American Society for Engineering Education*. American Society for Engineering Education, 2012.

R Fox, M Schleifer, and J Cellio. A tool to generate computer assisted instruction systems through hierarchical classification. In *Proceedings of the 2011 International Conference on Artificial Intelligence, ICAI 2011*, volume 1, pages 385–391, 2011.

Marylène Gagné and Edward L Deci. Self-determination theory and work motivation. *Journal of Organizational behavior*, 26(4):331–362, 2005.

Aldo Gangemi, Geri Steve, and Fabrizio Giacomelli. Onions: An ontological methodology for taxonomic knowledge integration. In *ECAI-96 Workshop on Ontological Engineering, Budapest*, 1996.

Dragan Gašević, Nima Kaviani, and Milan Milanović. Ontologies and software engineering. In *Handbook on Ontologies*, pages 593–615. Springer, 2009.

S. Ghaisas and N. Ajmeri. Knowledge-assisted ontology-based requirements evolution. In Walid Maalej and Anil Kumar Thurimella, editors, *Managing Requirements Knowledge*, pages 143–167. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-34418-3. doi: 10.1007/978-3-642-34419-0_7. URL http://dx.doi.org/10.1007/978-3-642-34419-0_7.

S. B.; Gilbert, S. B.; Blessing, and E. Guo. Authoring Effective Embedded Tutors: An Overview of the Extensible Problem Specific Tutor (xPST) System. *International Journal of Artificial Intelligence in Education*, NA, 2015.

Frank G Goble. *The third force: The psychology of Abraham Maslow*. Maurice Bassett, 2004.

Geoff Goehle. Gamification and web-based homework. *Primus*, 23(3):234–246, 2013.

Asunción Gómez-Pérez. Towards a framework to verify knowledge sharing technology. *Expert Systems with Applications*, 11(4):519–529, 1996.

Carina González, Alberto Mora, and Pedro Toledo. Gamification in intelligent tutoring systems. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM '14, pages 221–225, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2896-8. doi: 10.1145/2669711.2669903. URL http://doi.acm.org/10.1145/2669711.2669903.

Tony Gorschek, Claes Wohlin, Philippe Carre, and Stig Larsson. A model for technology transfer in practice. *Software, IEEE*, 23(6):88–95, 2006.

Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993. ISSN 1042-8143.

Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, December 1995. ISSN 1071-5819.

A. Grubisic, S. Stankov, and V. Glavinic. Agent based intelligent courseware generation in e-learning. *Proceeding of the IASTED International Conference on Computers and Advanced Technology in Education (CATE)*, 2009.

Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

Nathalie Guin and Marie Lefevre. From a customizable ITS to an adaptive ITS. *International Conference on Artificial Intelligence in Education (AIED)*, 7926 LNAI:141–150, 2013. ISSN 03029743.

Jianmei Guo, Yinglin Wang, Pablo Trinidad, and David Benavides. Consistency maintenance for evolving feature models. *Expert Systems with Applications*, 39(5):4987–4998, 2012. ISSN 0957-4174.

Christian Gütl. Moving towards a generic, service-based architecture for flexible teaching and learning activities. *Architecture Solutions for E-Learning Systems (peerreviewed), Peer-reviewed book chapter, Idea Group Inc., Hershey, USA*, pages 1–24, 2007.

Lasse Hakulinen, Tapio Auvinen, and Ari Korhonen. Empirical study on the effect of achievement badges in trakla2 online learning environment. In *Learning and Teaching in Computing and Engineering (LaTiCE), 2013*, pages 47–54. IEEE, 2013.

Shivashankar Halan, Brent Rossen, Juan Cendan, and Benjamin Lok. High score!-motivation strategies for user participation in virtual human development. In *International Conference on Intelligent Virtual Agents*, pages 482–488. Springer, 2010.

Juho Hamari and Janne Tuunanen. Player types: A meta-synthesis. *Transactions of the Digital Games Research Association*, 1(2), 2014.

Juho Hamari, Jonna Koivisto, and Tuomas Pakkanen. Do persuasive technologies persuade?-a review of empirical studies. In *Persuasive Technology*, pages 118–136. Springer, 2014a.

Juho Hamari, Jonna Koivisto, and Harri Sarsa. Does gamification work?–a literature review of empirical studies on gamification. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3025–3034. IEEE, 2014b.

Hans-Jörg Happel and Stefan Seedorf. Applications of ontologies in software engineering. In *Proc. of Workshop on Sematic Web Enabled Software Engineering"(SWESE) on the ISWC*, pages 5–9. Citeseer, 2006.

Neil T HeffernanCristina Lindquist Heffernan. The ASSISTments Ecosystem: Building a Platform that Brings Scientists and Teachers Together for Minimally Invasive Research on Human Learning and Teaching. *International Journal of Artificial Intelligence in Education*, 24(4), 2014.

Andreas Helferich, Georg Herzwurm, Stefan Jesse, and Martin Mikusz. Software product lines, service-oriented architecture and frameworks: Worlds apart or ideal partners? In Dirk Draheim and Gerald Weber, editors, *Trends in Enterprise Application Architecture*, volume 4473 of *Lecture Notes in Computer Science*, pages 187–201. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-75911-9. doi: 10.1007/978-3-540-75912-6_14. URL `http://dx.doi.org/10.1007/978-3-540-75912-6\_14`.

Martin Hepp, Pieter De Leenheer, Aldo De Moor, and York Sure. *Ontology management: semantic web, semantic web services, and business applications*. Springer Science & Business Media, 2007.

Elis Montoro Hernandes, Augusto Zamboni, Sandra Fabbri, and André Di Thommazo. Using gqm and tam to evaluate start - a tool that supports systematic review. *CLEI Electron. J.*, 15(1), 2012.

Luis A Hernández Ibáñez and Viviana Barneche Naya. Joint spaces between schools and museums via virtual worlds: a case study. In *Proceedings of the 2012 ACM workshop on User experience in e-learning and augmented technologies in education*, pages 19–24. ACM, 2012.

Pieter Heyvaert, Ruben Verborgh, Erik Mannens, and Rik Van de Walle. Linked data-enabled gamification in epub 3 for educational digital textbooks. In *Design for Teaching and Learning in a Networked World*, pages 587–591. Springer, 2015.

Olavo Holanda, Seiji Isotani, Ig Ibert Bittencourt, Endhe Elias, and Thyago Tenório. Joint: Java ontology integrated toolkit. *Expert Systems with Applications*, 40(16): 6469 – 6477, 2013. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2013.05.040. URL `http://www.sciencedirect.com/science/article/pii/S0957417413003382`.

Heather Holden and Roy Rada. Understanding the influence of perceived usability and technology self-efficacy on teachers' technology acceptance. *Journal of Research on Technology in Education*, 43(4):343–367, 2011.

Kai Huotari and Juho Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th International Academic MindTrek Conference*, pages 17–22. ACM, 2012.

Seiji Isotani, Ig Ibert Bittencourt, Ellen Francine Barbosa, Diego Dermeval, and Ranilson Oscar Araujo Paiva. Ontology driven software engineering: A review of challenges and opportunities. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 13(3): 863–869, 2015.

G Tanner Jackson and Danielle S McNamara. Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology*, 105(4):1036, 2013.

Zhenhui Jiang, Weiquan Wang, Bernard CY Tan, and Jie Yu. The determinants and impacts of aesthetics in users' first interaction with websites. *Journal of Management Information Systems*, 33(1):229–259, 2016.

Jesper Juul. The game, the player, the world: Looking for a heart of gameness. *PLURAIS-Revista Multidisciplinar da UNEB*, 1(2), 2010.

K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon University Software Engineering Institute, November 1990.

Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

N. Kaviani, B. Mohabbati, D. Gasevic, , and M. Finke. Semantic annotations of feature models for dynamic product conguration in ubiquitous environments. In *4th International Workshop on Semantic Web Enabled Software Engineering at 7th International Semantic Web Conference.*, 2008.

Phillip King and Jason Tester. The landscape of persuasive technologies. *Communications of the ACM*, 42(5):31–38, 1999.

Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

Kenneth R Koedinger and Vincent Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

Pål Kraft, Harald Schjelderup-Lund, and Håvar Brendryen. Digital therapy: The coming together of psychology and technology can create a new generation of programs for more sustainable behavioral change. In *International Conference on Persuasive Technology*, pages 18–23. Springer, 2007.

James A Kulik and JD Fletcher. Effectiveness of intelligent tutoring systems a meta-analytic review. *Review of Educational Research*, page 0034654315581420, 2015.

Richard N Landers and Rachel C Callan. Casual social games as serious games: The psychology of gamification in undergraduate education and employee training. In *Serious games and edutainment applications*, pages 399–423. Springer, 2011.

H. Chad Lane, Mark G. Core, Matthew J. Hays, Daniel Auerbach, and Milton Rosenberg. *Situated Pedagogical Authoring: Authoring Intelligent Tutors from a Student's Perspective*, pages 195–204. Springer International Publishing, Cham, 2015. ISBN 978-3-319-19773-9.

LAPES. Start – state of the art through systematic review tool. Available in http://lapes.dc.ufscar.br/tools/start_tool, 2014. Accessed in October, 2013.

Soon-Bok Lee, Jin-Woo Kim, Chee-Yang Song, and Doo-Kwon Baik. An approach to analyzing commonality and variability of features using ontology in a software product line engineering. In *Software Engineering Research, Management Applications, 2007. SERA 2007. 5th ACIS International Conference on*, pages 727–734, Aug 2007. doi: 10.1109/SERA.2007.41.

Fundação Lemann. Conselho de classe: A visão dos professores sobre a educação no brasil. Disponível em http://fundacaolemann.org.br/novidades/a-visao-dos-professores-sobre-a-educacao-no-brasil, 2015. Acesso em 23 de Julho de 2015.

Wei Li, Tovi Grossman, and George Fitzmaurice. Gamicad: a gamified tutorial system for first time autocad users. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 103–112. ACM, 2012.

Miltiadis D Lytras, Miguel Angel Sicilia, Demetrios Sampson, Victor Pankratius, Wolffried Stucky, and Gottfried Vossen. Aspect-oriented re-engineering of e-learning courseware. *The Learning Organization*, 12(5):457–470, 2005.

Wenting Ma, Olusola O Adesope, John C Nesbit, and Qing Liu. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, pages 901–918, 2014. URL `http://psycnet.apa.org/doi/10.1037/a0037123`.

C. J. MacLellan, K. R. Koedinger, and N. Matsuda. Authoring Tutors with SimStudent: An Evaluation of Efficiency and Model Quality. *Intelligent Tutoring Systems*, NA, 2014.

Christopher J. MacLellan, Erik Harpstead, Eliane Stampfer Wiese, Mengfan Zou, Noboru Matsuda, Vincent Aleven, and Kenneth R. Koedinger. Authoring tutors with complex solutions: A comparative analysis of example tracing and simstudent. In *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education, AIED 2015, Madrid, Spain, June 22 + 26, 2015.*, 2015.

Sara Mahdavi-Hezavehi, Matthias Galster, and Paris Avgeriou. Variability in quality attributes of service-based software systems: A systematic literature review. *Information and Software Technology*, 55(2):320 – 343, 2013. ISSN 0950-5849. Special Section: Component-Based Software Engineering (CBSE), 2011.

N Marcus, D Ben-Naim, and M Bain. Instructional support for teachers and guided feedback for students in an adaptive elearning environment. In *Proceedings of the 8th International Conference on Information Technology: New Generations, ITNG 2011*, pages 626–631, 2010.

José Martí-Parreño, Diana Seguí-Mas, and Elies Seguí-Mas. Teachers' attitude towards and actual use of gamification. *Procedia-Social and Behavioral Sciences*, 228:682–688, 2016.

Judith Masthoff and Julita Vassileva. Tutorial on personalization for behaviour change. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 439–442, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3306-1. doi: 10.1145/2678025.2716264. URL http://doi.acm.org/10.1145/2678025.2716264.

N. Matsuda, W. W. Cohen, and K. R. Koedinger. Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring. *International Journal of Artificial Intelligence in Education*, 25(1), 2015.

Jane McGonigal. Jogando por um mundo melhor. Available at http://www.ted.com/talks/jane_mcgonigal_gaming_can_make_a_better_world/transcript?language=pt-br, March 2010. Acesso em 17 de Julho de 2015.

D. L. McGuinness and F. V. Harmelen. Owl web ontology language overview. URL: http://www.w3.org/TR/owl-features/, February 2004.

Susan Michie, Marie Johnston, Jill Francis, Wendy Hardeman, and Martin Eccles. From theory to intervention: mapping theoretically derived behavioural determinants to behaviour change techniques. *Applied psychology*, 57(4):660–680, 2008.

P. Mika and H. Akkermans. Towards a new synthesis of ontology technology and knowledge management. *KNOWLEDGE ENGINEERING REVIEW*, 19:317–345, 2004.

Antonija Mitrovic, Brent Martin, Pramuditha Suraweera, K Zakharov, N Milik, J Holland, and N McGuigan. ASPIRE : An Authoring System and Deployment Environment for Constraint-Based Tutors. *International Journal of Artificial Intelligence in Education*, 19: 155–188, 2009. ISSN ISSN-1560-4292.

Jonas A Montilva, Beatriz Sandia, and Judith Barrios. Developing instructional web sites–a software engineering approach. *Education and Information Technologies*, 7(3):201–224, 2002.

Alberto Mora, Daniel Riera, Carina González, and Joan Arnedo-Moreno. A literature review of gamification design frameworks. In *Games and Virtual Worlds for Serious Applications (VS-Games), 2015 7th International Conference on*, pages 1–8. IEEE, 2015.

Thomas Murray, Stephen Blessing, and Shaaron Ainsworth. *Authoring tools for advanced technology learning environments: Toward cost-effective adaptive, interactive and intelligent educational software*. Springer Science & Business Media, 2003.

Tom Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

Tom Murray. An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In *Authoring tools for advanced technology learning environments*, pages 491–544. Springer, 2003.

Tom Murray. Design tradeoffs in usability and power for advanced educational software authoring tools. *EDUCATIONAL TECHNOLOGY-SADDLE BROOK THEN ENGLEWOOD CLIFFS NJ-*, 44(5):10–16, 2004.

I.M. Murwantara. Hybrid anp: Quality attributes decision modeling of a product line architecture design. In *Uncertainty Reasoning and Knowledge Engineering (URKE), 2012 2nd International Conference on*, pages 30–34, Aug 2012. doi: 10.1109/URKE.2012.6319572.

Lennart E Nacke and Sebastian Deterding. The maturing of gamification research, 2017.

Lennart E. Nacke, Chris Bateman, and Regan L. Mandryk. Brainhex: A neurobiological gamer typology survey. *Entertainment Computing*, 5(1):55 – 62, 2014. ISSN 1875-9521. doi: http://dx.doi.org/10.1016/j.entcom.2013.06.002. URL `http://www.sciencedirect.com/science/article/pii/S1875952113000086`.

Roger Nkambou. *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media, 2010.

Mahdi Noorian, Alireza Ensan, Ebrahim Bagheri, Harold Boley, and Yevgen Biletskiy. Feature model debugging based on description logic reasoning. In *Proceedings of the 17th International Conference on Distributed Multimedia Systems, DMS 2011, October 18-20, 2011, Convitto della Calza, Florence, Italy*, pages 158–164, 2011.

Andreas Oberweis, Victor Pankratius, and Wolffried Stucky. Product lines for digital information products. *Information Systems*, 32(6):909 – 939, 2007. ISSN 0306-4379. doi: http://dx.doi.org/10.1016/j.is.2006.09.003. URL `http://www.sciencedirect.com/science/article/pii/S0306437906000809`.

Leo Obrst, Werner Ceusters, Inderjeet Mani, Steve Ray, and Barry Smith. The evaluation of ontologies. In *Semantic web*, pages 139–158. Springer, 2007.

Harri Oinas-Kukkonen and Marja Harjumaa. Persuasive systems design: Key issues, process model, and system features. *Communications of the Association for Information Systems*, 24(1):28, 2009.

Andrew M Olney and Whitney L Cade. Authoring intelligent tutoring systems using human computation: Designing for intrinsic motivation. In *International Conference on Augmented Cognition*, pages 628–639. Springer, 2015.

Jennifer K Olsen, Daniel M Belenky, Vincent Aleven, Nikol Rummel, Jonathan Sewall, and Michael Ringenberg. Authoring Tools for Collaborative Intelligent Tutoring System Environments. *Intelligent Tutoring Systems*, NA, 2014.

Jeff Z Pan, Steffen Staab, Uwe Aßmann, Jürgen Ebert, and Yuting Zhao. *Ontology-driven software development*. Springer Science & Business Media, 2012.

L. Paquette, J. F. Lebeau, and A. Mayers. Authoring Problem-Solving Tutors: A Comparison between ASTUS and CTAT. *Advances in Intelligent Tutoring Systems*, NA, 2010.

Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of sparql. *ACM Trans. Database Syst.*, 34(3):16:1–16:45, September 2009. ISSN 0362-5915.

Klaus Pohl, Günter Böckle, and Frank J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 3540243720.

ProjectTomorrow. Learning in the 21st century: Digital experiences and expectations of tomorrow's teachers. Disponível em http://www.tomorrow.org/speakup/tomorrowsteachers_report2013.html, 2013. Acesso em 23 de Julho de 2015.

ProjectTomorrow. Speak up 2014 research project findings - the results of the authentic, unfiltered views of 41,805 k-12 teachers nationwide. Disponível em http://www.tomorrow.org/speakup/pdfs/SU2014_TeacherTop10.pdf, 2014. Acesso em 23 de Julho de 2015.

Dimitrios SklavakisIoannis Refanidis. The MATHESIS Semantic Authoring Framework: Ontology-Driven Knowledge Engineering for ITS Authoring. *Knowlege-Based and Intelligent Information and Engineering Systems*, NA, 2011.

Francisco Ruiz and José R Hilera. Using ontologies in software engineering and technology. In *Ontologies for software engineering and software technology*, pages 49–102. Springer, 2006.

Katie Salen and Eric Zimmerman. *Rules of play: Game design fundamentals*. MIT press, 2004.

Pablo Sanchez Barreiro, Diego Garcia-Saiz, and Marta Elena Zorrilla Pantaleon. Building families of software products for e-learning platforms: A case study. *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, 9(2):64–71, 2014.

Antonio Sánchez-Mena and José Martí-Parreño. Gamification in higher education: Teachers' drivers and barriers. In *Conference Proceedings. The Future of Education*, page 180. libreriauniversitaria. it Edizioni, 2016.

Leonardo Santos, Alberto Castro, and Crediné Silva de Menezes. Flexible virtual environments for teaching and learning. In *Frontiers in Education Conference (FIE), 2012*, pages 1–6. IEEE, 2012.

Robert Schuwer and Rob Kusters. Mass customization of education by an institution of he: What can we learn from industry? *The International Review of Research in Open and Distributed Learning*, 15(2), 2014.

Katie Seaborn and Deborah I Fels. Gamification in theory and action: A survey. *International Journal of Human-Computer Studies*, 74:14–31, 2015.

John Self. Theoretical foundations for intelligent tutoring systems. *Journal of Artificial Intelligence in Education*, 1(4):3–14, 1990.

John Self. The defining characteristics of intelligent tutoring systems research: Itss care, precisely. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10: 350–364, 1998.

Brian Shackel. Usability-context, framework, definition, design and evaluation. *Human factors for informatics usability*, pages 21–37, 1991.

Lei Shi and Alexandra I. Cristea. *Motivational Gamification Strategies Rooted in Self-Determination Theory for Social Adaptive E-Learning*, pages 294–300. Springer International Publishing, Cham, 2016. ISBN 978-3-319-39583-8. doi: 10.1007/978-3-319-39583-8_32. URL `http://dx.doi.org/10.1007/978-3-319-39583-8_32`.

Timothy K Shih, Chin-Chen Chang, and Hsiau Wen Lin. Reusability on learning object repository. In *Advances in Web Based Learning–ICWL 2006*, pages 203–214. Springer, 2006.

Valerie J Shute and Joseph Psotka. Intelligent tutoring systems: Past, present, and future. Technical report, DTIC Document, 1994.

Alan Silva, Evandro Costa, and Ig Ibert Bittencourt. Uma linha de produto de software baseada na web semântica para sistemas tutores inteligentes. *Revista Brasileira de Informática na Educação*, 20(1):87, 2012.

Alan Pedro Silva, Ig Ibert Bittencourt, Walker Ataíde, Olavo Holanda, Evandro Costa, Thyago Tenório, and Patrick HS Brito. An ontology-based model for driving the building of software product lines in an its context. In *Third International Conference on Software, Services and Semantic Technologies S3T 2011*, pages 155–159. Springer, 2011.

Burrhus Frederic Skinner. *About behaviorism*. Vintage, 2011.

Derek Sleeman and John Seely Brown. *Intelligent tutoring systems*. London: Academic Press, 1982.

Anna-Lise Smith and Lesli Baker. Getting a clue: creating student detectives and dragon slayers in your library. *Reference Services Review*, 39(4):628–642, 2011.

Erin Snyder and Jason R Hartig. Gamification of board review: a residency curricular innovation. *Medical education*, 47(5):524–525, 2013.

R. Sottilare, A. Graesser, X. Hu, and H. Holden. *Design Recommendations for Intelligent Tutoring Systems*. Army Research Laboratory, Orlando, FL: U.S., 2013.

Robert Sottilare, Arthur Graesser, Xiangen Hu, and Keith Brawner. *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques*. Robert Sottilare, 2015.

Robert A Sottilare. Approaches to reduce workload and skill requirements in the authoring of intelligent tutoring systems. *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques*, page 257, 2015.

Michelle Spence, Jason A Foster, Robert Irish, Patricia Kristine Sheridan, and Geoffrey Samuel Frost. Gamifying a library orientation tutorial for improved motivation and learning. In *2012 ASEE-American Society for Engineering Education Annual Conference*, 2012.

Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2013.

S. Steenbergen-Hu and H. Cooper. A meta-analysis of the effectiveness of intelligent tutoring systems (its) on college students' academic learning. *Journal of Educational Psychology*, 106:331–347, 2014.

Saiying Steenbergen-Hu and Harris Cooper. A meta-analysis of the effectiveness of intelligent tutoring systems on k–12 students' mathematical learning. *Journal of Educational Psychology*, 105(4):970, 2013.

Moon Ting Su, Chee Shyang Wong, Chuak Fen Soo, Choon Tsun Ooi, and Shun Ling Sow. Service-oriented e-learning system. In *Information Technologies and Applications in Education, 2007. ISITAE'07. First IEEE International Symposium on*, pages 6–11. IEEE, 2007.

P Suraweera, A Mitrovic, and B Martin. Widening the knowledge acquisition bottleneck for constraint-based tutors. *International Journal of Artificial Intelligence in Education*, 20 (2):137–173, 2010.

Thyago Tenório, Diego Dermeval, and Ig Ibert Bittencourt. On the use of ontology for dynamic reconfiguring software product line products. In IARIA, editor, *Proceedings of the Ninth International Conference on Software Engineering Advances*, pages 545–550, 2014.

Timothy Teo. Factors influencing teachers' intention to use technology: Model development and test. *Computers & Education*, 57(4):2432–2440, 2011.

Ambrosio Toval, Juan M Carrillo-de Gea, JL Fernandez-Aleman, and R Toval. Learning systems development using reusable standard-based requirements catalogs. In *IEEE Global Engineering Education Conference, EDUCON*, pages 907–912, 2011.

Christos Troussas, Efthimios Alepis, Maria Virvou, and Ieee. Mobile authoring in a multiple language learning environment. *Proceedings of the 5th International Conference on Information, Intelligence, Systems and Applications, Iisa 2014*, pages 405–+, 2014.

Kurt Vanlehn. The behavior of tutoring systems. *Int. J. Artif. Intell. Ed.*, 16(3):227–265, August 2006. ISSN 1560-4292. URL `http://dl.acm.org/citation.cfm?id=1435351.1435353`.

Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Kurt VanLehn, Winslow Burleson, Maria-Elena Chavez Echeagaray, Robert Christopherson, Javier Gonzalez Sanchez, Jenny Hastings, Yoalli Hidalgo Pontet, and Lishan Zhang. The affective meta-tutoring project: How to motivate students to use effective meta-cognitive strategies. In *19th International Conference on Computers in Education, Chiang Mai, Thailand*, 2011.

Dessislava Vassileva, Boyan Bontchev, Boryana Chavkova, and Vladimir Mitev. Software construction of an authoring tool for adaptive e-learning platforms. In *Informatics, 2009. BCI'09. Fourth Balkan Conference in*, pages 187–192. IEEE, 2009.

Viswanath Venkatesh and Fred D Davis. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, 46(2):186–204, 2000.

M. Virvou and C. Troussas. Knowledge-Based Authoring Tool for Tutoring Multiple Languages. *Intelligent Interactive Multimedia Systems and Services*, NA, 2011.

Hai H. Wang, Yuan Fang Li, Jing Sunc, Hongyu Zhang, and Jeff Pan. Verifying feature models using owl. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2), 2007. ISSN 1570-8268. URL `http://www.websemanticsjournal.org/index.php/ps/article/view/116`.

Kevin Werbach and Dan Hunter. *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, 2012.

C O E Wilches and G V H Palacio. Development of example-tracing tutors for teaching control systems performance fundamentals. In *Proceedings of the 4th World Congress on Information and Communication Technologies, WICT*, pages 290–295, 2014.

Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

Beverly Park Woolf. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.

Nick Yee. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *Presence: Teleoperators and virtual environments*, 15(3):309–329, 2006.

Lamia Abo Zaid, Frederic Kleinermann, and Olga De Troyer. Applying semantic web technology to feature modeling. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1252–1256. ACM, 2009.

Ramón Zatarian-Cabada, M.Lucia Barrón-Estrada, and Carlos Alberto Reyes García. EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. *Expert Systems with Applications*, 38(8):9522–9529, 2011. ISSN 09574174.

Jelena Zdravkovic, Constantinos Giannoulis, and Eric-Oluf Svee. Using i* to capture consumer preferences as requirements for software product lines. In *iStar*, pages 97–102. Citeseer, 2013.

Dongdai Zhou, Zhuo Zhang, Shaochun Zhong, and Pan Xie. The design of software architecture for e-learning platforms. In *Technologies for E-Learning and Digital Entertainment*, pages 32–40. Springer, 2008.

Appendix A

This appendix describes the methodology we have used to identify our related works on the use of software product line to develop intelligent tutoring systems.

A.1 SLR method

The literature review on the use of software product line and feature model to develop gamified ITS was conducted through a systematic literature review. As such, we used the protocol and guidelines proposed by Kitchenham and Charters [2007]. A Systematic Literature Review (SLR) is a means of identifying, evaluating and interpreting the available research findings related to a research question, topic area, or phenomenon. The main purpose for conducting a systematic review is to gather evidence on which to base conclusions [Kitchenham and Charters, 2007].

In order to perform this SLR, the guidelines and the systematic review protocol template proposed by Kitchenham and Charters [2007] were used. According to these guidelines, the SLR process includes several activities, which can be grouped in three main phases: planning of the SLR, conducting the SLR and reporting the SLR. It consists of the following steps: i) identification of the need for a systematic review; ii) formulation of a focused review question; iii) a comprehensive, exhaustive search for primary studies; iv) quality assessment of included studies; v) identification of the data needed to answer the research question; vi) data extraction; vii) summary and synthesis of study results; viii) interpretation of the results to determine their applicability; and ix) report-writing.

A software tool was used to support the SRL protocol definition. The tool, called StArt (State of the Art through Systematic Reviews) [LAPES, 2014], is used to provide support to researchers conducting SLRs. StArt has been empirically evaluated and it was demonstrated that such tool had positive results in the execution of SLRs [Hernandes et al., 2012]. In the following section, we describe the research questions of this SLR.

A.1.1 Research questions

This systematic review's purpose is to better understand how software product lines have been supporting different types of online learning environments and identify to what extent they have been applied to this them. Thus, we intend to answer the main research question:

> *How are software product lines supporting the construction of online learning environments?*

Note that we are including other types of educational systems, rather than only ITS, because this review might be useful for other studies beyond the scope of thesis. In the context of this thesis, we focus on the works that use SPL or feature modeling to develop ITS. Based on the main research question, specific questions were raised according to aspects that we are interested. These questions, their descriptions and motivations are described in Table A.1.

Table A.1: Research questions and motivations

Research question	**Description and motivation**
RQ1. What types of online education environments have been supported by the use of SPLs?	This question provides a starting point to understand what are the main types of educational environments (e.g., ITS, CSCL, LMS and so on) supported by the use of software product lines.
RQ2. Which studies have used ontology-driven software product line approaches?	The answer to this question indicates the existing studies that have used ontologies to drive the building of software product lines as well as presents how ontologies are been applied in the studies.
RQ3. How SPLs have been used to support the construction of gamified educational environments?	The answer to this question aims to identify existing works that are concerned with using SPL to construct gamified educational systems.

A.1.2 Inclusion and exclusion criteria

The aim of defining a criterion is to identify those primary papers which provide direct evidence about the research questions and also to reduce the likelihood of bias. We consider as primary papers the studies which present some kind of proposal to the area or/and present some kind of empirical validation of its contributions, whereas secondary papers are studies which only review a topic area [Kitchenham and Charters, 2007], i.e., a systematic literature review.

Studies were excluded if they were secondary, short-papers, non-peer reviewed, duplicated, non-English written, gray literature papers (e.g., books, theses, dissertations and so on), redundant papers of same authorship and if their focus was not using SPL to support the creation of educational environments and, in particular, ITS. Studies were eligible for inclusion in the review if they presented a peer-reviewed primary study, published at any point until November 2014[1] and that presented some contribution on the use software product line in the process of building a educational environment. Table A.2 summarizes the exclusion and inclusion criteria of this review.

Table A.2: Inclusion/exclusion criteria

#	**Inclusion Criteria**
1	Primary studies
2	Peer-reviewed studies
3	Studies that use SPL or some kind of variability model in the construction of educational environments
#	**Exclusion Criteria**
1	Secondary studies
2	Short-papers
3	Non peer-reviewed studies
4	Non English written papers
5	Gray literature
6	Redundant paper of same authorship
7	Paper not available
8	Studies that do not use SPL (or any variability modeling) in the construction of educational environments

[1]Period on which the review was conducted.

A.1.3 Sources selection and search

The search strategy included only electronic databases and was validated by experts on the topics. By using a search string and based on Chen et al. [2010], the following electronic databases were automatically searched: ScienceDirect[2], ISI Web of Science[3], Scopus[4], SpringerLink[5], ACM Digital Library[6], IEEE Xplore[7], and Compendex[8].

In following we present the systematic review process and the number of papers identified at each stage. Before describing these stages, it is worth emphasizing that, although the scope of this paper is reviewing the use of software product line to develop ITS, this research is part of an ongoing work which intends to review the use of software product line in computers and education, including for example, several types of educational systems (e.g., computer supported collaborative learning, massive open online courses, adaptive educational hypermedia systems, etc). Hence, the search and selection strategy (i.e., the search string and Steps 1-5) aims to capture studies related to all these topics. As such they will be useful for several other studies. The papers related to ITS, which are the focus of this review, are only identified and in the extraction step of the SLR, as it will be further described.

In Step 1 the studies were obtained from electronic databases using the following search terms:

(1) "software product line"

(2) "feature model" OR "variability model"

(3) "software platform"

(4) "computers and education" OR "online education"

(5) "educational environment" OR "educational system" OR "learning management system" OR "learning environment" OR "artificial intelligence in education"

[2] `http://www.sciencedirect.com/`
[3] `http://apps.webofknowledge.com`
[4] `http://www.scopus.com`
[5] `http://link.springer.com/`
[6] `http://dl.acm.org/`
[7] `http://ieeexplore.ieee.org`
[8] `http://www.engineeringvillage.com/`

(6) "e-learning" OR "m-learning" OR "t-learning"

(7) "web-based education" OR "semantic web-based education" OR "semantic web and education"

(8) "collaborative learning" OR "computer supported collaborative learning" OR "CSCL"

(9) "adaptive educational hypermedia systems" OR "adaptive educational systems" OR "adaptive learning systems"

(10) "intelligent tutoring system" OR "intelligent educational systems"

(11) "MOOCS" OR "massive open online courses"

(12) "gamification"

These search terms for different types of SPL and education articles were combined in the following way:

(1 OR 2 OR 3) AND (4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11 OR 12)

The definition of these search terms was based on the mapping of different types of online educational environments that are researched by the computers and education community. Figure A.1 depicts the steps of the selection process showing the number of studies in each one these steps.

At Step 2, duplicated papers were automatically detected and removed using the StArt tool, remaining a set of 1,823 papers. Then, in Step 3 titles, keywords, and publication venue of each paper were reviewed and those that were not related to the research questions (-1,771 papers) were excluded. If there was insufficient data, the paper was left for the next assessment. After finishing the Step 3, 52 papers remained in the selection process and reviewers analyzed, in Step 4, paper's abstracts and excluded those according to the exclusion criteria (#1-8 criteria from Table A.2), excluding 20 papers. If there was insufficient data, the paper was left for the next step.

In Step 5, the complete texts of the papers selected at Step 4 (32 papers) were retrieved, the introduction and conclusion of each paper were read and each paper was full-screened. At this step, some papers were also excluded based on the exclusion criteria (-13 papers). As a result, 19 papers were finally included for the next stage of this review.

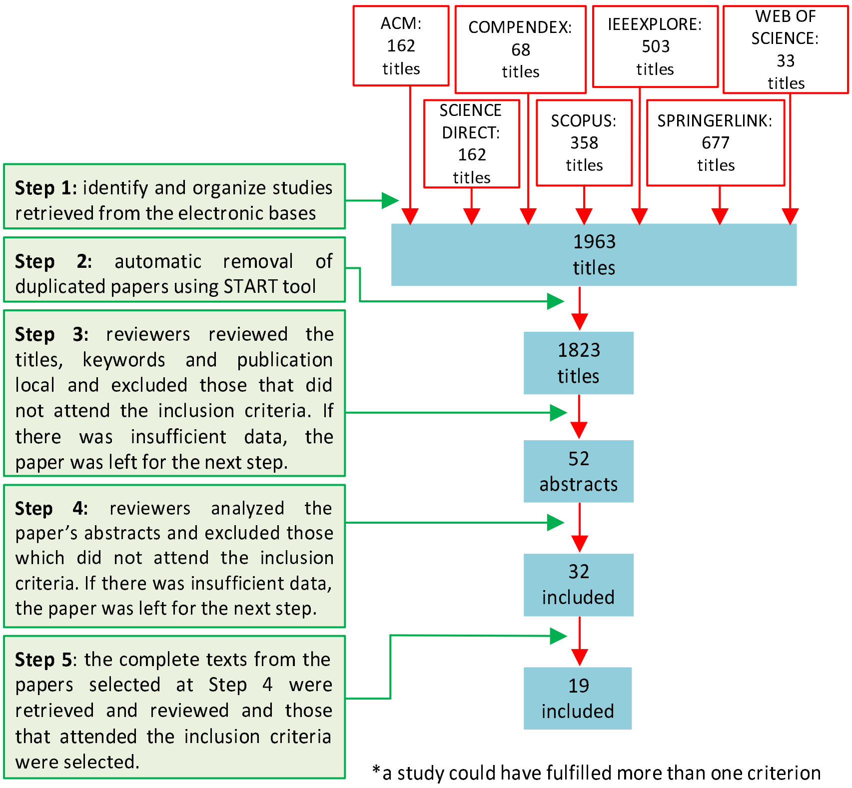


Figure A.1: Sources and selection flow of the SLR on the use of SPL in online learning environments

A.1.4 Quality assessment

The quality assessment (QA) of selected studies was achieved by a scoring technique to evaluate the credibility, completeness and relevance of the selected studies. All papers were evaluated against a set of 10 quality criteria. They were adapted from existing study quality assessment criteria used in the literature. The assessment instrument used is presented in Table A.3.

Table A.3: Study quality assessment criteria

#	Questions	Possible Answers
1	Is there a rationale for why the study was undertaken? [Mahdavi-Hezavehi et al., 2013]	Y=1, N=0, P=0.5
2	Is the paper based on research (or is it merely a “lessons learned” report based on expert opinion)? [Dybå and Dingsøyr, 2008]	Y=1, N=0
3	Is there a clear statement of the goals of the research? [Dybå and Dingsøyr, 2008]	Y=1, N=0, P=0.5
4	Is the proposed technique clearly described? [Achimugu et al., 2014]	Y=1, N=0, P=0.5
5	Is there an adequate description of the context (industry, laboratory setting, products used and so on) in which the research was carried out? [Dybå and Dingsøyr, 2008][Mahdavi-Hezavehi et al., 2013]	Y=1, N=0, P=0.5
6	Is the study supported by a tool? [Dermeval et al., 2015b]	Y=1, N=0
7	Was the study empirically evaluated? [Dermeval et al., 2015b]	Y=1, N=0
8	Is there a discussion about the results of the study? [Dermeval et al., 2015b]	Y=1, N=0, P=0.5
9	Are the limitations of this study explicitly discussed? [Ding et al., 2014]	Y=1, N=0, P=0.5
10	Does the research also add value to the industrial community? [Dybå and Dingsøyr, 2008][Achimugu et al., 2014]	Y=1, P=0.5

We relied on systematic literature reviews published high reputation venues (e.g., Information and Software Technology Journal) in the context of empirical software engineering research to define the quality assessment criteria. In particular, we adapted some of our criteria following the works by Mahdavi-Hezavehi et al. [2013] (Q1 and Q5), Dybå and Dingsøyr [2008] (Q3, Q5 and Q10), Achimugu et al. [2014] (Q4 and Q10), Ding et al. [2014] (Q9) and Dermeval et al. [2015b] (Q6, Q7 and Q8).

The scores of questions Q2, Q6 and Q7 were determined using a two-grade scale score (Yes/No). If the answer were Yes, the study received 1 point in this question, otherwise, it received 0 point. Besides these alternatives, the questions Q1, Q3, Q4, Q5, Q8 and Q9 also allowed a third one. If the contribution was not so strong, the study received 0.5 point, consisting in a three-grade scale score to these questions. Q10 receives 1 point if the study is applied in industry and 0.5 point if its setting is academy.

The study quality score is computed by finding the sum of all its scores of the answers to the questions. Each selected paper was assessed independently by the authors. All discrepancies on the scores were discussed among the authors, and the study was reevaluated in cases of non-agreement with the aim of reaching consensus.

A.1.5 Data extraction and synthesis

After the definition of the search and the selection processes, the data extraction process was performed by reading the introduction and conclusion; and full-text screening each one of the selected papers. In order to guide this data extraction, the data collection from Kitchenham and Charters [2007] was adopted. During this stage, data was extracted from each of the 19 primary studies included in this systematic review according to a predefined extraction form (see Table A.4). This form enabled us to record full details of the papers under review and to be specific about how each of them addressed our research questions. As well as the selection process, the data extraction was full aided by the StArt tool.

Table A.4: Extraction form

#	*Study Data*	*Description*	RQ
1	Study identifier	Unique id for the study	Study overview
2	Date of data extraction		Study overview
3	Authors, Year, Title, Country		Study overview
4	Article source		Study overview
5	Type of article	Journal, conference, workshop, book chapter	Study overview
6	Application context	Industrial, academic	Study overview
7	Research method (based on Easterbrook et al. [2008])	Controlled experiment, case study, survey, ethnography, action research, illustrative scenario, not applicable	Study overview
8	Educational system	What is the educational environment (ITS, CSCL, LMS, MOOC, etc) targeted by the study?	RQ1
9	Ontology usage	Is the SPL using an ontology-driven approach?	RQ2
10	Gamification usage	Is the SPL used to construct gamified educational environments?	RQ3

A.2 Quality assessment results

A total of 19 papers met the inclusion criteria and their data were extracted. Before presenting the results and analysis for each research question, we depict the quality assessment results and give a detailed overview of the general characteristics of the studies. The quality assessment results are showed in the Table A.5[9] according to the questions described in Table A.3.

Table A.5: List of papers included in the review along with their quality scores

ID	Authors	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total	Qual. (/%)
S01	Caballé and Xhafa [2010]	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0	0.5	0.5	8.0	80.0%
S02	Dalmon et al. [2012]	1.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	0.0	0.5	5.5	55.0%
S03	Damaševičius [2010]	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	0.5	7.5	75.0%
S04	Díez et al. [2009]	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	0.0	0.5	5.5	55.0%
S05	Gütl [2007]	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.5	4.5	45.0%
S06	Lytras et al. [2005]	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	4.0	40.0%
S07	Montilva et al. [2002]	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.5	8.5	85.0%
S08	Murwantara [2012]	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	0.0	0.5	5.5	55.0%
S09	Oberweis et al. [2007]	1.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	7.0	70.0%
S10	Sanchez Barreiro et al. [2014]	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.5	0.5	8.0	80.0%
S11	Santos et al. [2012]	1.0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	1.0	0.5	6.5	65.0%
S12	Schuwer and Kusters [2014]	1.0	1.0	1.0	1.0	0.5	0.0	1.0	0.5	0.5	0.5	7.0	70.0%
S13	Shih et al. [2006]	0.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.5	4.5	45.0%
S14	Silva et al. [2011]	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.5	4.5	45.0%
S15	Su et al. [2007]	1.0	1.0	1.0	1.0	0.5	0.0	0.0	1.0	0.5	0.5	6.5	65.0%
S16	Toval et al. [2011]	1.0	1.0	1.0	1.0	0.5	1.0	0.0	1.0	0.5	0.5	7.5	75.0%
S17	Vassileva et al. [2009]	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.5	6.5	65.0%
S18	Zdravkovic et al. [2013]	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.5	5.5	55.0%
S19	Zhou et al. [2008]	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.5	4.5	45.0%
	Average	0.95	0.84	1.00	1.00	0.45	0.37	0.21	0.45	0.34	0.55	6.16	61.6%

In fact, the quality score of the papers is quite scattered. There are papers with high-quality scores, whereas there are papers with low-quality scores. Taken together, these 10 criteria provided a measure of the extent to which we could be confident that a particular

[9] Ids are assigned to the papers per its position in a list alphabetically sorted by the first author of the papers.

study's findings could make a valuable contribution to this review. The overall average score of quality for the studies is 6.16, which we may consider regular. With respect to specific averages per quality criterion, questions Q1, Q2, Q3, and Q4 received the highest scores ($\geq$ 0.84). Q10 received an intermediary score, with 0.55 points, whereas the questions Q5, Q6, Q7, Q8, and Q9 received the lowest scores (<0.5)

A.3 Overview of the studies

In following we depict general characteristics of the studies included in the review: year of publication, type of source, countries where the researches were conducted, research method and application context.

A.3.1 Publication year

The reviewed studies were published between 2002 and 2014[10]. From a temporal point of view (Fig. A.2) it is possible to see a increase in the number of publications in the middle of the time frame with a slight decrease in the end. Note that there is an increasing tendency in the number of papers using SPL to address the educational environments. However, it is also worth noting that, as the search process of this review was performed in 2014, a slight decrease in the number of publications would be expected in such year because some papers might be in press.

A.3.2 Application context

The application context on which studies were published are categorized either as industrial or academic settings. As shown in Figure A.3, majority of the papers (77.6%; 52 studies) are considered academic. However, it is worth noting that a considerable amount of the studies were conducted in an industrial setting (22.4%; 15 studies), indicating that, even though the concept of ontology is not widespread within the requirements engineering community, its use has also been significantly investigated in industry.

[10]Note that this SLR considers papers from 2002 since it is one year ahead when the first work on SPL was published [Clements and Northrop, 2001]

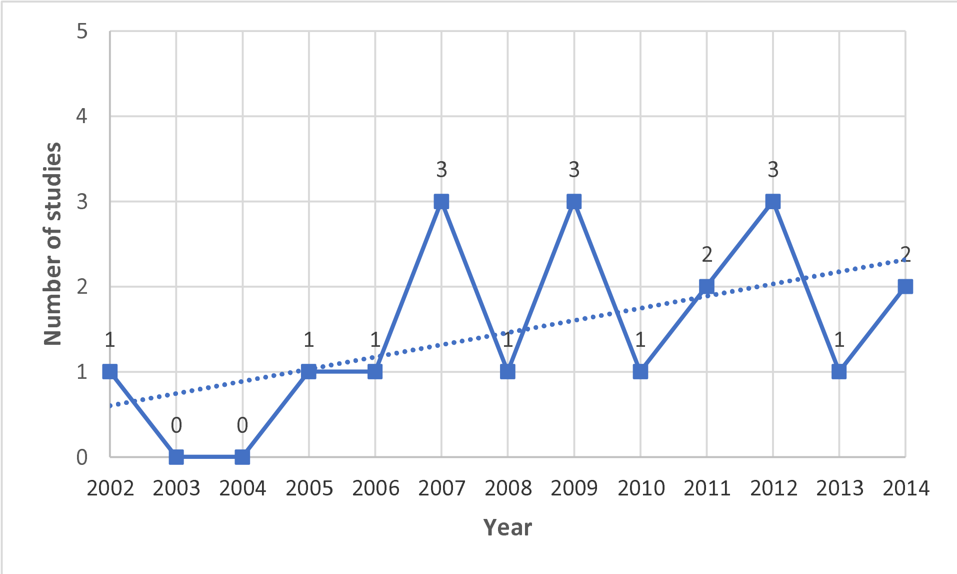


Figure A.2: Temporal view of the studies

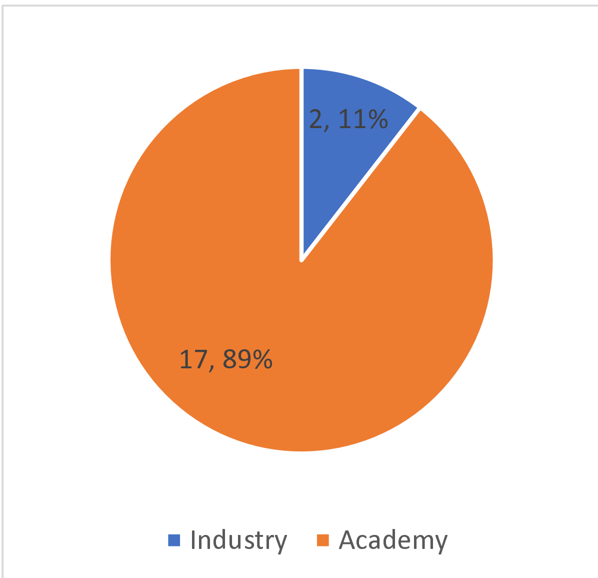


Figure A.3: Distribution of papers by application context

A.3.3 Type of source

The studies included in this review may be of journal, conference proceedings, workshop or book chapter publications. As shown in Figure A.4, majority of studies are published in conference proceedings (42.10%; 8 studies), followed by journal publications (36.84%; 7 studies), book chapter publications (15.78%; 3 studies) and workshop (5.26%; 1 study).

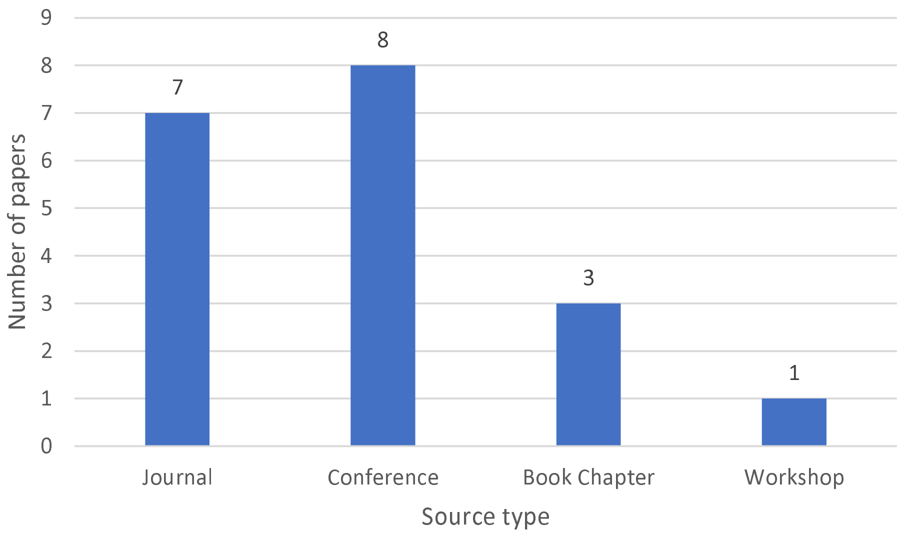


Figure A.4: Distribution of papers by type of publication

A.3.4 Research method

The classification of publications was based on the categories (i.e., controlled experiment, quasi-experiment, case study, survey research, ethnography and action research) defined by Easterbrook et al. [2008]. However, we have defined two extra categories: illustrative scenario and not applicable. The first is appropriate for papers that just explain their contributions using small examples or argumentation. The latter refers to the papers that do not present any kind of research method or explanation of using the proposal.

Illustrative Scenarios (39.39%; 13 studies) constitute the majority of the studies, followed by Controlled Experiments (27.27%; 9 studies), Case Studies (15.15%; 5 studies), Not Applicable (15.15%; 5 studies) and Survey (3.03%; 1 study). There were no quasi-experiment, ethnography and action research papers in our classification.

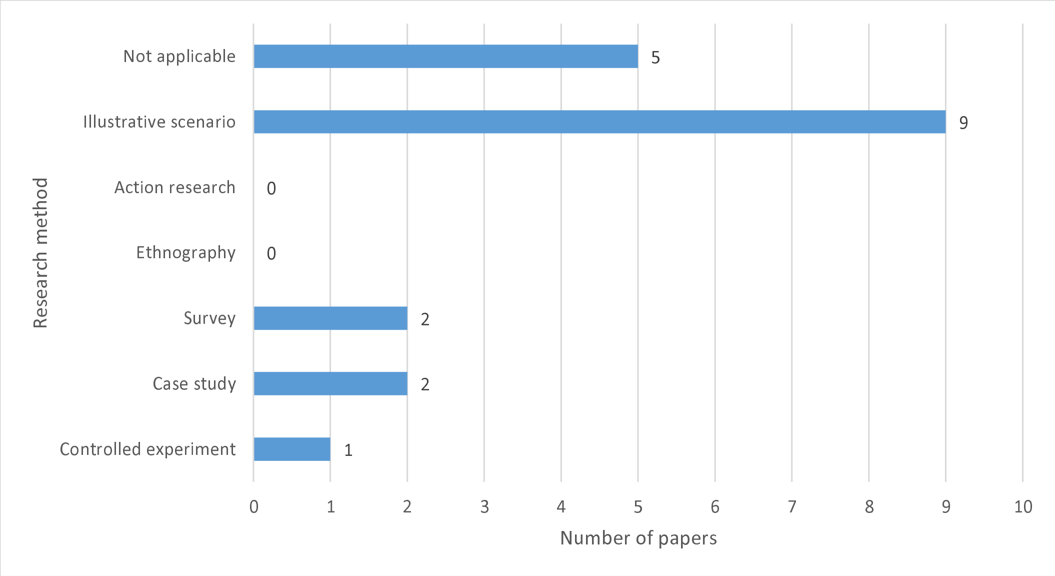


Figure A.5: Distribution of papers by research method

A.3.5 Country

The studies were also categorized according to the authors' country (Figure A.6). Most of the papers are published by researchers from Brazil and Spain, each one with 15.8% of total number of papers. Next, researchers from Germany are responsible by 10.5% of the papers. The remaining countries identified appear with 5.2%, each one, as shown in Figure A.6.

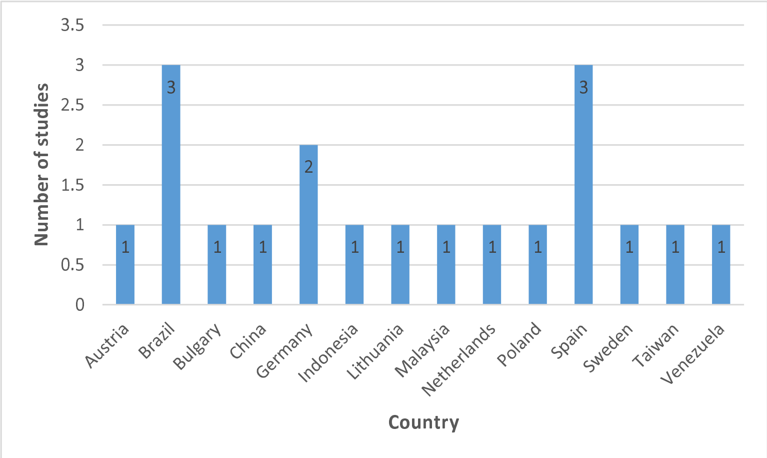


Figure A.6: Distribution of papers by country

A.4 RQ1: SPL in educational environments

As previously mentioned, the purpose of this question was to identify the types of educational environments in which SPL has been used. Particularly, we are interested, in the context of this thesis, in intelligent tutoring systems. As shown in Table A.6, most of the papers identified in this review are targeting the use of SPL to construct learning management systems (89.47%; 17 papers); followed by the use of SPL in ITS and computer-supported collaborative learning systems (CSCL), each type with 1 paper (5.26%). No paper were found to address the other types of educational environments.

Table A.6: Distribution of works using SPL over educational environments

Educational environment	Studies	Freq.	%
LMS	S01, S03, S04, S05, S06, S07, S08, S09, S10, S11, S12, S13, S14, S15	17	89.47%
CSCL	S02	1	5.26%
ITS	S16	1	5.26%

A.5 RQ2: Use of ontologies along with SPL to develop educational systems

The intention of this question was to identify if the papers included in this review use an ontology-driven software product line approach – in a similar line of one of our objectives – to construct educational environments. To answer this question, we divide the studies into two categories: using ontologies along with SPL and not using ontologies in development process of SPL to be used in the design of educational environments. In fact, only two papers (10.5%), among the 19 papers included, use ontologies to drive the development process of SPL along with educational environments, as seen in Figure A.7. Note that there is only one paper that uses ontologies to support the SPL engineering of intelligent tutoring systems, as also show in Figure A.7. This paper will be further presented in Section 3.1.2 when we discuss our related works in the use of feature modeling/SPL to design ITS.

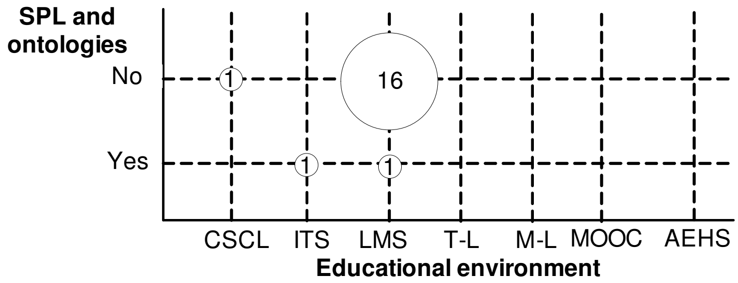


Figure A.7: Using ontologies to support SPL engineering of educational environments

A.6 RQ3: Use of SPL to develop gamified educational systems

The purpose of this question was to identify if papers were using SPL to develop gamified educational environments. To answer this question, we divided the studies in two categories: the papers that use SPL to develop gamified educational environments and the papers that do not use SPL for such aim. As shown in Figure A.8, we could not find any work that uses SPL to target gamified educational systems, including, obviously, intelligent tutoring systems.

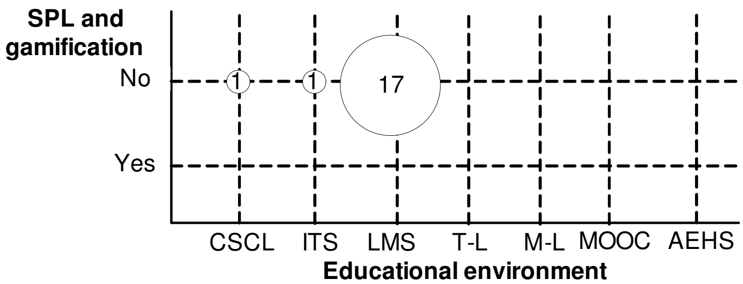


Figure A.8: Using SPL along with gamified educational environments

Appendix B

This appendix describes the methodology we have used to identify the related works on the use of ITS authoring tools.

B.1 SLR method

Likewise the systematic literature review described in Section A.1, a SLR on the use of ITS authoring tools was also conducted following the protocol and guidelines proposed by Kitchenham and Charters [2007]. In the following sections, we describe the details of this method.

B.1.1 Research questions

This systematic review's purpose is to understand and synthesize how authoring tools support intelligent tutoring systems design regarding non-programmer authors' point of view and identify to what extent these tools have been applied for designing this kind of system. Thus, we intend to answer the main research question:

> *How are authoring tools supporting the design of intelligent tutoring systems for non-programmer authors?*

Based on the main research question, specific questions were raised according to authoring tools and ITS aspects that we are interested. The questions, along with their descriptions and motivations are described in Table B.1.

Table B.1: Research questions and motivations

Research Question	Description and Motivation
RQ1. Which ITS components can be authored?	This question provides a starting point to understand which are the main ITS components (i.e., student model, domain model, pedagogical model and interface model) supported by the use of authoring tools. This question also investigates if authoring tools are targeting the "gamification model" of ITS;
RQ2. Which ITS types can be authored?	This question intends to identify which are the main ITS types (e.g., example-tracing, constraint-based and so on) that are been designed by the use of authoring tools. This question also investigates if there are studies that propose authoring tools for gamified ITS;
RQ3 How tools are supporting ITS authoring process?	This question aims to describe how authoring tools are supporting the authoring process of ITS. It is important because it provides a set of contributions regarding the use of authoring tools to address ITS design, which can be used by researchers that might be interested in using authoring tools for this kind of educational system;
RQ4. What authoring technologies have been used to design ITS?	This question intends to identify which are the main technologies used to develop authoring tools in order to design ITS. The answer to this question is important because it can serve as a guide to researchers that might use some specific technology to develop authoring tools for ITS;

B.1.2 Inclusion and exclusion criteria

The aim of defining a criterion is to identify those primary papers which provide direct evidence about the research questions and also to reduce the likelihood of bias [Kitchenham and Charters, 2007]. Note that we consider as primary papers the studies which present some kind of proposal to the area or present some kind of empirical evaluation of its contributions, whereas secondary papers are studies which only review a topic area, e.g., surveys, systematic literature reviews or systematic mappings.

Studies were eligible for inclusion in the review if they presented a peer-reviewed primary study, published since January 2009 to June 2016 and that presented some contribution on the use of authoring tools to support ITS design. Our decision on such period was made to reduce repetitive effort and make use of existing work since Woolf [2010] provides a general description of the use of authoring tools to design ITS before 2009 updating the analysis of state of the art provided by Murray [2003]. Moreover, we also intend to gather more recent papers about the topic in order to get insights as well as to consider emerging technologies that could be used along with authoring tools (e.g., mobile learning, gamification, learning analytics and so on) for non-programmer authors.

Table B.2: Inclusion/exclusion criteria

#	Inclusion Criterion
1	Primary studies
2	Peer-reviewed studies
3	Study published between January 2009 and June 2016
4	Studies that use authoring tool to design ITS for non-programmer authors
#	**Exclusion Criterion**
5	Secondary studies
6	Short-papers ($\leq$ 5 pages)
7	Non peer-reviewed studies
8	Duplicated studies (only one copy of each study was included)
9	Non English written papers
10	Gray literature
11	Redundant paper of same authorship
12	Position paper
13	Studies that do not present or evaluate any authoring tool for non-programmers
14	Papers about simulation
15	Papers about augmented reality
16	Papers about serious games
17	Papers about storytelling
18	Papers about disability
19	Studies that do not use authoring tools to design ITS

Studies were excluded if they were secondary, short papers, non-peer reviewed, duplicated, non-English written, gray literature papers (e.g., books, theses, dissertations and so on), redundant papers of same authorship[1], position papers and if their focus was not using authoring tools to support ITS design for non-programmer authors. Furthermore, this research is concerned with generic and technology/paradigm ITS authoring tools, i.e., we are not including works that propose authoring tools that need to handle strict ITS constraints. For this reason, simulation, augmented reality, serious games, storytelling, and disability (focusing on learners' disabilities) exclusive papers were also excluded. For instance, an ITS authoring tool that considers learners' disabilities (e.g., blindness) should need to design a special pedagogical model tied to such disability that would not be generic enough to be used in other contexts. The summarized inclusion and exclusion criteria are presented in

[1]If similar papers are included from the same authorship, we keep in the review the more complete and recent paper (priority is given to journal papers)

Table B.2.

B.1.3 Sources selection and search

The search strategy included only electronic databases and was validated by experts on ITS and authoring tools. According to Chen's recommendation [Chen et al., 2010], the following electronic databases were automatically searched: ScienceDirect[2], ISI Web of Science[3], Scopus[4], SpringerLink[5], ACM Digital Library[6], IEEE Xplore[7] and Compendex[8].

Figure B.1 shows the systematic review process and the number of papers identified at each stage. Before describing these stages, it is worth emphasizing that, although the scope of this review is reviewing the use of authoring tools in ITS design, this research is part of an ongoing work which intends to review the use of authoring tools in computers and education, including for example, several types of educational systems (e.g., computer supported collaborative learning, massive open online courses, adaptive educational hypermedia systems, etc) and research trends (e.g., gamification, mobile learning and education data mining/learning analytics). Hence, the search and selection strategy (i.e., the search string and Steps 1-5) aims to capture studies related to all these topics. As such they will be useful for several other studies. The papers related to ITS, which are the focus of this review, are only identified and selected in Step 6 of the process, as it will be further described.

In Step 1 the studies were obtained from electronic databases using the following search terms:

(1) "authoring tool" OR "authoring system" OR "intelligent authoring"

(2) "computers and education" OR "e-learning"

(3) "educational environment" OR "educational system" OR "learning environment"

[2] `http://www.sciencedirect.com/`
[3] `http://apps.webofknowledge.com`
[4] `http://www.scopus.com`
[5] `http://link.springer.com/`
[6] `http://dl.acm.org/`
[7] `http://ieeexplore.ieee.org`
[8] `http://www.engineeringvillage.com/`

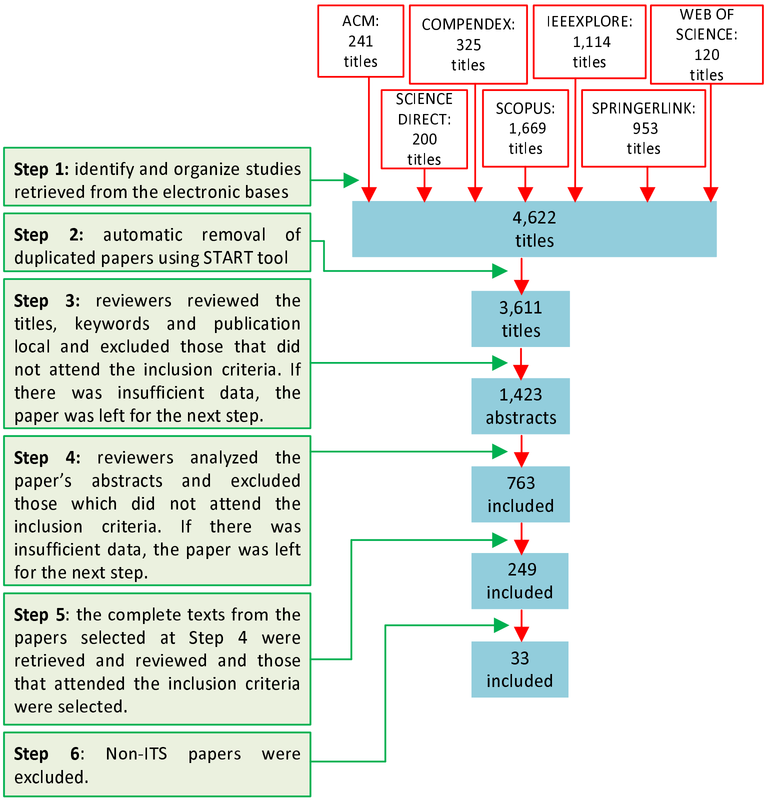


Figure B.1: Paper selection flowchart

(4) "learning management system"

(5) "m-learning"OR "mobile learning"

(6) "t-learning" OR "tv learning"

(7) "online education"OR "online learning" OR web-based education" OR "semantic web-based education" OR "semantic web and education"

(8) "collaborative learning" OR "computer supported collaborative learning" OR "CSCL"

(9) "intelligent tutoring system" OR"intelligent educational systems"

(10) "MOOCS" OR "massive open online courses"

(11) "adaptive educational hypermedia systems"

(12) "adaptive educational systems" OR "adaptive learning systems"OR "artificial intelligence in education"

(13) "gamification"

These search terms for several applications of authoring tools to computers and education were combined in the following way:

(1 AND (2 OR 3 OR 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10 OR 11 OR 12 OR 13))

The definition of these terms was based on two main sources: i) the scope of relevant journals on the topic (e.g., the International Journal of Artificial Intelligence and Education (IJAIED) and IEEE Transactions on Learning Technologies) in order to identify different types of educational systems and ii) asking suggestions to experts on the topic (authoring tools and ITS). Furthermore, as we intended to retrieve recent papers in the literature, our search only considered the period between January 2009 and June 2016, which is an inclusion criterion, as described in Section B.2.

The search results (4,622 papers) were automatic downloaded and were inserted into and organized with the aid of StArt tool. Figure B.1 depicts the steps of the selection process showing the number of studies in each one these steps.

At Step 2, duplicated papers were automatically detected and removed using the StArt tool, remaining a set of 3,611 papers. Then, in Step 3 authors reviewed titles, keywords, and publication venue of each paper and excluded those that were not related to the research questions (-2,188 papers). If there was insufficient data, the paper was left for the next assessment. After finishing the Step 3, 1,423 papers remained in the selection process and reviewers analyzed, in Step 4, paper's abstracts and excluded those according to 14 exclusion criteria (#4-18 criteria from Table 2), excluding 660 papers. If there was insufficient data, the paper was left for the next step.

In Step 5, the complete texts of the papers selected at Step 4 (763 papers) were retrieved, the introduction and conclusion of each paper were read and each paper was full-screened. Papers were excluded according to the #4-18 exclusion criteria again (-514 papers).

Until Step 5, any application of authoring tool to computers and education was considered to be included in the review. Recall that this is intentional, as we may use the studies identified so far for several types of research under development. Hence, the specific exclusion criterion for non-ITS authoring papers was applied, in Step 6, to the 249 remaining studies of Step 5, in order to filter the papers exclusively related to ITS design (the focus of this paper). As a result, 33 papers were finally included for the next stage of the review.

B.1.4 Quality assessment

The quality assessment (QA) of selected studies was achieved by a scoring technique to evaluate the credibility, completeness, and relevance of the selected studies. All papers were evaluated against a set of 10 quality criteria. Seven of them were adapted from existing study quality assessment criteria used in the literature, the remaining four questions were proposed according to the scope and research questions of this systematic literature review. The assessment instrument used is presented in Table B.3. Q1, Q2, Q3, Q4, Q5, Q6, Q9, and Q10 were adopted from the literature, while Q7 and Q8 were proposed.

Table B.3: Study quality assessment criteria

#	Questions	Possible Answers
Q1	Is there a rationale for why the study was undertaken? [Mahdavi-Hezavehi et al., 2013]	Y=1, N=0, P=0.5
Q2	Is the paper based on research (or is it merely a “lessons learned” report based on expert opinion)? [Dybå and Dingsøyr, 2008]	Y=1, N=0
Q3	Is there a clear statement of the goals of the research? [Dybå and Dingsøyr, 2008]	Y=1, N=0, P=0.5
Q4	Is the proposed technique clearly described? [Achimugu et al., 2014]	Y=1, N=0, P=0.5
Q5	Is there an adequate description of the context (industry, laboratory setting, products used and so on) in which the research was carried out? [Dybå and Dingsøyr, 2008][Mahdavi-Hezavehi et al., 2013]	Y=1, N=0, P=0.5
Q6	Does the study provide a tool? If yes, is the tool available for download or on the web? [Dermeval et al., 2015b]	Y=1, P=0.5, N=0
Q7	Was the study empirically evaluated?	Y=1, N=0
Q8	Is there a discussion about the results of the study?	Y=1, N=0, P=0.5
Q9	Are the limitations of this study explicitly discussed? [Ding et al., 2014]	Y=1, N=0, P=0.5
Q10	Does the study also evaluate the proposal in industrial settings? [Dybå and Dingsøyr, 2008][Achimugu et al., 2014]	Y=1, N=0

We relied on systematic literature reviews published in a high reputation venue (i.e., Information and Software Technology Journal) in the context of empirical software engineering research to define seven of the quality assessment criteria. In particular, we adapted some of our criteria following the works by Mahdavi-Hezavehi et al. [2013] (Q1 and Q5), Dybå and Dingsøyr [2008] (Q2, Q3, Q5 and Q10), Achimugu et al. [2014] (Q4 and Q10) Dermeval et al. [2015b] (Q6) and Ding et al. [2014] (Q9).

The scores of questions Q2 and Q7 were determined using a two-grade scale score (Yes/No). If the answer were Yes, the study received 1 point in this question, otherwise, it received 0 point. Besides these alternatives, the questions Q1, Q3, Q4, Q5, Q8 and Q9 also allowed a third one. If the contribution was not so strong, the study received 0.5 point, consisting of a three-grade scale score to these questions. Q6 receives 1 point if the paper proposes an authoring tool which is available for download or on the web, it receives 0.5 point if the tool is not available and receives 0 if it does not propose an authoring tool. Q10 receives 1 point if the study is applied in industry and 0.5 point if its setting is academic.

After finishing the selection and extraction stages of our review, first and second authors independently assessed – according to the criteria presented in Table B.3 – the 33 papers included in the review. Then, the scores marked by the authors are organized in a spreadsheet and, for each criterion and paper, scores are compared to identify disagreements. All studies with non-agreement are discussed among all the authors, and the study is reevaluated with the aim of reaching consensus. The resulting study quality score is computed by finding the sum of all consensual scores of the answers to the questions on Table B.3.

B.1.5 Data extraction and synthesis

After the definition of the search and the selection processes, the data extraction process was performed by reading each one of the selected papers. In order to guide this data extraction, the data collection from Kitchenham and Charters [2007] was adopted. During this stage, data was extracted from each of the 33 primary studies included in this systematic review according to an extraction form (see Table B.4). This form enabled us to record full details of the papers under review and to be specific about how each of them addressed our research questions. Like the selection process, the data extraction was fully aided by the StArt tool.

B.2 Quality assessment results

The quality assessment of the selected studies is useful to increase the accuracy of the data extraction results. This evaluation helped to determine the validity of the inferences proffered and in ascertaining the credibility and coherent synthesis of results.

The quality assessment results are showed in the Table B.5[9] according to the questions described in Table B.3. Note that this step was performed by the author of thesis and by experts on ITS authoring tools. In fact, the quality score of the papers is quite scattered.

[9]Ids are assigned to the papers per its position in a list alphabetically sorted by the first author of the papers.

Table B.4: Extraction form

#	*Study Data*	*Description*	Relevant RQ
1	Study identifier	Unique id for the study	Study overview
2	Date of data extraction		Study overview
3	Authors, Year, Title, Country		Study overview
4	Article source		Study overview
5	Type of article	Journal, conference, workshop, book chapter	Study overview
6	Application context	Industrial, academic	Study overview
7	Research method (based on Easterbrook et al. [2008])	Controlled experiment, case study, survey, ethnography, action research, illustrative scenario, not applicable	Study overview
8	Name of the contribution		Study overview
9	ITS component	What were the ITS components addressed by the authoring tool? (Student Model, Domain Model, Pedagogical Model and Interface Model)	RQ1
10	ITS type	What ITS type has been authored by the tool?	RQ2
11	Kind of support (feature)	How tools are supporting ITS authoring process?	RQ3
12	Technology	Which technologies have been used?	RQ4
13	Authoring time regarding course	When does the authoring occurs? (Pre-course, during the course and post-course)	RQ5
14	Evidence	What was the evidence which indicate that the use of authoring tools benefits the ITS design? (Negative argumentation, negative with empirical evaluation, positive argumentation, positive with empirical evaluation)	RQ6

There are papers with high-quality scores, whereas there are papers with low-quality scores. Taken together, these 10 criteria provided a measure of the extent to which we could be confident that a particular study's findings could make a valuable contribution to this review.

B.3 Overview of the studies

In following we depict general characteristics of the studies included in the review: year of publication, type of source, research method and application context.

B.3.1 Publication year

The reviewed papers were published between 2009 and 2016. From a temporal point of view (Fig. B.2), we can note an increasing number of papers from 2009 to 2011, followed by a decrease in 2012 and 2013 years. Then, as shown in the figure, there is an increase in the

Table B.5: List of papers included in the review along with their quality scores

ID	Author	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Total Score	Qual.
S03	Aleven et al. [2009a]	1	1	1	1	0.5	1	1	1	0.5	1	9	90.0%
S04	Aleven et al. [2016]	1	1	1	1	1	1	1	1	1	0	9	90.0%
S08	Blessing et al. [2009]	1	1	1	1	1	1	1	1	1	0	9	90.0%
S15	Gilbert et al. [2015]	1	1	1	1	1	1	1	1	1	0	9	90.0%
S20	MacLellan et al. [2014]	1	1	1	1	1	0.5	1	1	1	0	8.5	85.0%
S07	Blessing et al. [2015]	1	1	1	1	1	1	1	1	0	0	8	80.0%
S24	Mitrovic et al. [2009]	1	1	1	1	1	0.5	1	1	0.5	0	8	80.0%
S29	Suraweera et al. [2010]	1	1	1	1	1	0.5	1	1	0.5	0	8	80.0%
S11	Chou et al. [2011]	1	1	0.5	1	1	0.5	1	1	0.5	0	7.5	75.0%
S19	Lane et al. [2015]	1	1	1	1	1	0.5	1	1	0	0	7.5	75.0%
S23	Matsuda et al. [2015]	1	1	1	1	0.5	0.5	1	1	0.5	0	7.5	75.0%
S12	Devasani et al. [2012]	1	1	1	1	1	0	1	1	0	0	7	70.0%
S01	Abbas et al. [2014]	1	1	1	1	1	0.5	1	0	0	0	6.5	65.0%
S33	Zatarian-Cabada et al. [2011]	0.5	1	1	1	1	0.5	1	0.5	0	0	6.5	65.0%
S10	Chakraborty et al. [2010]	1	1	1	1	0.5	0.5	1	0	0	0	6	60.0%
S18	Heffernan [2014]	1	0	1	1	0.5	1	0	1	0.5	0	6	60.0%
S21	MacLellan et al. [2015]	1	0	1	1	0.5	0.5	0	1	1	0	6	60.0%
S17	Guin and Lefevre [2013]	0.5	0	1	1	1	0.5	0	1	0.5	0	5.5	55.0%
S27	Paquette et al. [2010]	1	0	1	1	0.5	1	0	0.5	0.5	0	5.5	55.0%
S32	Wilches and Palacio [2014]	1	0	1	1	0.5	0	0	1	0	1	5.5	55.0%
S14	Fox et al. [2011]	1	0	1	1	1	0.5	0	0.5	0	0	5	50.0%
S22	Marcus et al. [2010]	1	0	0.5	1	1	0.5	0	1	0	0	5	50.0%
S26	Olsen et al. [2014]	1	0	1	1	0.5	0.5	0	1	0	0	5	50.0%
S25	Olney and Cade [2015]	1	0	1	1	0.5	0.5	0	0.5	0	0	4.5	45.0%
S30	Troussas et al. [2014]	1	0	1	1	0.5	0.5	0	0	0	0	4	40.0%
S13	Escudero and Fuentes [2010]	1	0	0.5	1	0.5	0.5	0	0	0	0	3.5	35.0%
S16	Grubisic et al. [2009]	0.5	0	1	1	0	0.5	0	0.5	0	0	3.5	35.0%
S28	Refanidis [2011]	0.5	0	1	1	0	0.5	0	0.5	0	0	3.5	35.0%
S31	Virvou and Troussas [2011]	1	0	0.5	1	0.5	0.5	0	0	0	0	3.5	35.0%
S06	Barron-Estrada et al. [2010]	0	0	0.5	1	0	1	0	0	0	0	2.5	25.0%
S09	Brawner [2015]	0	0	0.5	1	0	1	0	0	0	0	2.5	25.0%
S02	Alepis and Virvou [2014]	0	0	0.5	1	0	0.5	0	0	0	0	2	20.0%
S05	Barrón-Estrada et al. [2011]	0	0	0.5	0.5	0.5	0.5	0	0	0	0	2	20.0%
	Average	0.82	0.45	0.88	0.98	0.65	0.61	0.45	0.64	0.27	0.06	5.82	58.2%
	Standard Deviation	0.35	0.51	0.22	0.09	0.36	0.27	0.51	0.44	0.38	0.24	2.19	22%

number of publications in 2014 with a similar number of papers in 2015, followed by a new decrease in 2016.

By analyzing Fig. B.2, it is difficult to point out that there is a research trend in the use of authoring tools to ITS design for non-programmer authors. Indeed, we can observe that researchers were concerned with the topic in different time frames, but we can not state that there is some kind of tendency. Note that, as the search process of this review was performed in June 2016, a decrease in the number of publications would be expected in this year because some papers might be in press or under submission/review process.

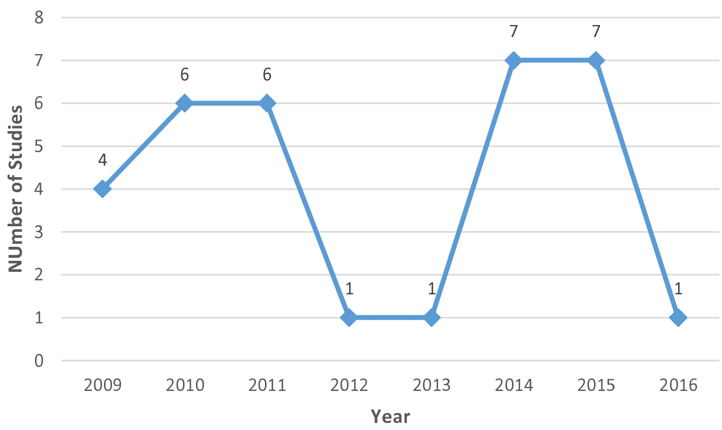


Figure B.2: Temporal view of the studies

B.3.2 Application context

The study settings were categorized either as an industry or academic context. Most the papers (31 studies) are considered academic, while 2 studies (S03 and S32) were conducted in an industrial setting. This result indicates that the application of authoring tools for designing ITS has been receiving much more attention from researchers than practitioners recently since only 6% of the papers are applied in an industrial context.

B.3.3 Type of source

The studies included in this review may be a journal, conference, workshop or book chapter publications. The majority of studies are conference papers (51.51%; 17 studies), followed

Table B.6: Distribution of studies over publication sources.

Publication Source	**Type**	**Count**	**%**
International Journal of Artificial Intelligence in Education	Journal	7	21.2%
International Conf. on Artificial Intelligence in Education (AIED)	Conf.	2	6.1%
International Conf. on Artificial Intelligence in Education (AIED) Workshops	Workshop	2	6.1%
International Conf. on Intelligent Tutoring Systems (ITS)	Conf.	2	6.1%
Advances in Intelligent Tutoring Systems	Book Ch.	1	3.0%
Annual Conf. on Behavior Representation in Modeling and Simulation (BRiMS)	Conf.	1	3.0%
Computers and Education	Journal	1	3.0%
Expert Systems with Applications	Journal	1	3.0%
IASTED International Conf. on Computers and Advanced Technology in Education (CATE)	Conf.	1	3.0%
Ibero-American Conf. on Artificial Intelligence (IBERAMIA)	Conf.	1	3.0%
IEE Students' Technology Symposium (TechSym)	Conf.	1	3.0%
IEEE Transactions on Learning Technologies	Journal	1	3.0%
International Conf. on Artificial Intelligence (ICAI)	Conf.	1	3.0%
International Conference on Foundations of Augmented Cognition	Conf.	1	3.0%
International Conf. on Information Technology: New Generations (ITNG)	Conf.	1	3.0%
International Conf. on Information, Intelligence, Systems and Applications (IISA)	Conf.	1	3.0%
International Conf. on Intelligent Interactive Multimedia Systems and Services (IIMSS)	Conf.	1	3.0%
International Conf. on Knowlege-Based and Intelligent Information and Engineering Systems (KES)	Conf.	1	3.0%
International Journal on Learning Technologies	Conf.	1	3.0%
Journal of Information Science and Engineering	Journal	1	3.0%
Knowledge-Based Systems	Journal	1	3.0%
Mexican International Conf. on Artificial Intelligence (MICAI)	Conf.	1	3.0%
Object-Oriented User Interfaces for Personalized Mobile Learning	Book Ch.	1	3.0%
World Congress on Information and Communication Technologies (WICT)	Conf.	1	3.0%

by journal publications (36.36%; 12 studies) and workshop and book chapter publications, each with 6.06% (2 studies).

Table B.6 presents the distribution of selected studies over publication sources, including the publication name, type, count (i.e., the number of selected studies from each source), and the percentage of selected studies. The 33 selected studies are distributed over 25 publication sources.

As shown in Table B.6, the leading venues in this study topic are the International Journal of Artificial Intelligence in Education (IJAIED), followed by the International Conference on Intelligent Tutoring Systems (ITS), International Conference on Artificial Intelligence (AIED) and the International Conference on Artificial Intelligence workshops. These results are expected since most papers on ITS are published by these communities, but might also indicate a positive aspect in the quality of the papers included in this review since the leading venues are high reputation vehicles in ITS research. However, a great number of the publications about the topic is widespread in different venues from computers and education, and artificial intelligence research areas.

B.3.4 Research method

The classification of publications was based on the categories (i.e., controlled experiment, quasi-experiment, case study, survey research, ethnography and action research) defined by Easterbrook et al. [2008]. However, we have defined two extra categories: illustrative scenario and not applicable. The first is appropriate for papers that just explain their contributions using small examples or argumentation. The latter refers to the papers that do not present any kind of research method or explanation of using the proposal.

Illustrative Scenarios (39.39%; 13 studies) constitute the majority of the studies, followed by Controlled Experiments (27.27%; 9 studies), Case Studies (15.15%; 5 studies), Not Applicable (15.15%; 5 studies) and Survey (3.03%; 1 study). There were no quasi-experiment, ethnography and action research papers in our classification.

Note that there are more non-empirical papers than empirical papers. Fifteen papers (45.45%) are concerned in conducting empirical studies (i.e., controlled experiment, case study, and survey) on the applications of authoring tools ITS design. The significant number of papers that conducted controlled experiments might indicate a recent maturity in the area about evaluating authoring tools since controlled experiments provide more reliable evidence about specific research hypotheses. However, the number of papers that do not perform any kind of empirical evaluation for their proposal is still high and deserves attention by the community.

B.4 RQ1: Authoring tools in ITS components

The purpose of this research question was to identify the main ITS components that have been supported by the use of authoring tools. In following we present the results and analysis and discussion of this research question.

B.4.1 Results

We categorized these components according to the well-known ITS components [Woolf, 2010]: domain model, pedagogical model, interface model and student model (see Table B.7). Most of the papers use authoring tools to design the *Pedagogical model* of ITS (81.82%; 27 studies) and *Domain model* (75.75%; 25 studies), followed by *Student model* (18.18%; 6 studies) and *Interface model* (15.15%; 5 studies). Note that a study could have met more than one ITS component, thus the sum of the percentages is greater than 100%.

Table B.7: Authoring tools in ITS components

ITS Component	Studies	Freq.	%
Pedagogical Model	S02, S03, S04, S05, S06, S07, S08, S09, S11, S12, S14, S16, S17, S18, S19, S20, S21, S22, S23, S24, S25, S26, S27, S28, S29, S32, S33	27	81.82%
Domain Model	S01, S03, S04, S05, S07, S08, S09, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S21, S23, S24, S25, S29, S30, S31, S33	25	75.76%
Student Model	S06, S10, S13, S16, S30, S31	6	18.18%
Interface Model	S04, S08, S23, S24, S27	5	15.15%

B.4.2 Analysis and Discussion

In summary, results shown in Table B.7 indicate that all classic ITS components are covered by the studies. The *Pedagogical model* is addressed by more than 80% of the studies. This result was somewhat expected since users of authoring tools are non-programmer authors that may intend to customize how learning process should take place in the ITS. The *Domain model* component also has a great number of studies (more than 75%). This result is also interesting because it shows that a great part of the studies are delegating or aiding authors in defining what should be learned by students using the designed ITS. Moreover, 19 studies (more than 57% of papers included) met both *Pedagogical model* and *Domain model* in the same paper, indicating the interest of using authoring tools not only to customize learning processes but also to allow the definition of content, problems and so on, according to learning processes defined. It may be worth noting that 6 other papers covered a combination of two different models, e.g., Domain Model and Student Model - 4 papers (S10, S13, S30 and S31), one (S06) covered Pedagogical and Student Model and another one (S27) - Pedagogical and Interface Model.

On the other hand, the use of authoring tools to design *Student models* and *Interface models* are not so much significant in comparison to other ITS components, respectively, 18.18% and 15.15%. For the case of student models, these results are expected since most of the papers are strongly relying only on the artificial intelligence features of tutoring systems to automatic represent student models during instruction, i.e., mainly using mechanisms such as overlay models and Bayesian networks. However, some works still allow authoring of the student model component enabling authors to configure student modeling rules. For instance, S10 presents an authoring tool that allows teachers to author different aspects of the student model for different categories of students. With respect to the authoring of interface models, we suspect that most of the authoring tools identified in the papers are relying on fixed tutor interfaces, which may not favor authoring of this component. Few works are allowing interface authoring, for example, in CTAT (S04), authors can design and create one or more tutor interfaces specific to the problem types for which the tutor will provide tutoring. Tutor interfaces can be built through drag and drop techniques within an existing interface builder, such as the Flash IDE.

Among all 33 studies, none of them addressed all four classic ITS components. Four papers (S04, S08, S23, S24) met at the same time *Domain model*, *Pedagogical model*, and *Interface model*. One paper (S16) met Domain, Pedagogical, and Student models. These results might suggest an opportunity to use ITS authoring tools to support the design of ITS considering the four main classic components. However, each component has its own function and unique properties which may be more or less amenable to authoring depending on several aspects, for instance, type of ITS, technologies used, needed pedagogical expertise, trade-off choices between usability and flexibility, and so on.

B.5 RQ2: ITS types

The purpose of this research question was to identify the main ITS types that have been developed by the use of authoring tools. In following we present the results and analysis and discussion of this research question.

B.5.1 Results

The classification of the ITS types was made after the data extraction of the studies, i.e., during the extraction, the ITS type addressed in the paper was identified according to the type explicitly stated by the authors. Next, in the syntheses step, the categories presented in Table B.8 were defined according to the distribution of the studies. Note that, even though an ITS could be classified in more than one category, we classified the study in the ITS type that is explicitly argued in the paper. We also defined some categories (i.e., *Content and problem-based* and *Machine and human-based*) according to ITS features discussed in the paper.

Table B.8: Authoring tools in ITS Types

ITS Type	Studies	Freq.	%
Model-Tracing/Cognitive Tutor	S08, S15, S16, S23, S26, S27, S28	7	21.21%
Example-Tracing	S03, S04, S12, S20, S21, S32	6	18.18%
Content and problem-based	S02, S10, S17, S19	4	12.12%
Dialogue-based	S07, S09, S25	3	9.09%
Constraint-based	S24, S29	2	6.06%
Machine and Human-based	S11, S22	2	6.06%
Non-specific	S01, S05, S06, S13, S14, S30, S31, S33, S18	9	27.27%
Total		33	100.00%

As shown in Table B.8, the predominant ITS types identified was *Model Tracing/Cognitive Tutor* (21.21%/ 7 studies), followed by *Example-Tracing* (18.18%; 6 studies), *Content and problem-based* (12.12%; 4 studies), and *Dialogue-based* (9.09%; 3 studies). *Constraint-based* and *Machine and human-based* have 6.06% (2 studies) each one. In nine studies (27.27%), we could not define a specific ITS type, thus they were categorized as *Non-Specific* type.

B.5.2 Analysis and Discussion

Model-tracing tutors contain a cognitive model of the domain that the tutor uses to check student responses. This model is based on a cognitive psychology theory of problem-solving and learning and is verified by the tutor in each step of the problem-solving process in order to maintain the student in the model path [Blessing et al., 2009]. Cognitive tutors are special trademark products that implements model-tracing tutors. They provide a problem-solving environment, including some features such as step-by-step feedback, messages in response to common errors, and instructional hints [Koedinger and Aleven, 2007]. Once these tutors are very similar, authoring tools targeting them are categorized in the same ITS type. This category includes studies which address the use of authoring tools for designing model-tracing tutors in all four ITS components (see Figure B.3), with an emphasis on domain and pedagogical model.

Example-tracing is also a significant ITS type identified in our results. This category includes studies on the domain, interface, and pedagogical models, but all of them are concerned with the pedagogical model, as seen in Figure B.3. Example-tracing tutors interpret and assess student behavior with reference to generalized examples of problem-solving behavior [Aleven et al., 2009b]. These examples intend to reduce the technical costs of tutor development by allowing domain experts and cognitive psychologists to build a cognitive model by demonstration rather than by programming a production rule model [MacLellan et al., 2014].

Content and problem-based category contains five studies and includes papers which mainly relies on authoring tools to author content and learning objects for ITS. Authoring tools categorized in this type basically target ITSs on which students intensively interact with some content and answer problems/tests in the tutor. For example, S02 describes an authoring tool that has been re-built for the Android OS. In this authoring tool, students have the possibility to read the theory offered by the mobile application, interact with it and take tests in order to evaluate his/her level of knowledge. The studies within this category addressed domain, pedagogical and student models.

Dialogue-based category is represented by three studies which propose to use authoring tools to design this type of tutor. The studies within this category rely on natural language processing mechanisms to provide a more natural tutoring with studies. These papers address pedagogical and domain models.

Constraint-based tutors are based on Ohlsson's theory of learning from performance errors and are designed to reduce the effort needed to develop a generic model of the domain (i.e., which is the case of model-tracing tutors) [Mitrovic et al., 2009]. It uses an evaluative model involving constraints defined over a set of pedagogically relevant solutions. The two studies within this category also addressed domain, pedagogical and interface models.

Machine and Human-based category is created to include studies that use authoring tools to design ITS which strongly relies on a machine and human intelligence in a complementary way during the tutoring process. The two studies within this category addressed only the domain and pedagogical ITS components.

The *Non-specific* category includes several tutors with distinct features. Papers are classified into this category if their authoring tool are specific enough to not deserve an own category. For instance, the ASSISTments platform (S18) provides a way to assist student while it assesses them. In this authoring tool, students find out immediately if they had the wrong answer to a problem allowing them to try again right away, whereas, teachers get assessment results in real time, which can be used to plan their next lesson, bring attention to misconceptions, and so on.

Note that two of the most frequent categories presented in Table 7 share a similar tutoring theory, i.e, Anderson's ACT Theory of Cognition [Anderson, 1983]. *Model-tracing/Cognitive Tutor* and *Example-tracing* ITS types are responsible for almost 40% of the total of papers included in this review. This result might happen due to the popularity of these tutors (i.e., CTAT) that provides several features for authoring these types of ITS for non-programmer authors. In fact, it is likely that the number of authoring tools is simply following the popularity of the ITS types they are targeting. Another result that deserves some attention is that almost 30% of the papers are categorized as *Non-specific*. This result may indicate that there is not a shared understanding in the ITS community of the underlying theories, technologies, and features of ITSs since many researchers are developing authoring tools for designing their own type of tutor.

To aid our analysis, Figure B.3 depicts the number of studies considering the ITS types over the ITS components. Note that the sum of the numbers of studies on specific ITS components exceeds the total number of studies within a specific category because one study could have been addressed by more than one ITS component. As presented in Figure B.3, the *Domain model* and *Pedagogical model* were addressed by all ITS types. The *Interface model* was met by *Model-tracing/Cognitive Tutor* and *Example-tracing* types as well as by the *Constraint-based* ITS type. The *Student model* was addressed by the *Model-tracing/Cognitive Tutor*, *Content and problem-based*, and *Non-Specific*.

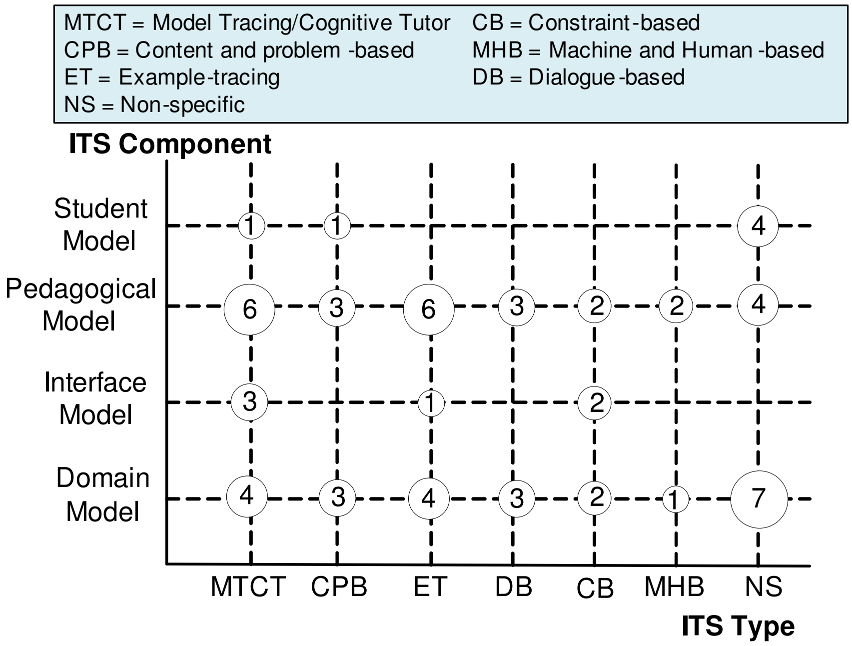


Figure B.3: ITS types over ITS components

These results may also suggest that some ITS types are more amenable than others to target ITS components. For instance, for a paper that presents an authoring tool for example-tracing tutors (e.g., S04), it might be more amenable than dialogue-based tutors to author interface models, since the former type of ITS has a flexible architecture that allows personalization of interfaces. In this way, ITS types may constrain the ITS components that authoring tools can address, but it is not clear how it happens and what components and other aspects should be considered when designing authoring tools for specific types of tutors.

Based on the categorization resultant from the analysis of this research question, we may classify the ITS type targeted in this thesis as a content and problem-based tutor. In the next chapter, we describe in more details the features that are considered in the design of the ITS type addressed in this work.

B.6 RQ3: Features for aiding authoring process

This question intends to identify the features provided by the authoring tools that aid the authoring process. In following we present the results and analysis and discussion of this research question.

B.6.1 Results

As well as in RQ1, we identified the categories by classifying the studies after the extraction step. A study could also have met more than one feature, thus the sum of the percentages is greater than 100%. As presented in Table B.9, we identified 21 categories for the studies. The *Not Applicable* category was defined to classify papers we could not identify any special feature to aid authoring process as well as papers that do not present a new authoring tool, i.e., they use or evaluate authoring tools proposed by other authors.

Table B.9: Features for aiding authoring process

ITS Component	**Feature/Facility**	**Studies**	**Freq.**	**%**
Student Model	Define students stereotypes	S10, S16	2	6.06%
	Authoring based on learning styles	S06	1	3.03%
	Reuse of students' profiles	S31	1	3.03%
Pedagogical Model	Define/Give feedback	S02, S04, S07, S11, S19, S20, S22, S23	8	24.24%
	Define behavior graphs	S04, S21, S32	3	9.09%
	Make assignments	S03, S11, S14	3	9.09%
	Define cognitive model	S08, S32	2	6.06%
	Define collaboration scripts	S26	1	3.03%
Interface Model	Drag and drop interface authoring	S04, S26	2	6.06%
Domain Model	Define problem solutions	S07, S11, S14, S23, S24, S29	6	18.18%
	Authoring by demonstration	S04, S11, S19, S21, S23	5	15.15%
	Automatic domain model generation	S09, S23, S24, S29	4	12.12%
	Define hints	S04, S15, S19	3	9.09%
	Reuse of learning content/domain model	S01, S13	2	6.06%
	Human Computation	S25	1	3.03%
General	View learners' statistics	S02, S03, S30, S31	4	12.12%
	Mobile authoring	S02, S30	2	6.06%
	Reuse/Export tutor design	S13, S30	2	6.06%
	Create class lists	S03	1	3.03%
	Not applicable	S05, S12, S27, S28	4	12.12%

The most frequent feature identified was the *Define/Give feedback* (24.24%; 8 studies), followed by the *Define problem solutions* (18.18%; 6 studies). Five studies (15.15%) provided the *Authoring by demonstration* feature. The features *Automatic domain model generation* and *View learners' statistics* feature are both presented by four studies each one (12.12%). The *Define behavior graphs*, *Make assignments* and *Define hints* are provided by three studies, each one with 9.09%.

The *Define cognitive model*, *Reuse of learning content/domain model*, *Define students stereotypes*, *Drag and drop interface authoring*, *Mobile authoring* and *Reuse/Export tutor design* features are included by 2 studies (each with 6.06%). We also found several features presented in only one study (3.03%): *Authoring based on learning styles*, *Create class lists*, *Define behavior graphs*, *Define collaboration scripts*, *Define hints*, *Human computation*, and *Reuse of students' profiles*. Four papers (12.12%) were categorized as *Not applicable*.

B.6.2 Analysis and Discussion

The results of this research question show a plethora of features that have been considered to aid authoring decision-making process. In following we depict the function of each feature and present how the feature is supporting the authoring tools presented by the papers. In the end of this section, we discuss these results. As expected, most of these features are related to the Pedagogical and Domain models since most of them are designed to assist authors in defining pedagogical instruction as well as to aid authors to define learning objects to be used in the authored ITS. However, as shown in Table B.9, there are also some features related to the Student and Interface models as well as features related to general aspects of authoring tools.

As seen in Table B.9, the features *Define students stereotypes*, *Authoring based on learning styles*, and *Reuse of students' profiles* are targeting the student model component. The first feature allows teachers to define student stereotypes by defining characteristics that are used by agents to generate different courseware plans for each stereotype defined [Chakraborty et al., 2010]. The second feature aids authors to design student models based on a learning style model (i.e., the Felder-Silverman model [Felder and Silverman, 1988]) that classifies students according to where they fit on a number of scales pertaining to the ways they receive and process information. The last feature let teachers updating a student's profile by interacting with the system, namely pressing buttons, choosing from a drop-down list and picking one from given multiple choices. This feature also offers the possibility to the teachers to register a new student so that s/he is able to make use of the system and learn multiple languages [Virvou and Troussas, 2011].

The features *Define/Give feedback*, *Define behavior graphs*, *Make assignments*, *Define cognitive model*, *Define collaboration strategy* and *Diagnose student solutions* are supporting users to author pedagogical model. The first one is basically the function that enables authors to define some kind of feedback in the authoring tool to be given to students during instruction. The second feature is frequently used in example-tracing tutors. In this feature, an author can create different ways of solving a problem that is captured as different paths in a behavior graph. Next, the author may generalize the graph to indicate the range of student behavior that the graph stands for [Aleven et al., 2016]. The third feature allows authors to create assignments specifically to adjust the students' learning behavior, for instance, S14 enables teachers to make assignments after diagnosing students' learning errors. In order to lower the bar in creating the cognitive model of model-tracing/cognitive tutors, the *Define cognitive model* feature aims to allow non-programmer authors to create the intelligence behind these types of tutors, or at least modify in a meaningful way an already produced cognitive model [Aleven et al., 2009a]. Finally, using the *Define collaboration scripts* feature, authors can develop collaborative ITSs with embedded collaboration scripts, so that features that support effective collaboration can be intertwined with those that support problem-solving [Olsen et al., 2014].

The features *Define problem solutions*, *Authoring by demonstration*, *Automatic domain model generation*, *Define hints*, *Reuse of learning content/domain model*, and *Human computation* are addressing the Domain model. The function of the first feature is to allow authors to enter (before tutor instruction) into the authoring tool, the solution of problems that are given to students. The next feature is mainly used in a special type of cognitive tutor (e.g., SimStudent [Matsuda et al., 2015]) and enables authors to demonstrate solution steps, and, in the meantime, the authoring tool attempts to induce underlying domain principles by generalizing those worked-out examples. In the third feature, the authoring tool provides a way to automatically generate elements of the domain model of a tutor. For instance, S24 and S29 use constraint-generation algorithms to produce constraints that verify the syntactic validity of solutions. Similarly to the first feature, the *Define Hints* feature enables authors to create and associate hints to problems of ITS. The next feature supports the reuse of existing learning content from other tutors in the same domain of the tutor being authored. Finally, we found a work (S25) that uses human computation – i.e., a subfield of computer science on which studies represent computationally difficult tasks so that humans will be motivated to work on them [Olney and Cade, 2015] – to motivate authors in creating ITS.

We have found only one feature that is supporting users to author the interface model component. As seen in Table B.9, S04 and S26 support drag-and-drop interface building to author the interface model of their tutors. We also identify some features that are targeting general aspects of authoring tools. As seen in Table B.9, the features *View learners' statistics*, *Reuse/Export tutor design*, *Mobile authoring*, and *Create class lists*. The first one is basically supporting authors to check learners' statistics in the authoring tool, for instance, students' performance in the tutor, interaction with the tutor, and so on. The second feature enables authors to reuse or export previous authoring decisions in a new tutor. This feature saves author time in designing new tutors as well as may favor reuse of already validated tutors. We also identified the mobile authoring feature, which enables authors to design ITS in mobile devices (e.g., S30). Last but not least, using the *Create class lists* feature, teachers can create class lists in order to assign work to an entire class or an individual student and view reports of their students' progress.

One might note that the identified features are much more focused to aid authors in aspects regarding domain (6 features), pedagogical (5 features) and more general purpose (4 features). Whereas, as previously mentioned, few authoring tools have been presenting facilities to enable student (only 3 features) and interface (1 feature) authoring. Particularly, it is possible that researchers are, in general, considering the tradeoff between flexibility and usability to decide whether to incorporate or not features for authoring interface model. We suspect that the extra effort needed to author ITS interfaces has a higher weight over the potential flexibility benefits that could be given to authors.

Another result that deserves to be highlighted is the significant number of features related to general aspects (e.g., *View learners' statistics* and *Reuse/Export tutor design*) of authoring tools. As shown in Table B.9, our results suggest that researchers are also interested in providing more powerful authoring tools in order to support authoring beyond traditional

ITS components.

Note that there might be a direct relation between the number of papers that address specific features and particular kinds of tutors that are more targeted by authoring tools. For example, as previously presented, example-tracing tutors are addressed by more papers than constraint-based tutors, thus, it is expected a higher frequency in the number of features that are commonly provided by example-tracing tutors (e.g., *Define behavior graphs*).

B.7 RQ4: Authoring technologies

The purpose of this research question was to identify the main technologies used to build authoring tools as well as the problems that such technologies are intending to address. In following we present the results and analysis and discussion of this research question.

B.7.1 Results

In order to classify the studies, we have clustered them according to the type of technology used in the work. The classification of such technologies was made after data extraction, by analyzing and grouping the technologies reported in the papers.

As shown in Table B.10, most of the papers (39.39%; 13 studies) are using artificial intelligence technologies, concepts or theories to address different kinds of problems within ITS authoring tools (e.g., to support domain knowledge representation, to enable intelligent tutoring, and so on). Moreover, eleven studies (33.33% of the total) are using specific tools, platforms, frameworks or plugins to address software engineering problems related to the construction of ITS by using authoring tools, for instance, faster the development of tutors, enable the extensibility of ITS, etc. Three papers (9.09%) use technologies from the distributed systems subarea in order to address interoperability problems regarding ITS. In eight papers (9.09%) we could not identify any specific technology, hence they are classified in the *Non-specific* category.

Table B.10: Technologies used to build authoring tools

Technology	Studies	Freq.	%
Tools, platforms, frameworks or plugins	S03, S04, S08, S09, S15, S17, S20, S21, S23, S26, S32	11	33.33%
AI technologies, concepts or theories	S01, S05, S06, S07, S10, S14, S16, S23, S24, S25, S28, S29, S33	13	39.39%
Distributed systems technologies	S02, S04, S30	3	9.09%
Non-Specific	S11, S12, S13, S18, S19, S22, S27, S31	8	24.24%

B.7.2 Analysis and Discussion

The results of this research question may be analyzed from the research background on which the technology belongs as well as by identifying particular technologies used in the papers and the problems they are targeting.

As seen in Table B.10, 39.39% of the papers are using some kind of AI technology, concept or theory. Ontologies are used by the papers S01, S16, S24, S28, and S29 to mainly support domain knowledge representation. These works are aiding authors in defining the domain model of tutors as well as relying on the reasoning and inference capabilities provided by ontologies to effectively use the domain model during tutoring. Particularly, S16 uses semantic networks, which is more focused on a visual notation to represent knowledge. It also uses intelligent agents arguing that agents can make a good choice to adapt courseware elements to students since they have abilities to learn, personalize and adapt, allowing to manage new situations and providing pedagogically appropriate courseware presentation. Machine learning is also used by four papers, in which S05, S06, and S33 use specific algorithms based on neural networks to address different kinds of problems. The first one is using this AI technique to implement emotions recognition in the tutor supported by its authoring tool, whereas the others two use it to automatic provide to authors discovered features based on several patterns from students (e.g., learning style, students' grades in the course, and so on). Furthermore, S23 developed a machine-learning solution, called SimStudent to help novice authors to create cognitive tutors. This tool is integrated into CTAT and helps authors to create an expert model for a cognitive tutor by tutoring it on how to solve problems [Matsuda et al., 2015]. Moreover, S07 and S25 relies on natural language processing techniques for improving the authoring of natural language ITS. For instance, S07 proposes the ConceptGrid authoring tool that intends to check sentence-length natural language answers by using natural language processing. S10 supports pedagogical model authoring by using a fuzzy rule-based strategy, it allows authors to configure the rule-base and define the teaching strategy, which is represented by the rules. The last work in this category (S14) uses hierarchical classification to perform students' diagnosis receiving as input some types of information (e.g., hierarchy of learning errors) from authors.

With respect to the eleven papers that are addressing software engineering issues of ITS authoring tools, they include works that propose tools (e.g., CTAT), platforms (e.g., Ambre-Add) or frameworks (e.g., GIFT and Tutor Runtime Engine) to support ITS development. Most of the papers included in this category are using CTAT (S03, S04, S20, S21, S23, S26, and S32), which is, to best of our knowledge, the most advanced solution reported in the literature to develop different types of ITS (i.e., cognitive and example-tracing tutors). CTAT mainly target the problem of supporting non-programmers authors to efficiently and cost-effectively develop ITS capable of capturing sophisticated tutoring behaviors that are effective in helping students learn in a wide range of domains [Aleven et al., 2016]. In other direction, S09 integrates their own authoring tool (called TRADEM) with components of the GIFT framework. It uses the domain module and the engine for providing the pedagogical model from the GIFT framework in order to allow authoring of these components in their tool. S08 is using the Tutor Runtime Engine (TRE), which is a representation of a tutor delivery environment, in order to provide a clear separation between student's interface and the underlying cognitive model that provide the tutoring. This technology intends to enable the integration of third-party interfaces with the tutor-generated by the authoring tool. In addition, in a manner similar to what it did for the TRE, S15 uses the Tutor Link plugin to make an existing tool (called xPST) extensible to serve as an intermediary between third-party applications and the xPST Engine. It knows how to map actions in the interface to the proper pieces in the tutor model and how to display hints and other tutoring information within the application [Gilbert et al., 2015]. We also identified a paper (S17) that enable teachers to adapt a specific tool (i.e., AMBRE-add) in order to act on how the ITS automatically adapts itself to the profile of the student.

Only three papers are using technologies related to the distributed systems area. These works (S02, S04, and S30) are relying on the use of Web Services to enable interoperability between different architectures used in the ITS authoring tool. For instance, S30 uses this technology to allow interoperation between mobile devices used by authors and an ITS architecture.

It is worth noting that many ITSs are actually not that tied to AI – i.e., they focus on VanLehn's inner and outer loops [Vanlehn, 2006], and may be based on more simple mechanics. In this way, we believe that this is a possible reason for why some papers are not intensively relying on AI technologies to address ITS authoring tools. However, as we could not identify the explicit use of technologies in eight papers (categorized as Non-Specific papers), we can not say that the papers that are not included in the AI category are indeed not using artificial intelligence in the research.

Moreover, with respect to the results found in this research question, one might note that the use of CTAT is remarkable. CTAT-built tutors have been demonstrated to be robust for use in real educational settings over a wide range of projects [Aleven et al., 2016]. Many reasons could explain why CTAT is much more popular than other authoring tools. For example, in order to create example-tracing tutors, it provides facilities allowing authors to create graphical user interfaces, to generate behavior graphs, and to aid the deployment of components in structures that executes the example tracer algorithm. Moreover, many researchers are contributing to extending and improve CTAT in several different situations.

We can also discuss the technologies from the perspective of using or not web technologies. As mentioned in our analysis, only seven studies (21.21% of the total) are strongly relying on web technologies, i.e., ontologies (S01 S24, S28, and S29) and web services (S02, S04, and S30), while 26 studies (78.79%) are not explicitly using web technologies on their works. This result might show that there is still space for improving existing ITS authoring tools to take advantage of the web technologies capabilities, for example, interoperability, distribution, portability and so on.

Furthermore, we can also discuss the results of this research question by analyzing the technologies' subareas over the ITS components and types identified in our previous research questions. Figure B.4 presents a bubble plot to aid in this analysis. As seen in the figure, AI technologies, software solutions, and distributed technologies are targeting the domain, pedagogical, and interface models. As expected, there is an emphasis on the first two components since these two components are much more targeted by the papers in general (see Table B.7). Note that our results suggest that the papers related to software solutions are not addressing the student model component. This result is interesting but not surprising since it is expected that authoring tools are relying on the automatic student modeling representation of intelligent tutors (on the learner's side) to target this component.

As shown in Figure B.4, our results might indicate that authoring tools targeting example-tracing tutors are more concerned with providing software solutions to construct tutors than relying on specific artificial intelligence techniques. Note that this result does not imply that these authoring tools are not using artificial intelligence since there is a paper identified in the Non-specific category that may use or not it. These results can also suggest that the dialogue-based ITS type is more supported by artificial intelligence, which can be explained by the fact that these kinds of tutors are strongly tied to AI techniques, for example, natural language processing. Moreover, as seen in Figure B.4, our results indicate that constraint-based tutors are mainly supported by AI techniques. This result may be correlated with the results of our previous research question since we could identify a feature (i.e., Automatic domain model generation) dependent on AI algorithms that are used by the two papers targeting this type of ITS. One might also note in Figure B.4 that the authoring tools which address machine and human-based tutors are only using non specific technologies, which can suggest that these tools are investigating non-conventional technologies to contribute to the development of such type of tutor.

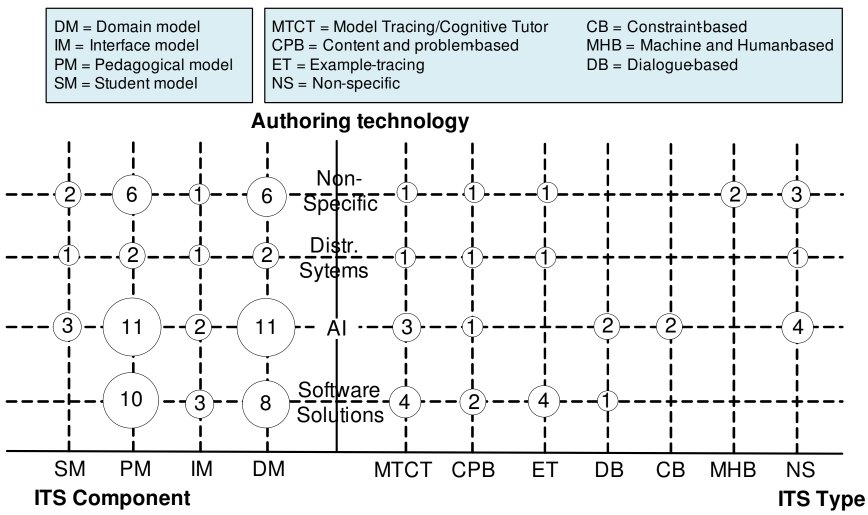


Figure B.4: ITS components and types over technologies subareas