Logicamente: A Virtual Learning Environment for Logic Based on Learning Objects

Patrick Terrematte, Fabrício Costa, and João Marcos

Federal University of Rio Grande do Norte (UFRN)
Department of Informatics and Applied Mathematics (DIMAp)
Group for Logic, Language, Information, Theory and Applications (LoLITA)
Natal – RN – Brazil
terrematte@ppgsc.ufrn.br, fabriciocscte@ppgsc.ufrn.br,
jmarcos@dimap.ufrn.br

Abstract. Logic is a subject connected to several fields of study, by which it is possible to improve the understanding of information and the reasoning process in many domains. In most courses, it is remarkable how Logic represents a pedagogical challenge for both tutors and pupils, and the recorded number of cases of failures and of discontinuity is often high. One of the reasons for this situation is the gap between, on the one hand, the repetitive aspects of exercises for learning and, on the other hand, the inventive activities of researching or applying Logic to practical situations. Given the need to provide a solid basis for the subject at undergraduate level, and also to focus on inductive learning with creative skills, we propose the project LOGICAMENTE[1], a Virtual Learning Environment (VLE) for Logic composed of a growing collection of Learning Objects combined with the respective learning scripts, expositions, tasks and activities on subjects of Logic. The VLE illustrates fundamental concepts and algorithms from Logic, as well as allows students to conduct interactive experiments involving the understanding of various logical concepts belonging to topics ranging from Theorem Proving to Formal Semantics.

Keywords: Learning Objects, Teaching Logic, Logic.

1 Introduction

"Contrariwise," continued Tweedledee*, "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic."*
— LEWIS CARROLL
Through the Looking-Glass (1871).

Tutors of undergraduate Logic classes are often faced with problems directly linked to causes of failures and discontinuity of their pupils, such as:

– *The laborious application of theoretical subjects*: little visualization is provided for logical concepts, such as the construction of proofs, the recursive properties of language, and the construction of models.

[1] Available at http://www.lolita.dimap.ufrn.br/logicamente

P. Blackburn et al. (Eds.): TICTTL 2011, LNAI 6680, pp. 223–230, 2011.

– *The difficulty of a creative approach for logical skills*: some very significant aspects in Logic, such as proof search, the use of lemmas, the creation of definitions, and the search for counter-models, are unfairly treated as a repetitive matter.

To address such issues, many tools for teaching Logic are currently available, such as those presented in the comprehensive list compiled by the Association for Symbolic Logic [4]. Many of them are inspiring and widely used, e.g. *AproS Project*, *JAPE*, *ProofWeb*, *Tarski's World*, *Fitch* and *Boole*, among others. Such systems are appropriate to deal with the topics they are aimed at, but they cover, in isolation, very restricted areas of Logic, as only one or two topics are explored by each system. The Logicamente intends to be a multi-subject Virtual Learning Environment (VLE) for the teaching and learning of the various facets of Logic. The basic idea of the project is to conceive an extensible framework to teach Logic without the following very common problems of e-learning tools:

– *Technological constraints:* Some tools are desktop systems, so they are not available at *'anytime and anywhere'*.
– *Usability of e-learning:* The tools don't have usability patterns to stimulate different learning styles of the users.
– *Integrated learning environment:* They do not offer the possibility of managing the feedback of students, their activities and exercises.
– *Collaborative development:* They do not offer any warranty of continuous development. Big and complex projects, however, must offer collaborative strategies.

A VLE is a computer system designed to support teaching and learning with an educational monitoring mechanism. A VLE should provide a set of tools, such as assessment tools, interactive communication, transfer of content management, student activities, tools, tracking exercises, forums, wikis, and learning objects specified for selected subjects. For the implementation of those features that compose a VLE, the Logicamente is to employ a module integration with Moodle, a well-known Learning Management System (LMS) that helps in organizing contents and educational activities. The advantages of using this platform is the possibility of reuse of specified Learning Objects and the management of activities, aiming at a constructive process for teaching and learning Logic.

In the paradigm of object-oriented computing, the compounds are developed to be reused and to solve the same recurring problem as applied to different contexts. An analogy may be employed to define the Learning Objects (LO) as units which provide an epistemological content to stimulate the reflection of a student about a given subject: an LO is any digital entity with a defined educational purpose that can be used, reused or referenced during a learning process [10,9].

The Logicamente implements the concept of LOs to customize contents and methodologies of Logic courses according to their focus. It also offers learning scripts with expositions, tasks and activities on subjects of Logic, in order to describe the epistemic content of each LO, guide the teaching and learning of Logic, what is intended to be learned, which skills should be developed, and the use of each LO with its purposes and values. The LOGICAMENTE represents an integrated learning environment which associates a context of activities (tasks or exercises) and a specific content of Logic to a Learning Object.

The LOGICAMENTE is being developed based on three senses of collaboration:

- *Collaborative learning*: We aspire to support the development of LOs that may be operated and reused by different VLE platforms and Learning Management Systems. The LOs are implemented by undergraduate students learning concepts of logic and applying them algorithmically.
- *Collaborative development*: We strive to apply a set of strategies with a development process based on the *open source* methodology, documenting and creating a *Web Service* interface to support the interaction for other tools and projects collaborating with the LOGICAMENTE .
- *Collaborative integration*: We design the LOGICAMENTE to be integrated with other systems and projects. For instance, we develop an LO —the *Theorem Proving Web System*— integrated with the system *ProofWeb*[2].

2 The Virtual Learning Environment: Logicamente

C'est par la logique qu'on démontre,
c'est par l'intuition qu'on invente...
— JULES HENRI POINCARÉ
In Science et méthode (1908).

The first prototype of the LOGICAMENTE was assembled in 2006 as a collection of final assigments for an undergraduate course on Logic Applied to Computing. The goal was to apply a methodology of Problem-Based Learning (PBL) to the implementation of some key concepts of Logic. In the following semesters, the project continued to be developed in collaborative learning through the work of students and volunteers, who had the opportunity to do research that explored the computational aspects of the algorithmic implementations as well as aspects of computer technology applied to education [13,2,12]. Currently, the project involves master's students and undergraduate students in its development. The prototype of the home screen can be seen in Figure 1.

The LOGICAMENTE is developed aiming at exploring the advantages of Web applications, implementing LOs to perform tasks such as: automatic generation of formulas with the desired complexity; configuration of a language and the definition of new connectives; translation between different syntaxes; construction of truth-tables; interactive presentation of formulas in the form of trees; implementation of the resolution method for Classical Logic; search for (counter-)models; use of a proof assistant for the practice with writing derivations in a formal deductive system.

[2] The *ProofWeb* is an *open source* software for teaching Natural Deduction which provides interaction between some proof assistants (*Coq, Isabelle, Lego*) and a Web interface [6]: Check `http://proofweb.cs.ru.nl/`

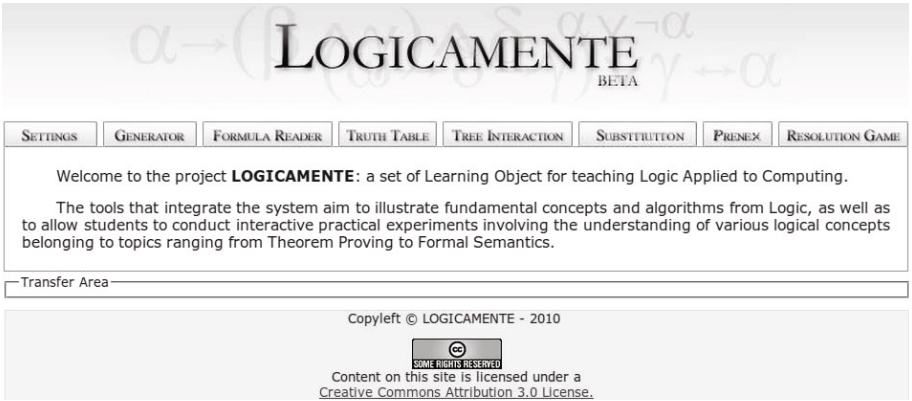

Fig. 1. The current version of the LOGICAMENTE

2.1 Logicamente on e-learning: Learning Objects and Modules

Nowadays, with the advancement of the Web, knowledge is related to the dynamic way in which we have access to it, searching it on-line or collaboratively constructing it in social networks. It is necessary to rethink the way by which we teach and adapt to this new dynamics on e-learning. It is still difficult, nonetheless, to devise a fair assessment method based exclusively on the e-learning paradigm. It is essential thus to identify and work with both the main current educational paradigms, *face-to-face* and *e-learning*. In agreement with Antonia Huertas [7], we believe that teaching activities for logic must be accomplished through both paradigms. Our proposal is to assign practical activities on the e-learning modality and to assign theoretical activities on the face-to-face paradigm. The e-learning is meant to explore the practical abilities of students and even help to emphasize the interactive and creative skills. However, the success of the e-learning modality does not depend only on techniques applied by most VLEs. To take the feedback from the students into account and maintain a high level of motivation for them, the main methodologies that we apply to the project LOGICAMENTE are instrumental, constructionist, conceptual modeling, personalization of the learning process and continuous evaluation process [1].

With the LOGICAMENTE , the tutor handles students, classes, lessons and exercises. Besides, the tutor also has access to activity reports of performances of students, and there is an agent for monitoring the learning modules and students' activities in interacting with the LOs. In turn, the student accesses exercises related to pre-defined learning scripts and is evaluated by his performance with personalized tasks.

The LOs that have been implemented aim at clarifying the relationships between different approaches to Logic, such as Formal Semantics and Theorem Proving. These different subjects have their contents associated to activities and challenges offered in MOODLE, aiming at a constructive process of learning Logic.

The LOGICAMENTE has two internal modules to support the translation between syntaxes and the interoperability between other learning tools and other programming languages used in the development. The following LOs are very basic and aim at helping both in the production of practical activities for the students and in the collaborative development of our tool:

• *Formula Translator*: This module takes care of the syntax for formulas and can represent their constructors using different symbols, images, HTML codes, Unicode or LaTeX. Also with this module it is possible to uniformly translate between formulas of different applications and to provide an interface of Web-services for the flexible interaction between distinct logic tools.

• *WFF Generator*: This module is responsible for generating well-formed formulas according to certain specifications (occurrences of connectives, sentential variables, and others). With this module, sets of formulas are generated according to the desired complexity and they may be used by other modules in producing personalized exercises.

2.2 The *(Non)Sequitur* Super-Module

One of the important contributions of LOGICAMENTE to produce challenging activities for stimulating the learning of Logic is the super-module *(Non)Sequitur*, where several basic LOs are combined in order to implement a learning script aimed at cultivating an important skill that will be described in the following.

In learning Logic one of the great difficulties lies in the distinctions and the between Proof Theory and Model Theory. The enlightenment in these topics is essential to the understanding of the metatheoretical results of soundness and completeness. In verifying whether an arbitrary argument represents a valid deduction in Classical Logic, most students are attempted to try proving the affirmative answer, instead of finding counter-models to the given assertion. But this is unrealistic, because in an arbitrary such a situation a falsifiable argument, with counter-models, is more likely to be detectable than a verifiably correct argument.

In order to improve the conceptual basis on Classical Logic we offer the students a module composed of five LOs —*Resolution Game*, *Theorem Proving Web System*, *Small Counter-Model Builder*, *Interactive Model Builder* and *Semantic Consequence Tester*— where they may choose between proving or finding a counter-model for an argument or a formula, automatically generated by the *WFF Generator*. The following LOs are responsible for constructing proofs:

• *Resolution Game*: Implements the resolution method, assisting the student in eliminating complementary literalsThe resolution method is used for deciding whether a sentence in conjunctive normal form is a theorem. In our interactive version of this method the student may choose which step she wishes to take in converting a given arbitrary formula to the adequate normal form, as well as in applying resolution at her discretion to the resulting collection of clauses.

• *Theorem Proving Web System*: implements a proof assistant in Natural Deduction. This module is developed by integrating the *ProofWeb* [6] to our project.

To analyze an argument semantically, the student can interactively search for counter-models or justify the correction of her own models, using the following LOs:

• *Interactive Model Builder*: Implements a counter-model searcher for propositional arguments, allowing for valuations to be proposed and stepwise justified. The main goal is to allow the user to freely choose her heuristic for argument analysis.

• *Small Counter-Model Builder*: Combines a bounded model checker with a search engine for finite counter-models of *first-order arguments*, receiving as input a limited number of objects, a collection of assumptions, and a goal formula as conclusion. This module partially simulates the functionality of the tool *Tarski's World*, in which the evaluation of formulas is made with a world's state composed by a finite stock of geometric forms, distributed over a finite board.

• *Semantic Consequence Tester*: Here the student may insert sets of sentences and verify step by step a classical entailment relation involving these formulas. The module may help in generating or justifying a counter-model whenever this is the case.

The aim of the super-module *(Non)Sequitur* is to provide a collection of different tools for evaluating arguments. Through the LOs *Resolution Game* and *Theorem Proving Web System*, students might attempt to justify validity using natural deduction arguments, on both a propositional and a first-order level. For falsifiable arguments, the student should choose to evaluate the arguments through the LOs *Interactive Model Builder* and the *Semantic Consequence Tester*, or using *Small Counter-Model Builder* to seek appropriate (counter-)models for given first-order arguments.

The main methodology used in these LOs is to promote the exploration of different strategies. With the *(Non)Sequitur* module, we offer the possibility for the user to try her hand with both the natural deduction rules and the semantic interpretations, and learn from her own choices and mistakes, being free to choose between proving or finding a counter-model. The methodology of learning from mistakes organizes the information in an inductive model of learning, where there are facts and observations given for inferring principles, as opposed to a deductive model of learning, where applications are consequences of given principles. The inductive learners "do not feel comfortable with the 'Trust me — this stuff will be useful to you some day' approach" [5, p. 6], and they play a major role among computer science students, faced with curricula organized along deductive lines.

2.3 The Modeling and Implementation of Logicamente

In many software projects, the main failures are caused by problems related to usability and scope of system, for instance the lack of user involvement with the software and the fact that the requirements implemented are incomplete and unsatisfied. For instance, to ensure that the users will be satisfied, the

(Non)Sequitur module has an interface designed and focused on the user with an inductive learning style in opposition to a deductive and procedural learning.

The LOGICAMENTE is implemented as a Web application programmed in PHP with AJAX technology in order for the system to be available for use anywhere and without requiring a previous local installation. Aiming at the reuse of LOs, the LOGICAMENTE uses the standard IMS Learning Design (IMS-LD) to describe the LOs with their specified content, and to maintain its resources, activities, tasks or exercises [8,3]. The IMS-LD is compatible with MOODLE and is a framework used for designing courses, supporting activities and organizing VLEs.

3 Future Works and Final Remarks

> "*La puissance de vision qui fait le poéte,*
> *et la puissance de déduction qui fait le savant,...*"
> — HONORÉ DE BALZAC
> *La Recherche de l'Absolu (1834).*

Many computational tools are currently available for the teaching of Logic. Those applications, however, are not organized into Virtual Learning Environments in an infrastructure that is really integrated and customized with Learning Objects. In the present note, we explained how the project LOGICAMENTE was conceived to fit that role, through the collaborative work of students and with a the problem-based approach to the learning of concepts of Logic. We have described some of the main implemented Learning Objects and finally we have also briefly explained our methodological approach to e-learning.

Particularly in the case of tools for teaching Logic, an inspiring example of collaboration is found with the *AproS Project*[3], a project that has been refined for roughly twenty years and developed with the help of colleagues and students [11]. The *AproS Project* implements a proof display and has a Web-based course *Logic & Proofs* for natural deduction proofs associated with the Open Learning Initiative (OLI) and an interactive learning environment. The differences between LOGICAMENTE and *AproS* lie on our initial challenging implementation by students, our focus being purely for the Web, the collaborative development, the integration with MOODLE, and our approach based on super-modules that implement learning scripts based on specific pedagogical proposals. As we have shown, this is the case of the *(Non)Sequitur*, for instance, a super-module designed to take several distinct concepts from Logic into account and provide a creative approach to their contrast and their practice.

A common problem with software development is the lack of collaborators and stakeholders to continue developing and maintaining the systems. To address such difficulties, the LOGICAMENTE is being built incrementally so that new features may be added gradually. The next challenge is to keep the development active in a collaborative way, ensuring the continuous improvement and growth of LOGICAMENTE . With the *open source* methodology, we aim to facilitate the integration with the work of collaborators from elsewhere.

[3] Check `http://automath.org/`

Acknowledgment. The authors have been partially funded by CNPq and CAPES. Their gratitude is also expressed to all undergraduate students of Computer Science and Computer Engineering who have contributed to the project Logicamente during several semesters of the course of Logic Applied to Computing at DIMAp/UFRN.

References

1. Bannan-Ritland, B., Dabbagh, N., Murphy, K.: Learning Object systems as constructivist learning environments: related assumptions, theories, and applications. In: Wiley, D.A. (ed.) The Instructional Use of Learning Objects, pp. 61–98. AECT, Bloomington (2002)
2. Barros, T.M., Araújo, A.E.F.D., Marcos, J.: A implementação colaborativa de uma suíte de ferramentas on-line de apoio ao ensino de Lógica. VIII ERMAC–R3 (2008)
3. Beauvoir, P., Griffiths, D., Sharples, P.: Learning Design Authoring Tools in the TENCompetence Project. In: Koper, R. (ed.) Learning Network Services for Professional Development, pp. 379–387. Springer, Heidelberg (2009)
4. van Ditmarsch, H.: A comprehensive list of tools for doing Logic. Association for Symbolic Logic (2010), `http://www.ucalgary.ca/aslcle/logic-courseware/` (verified at March 21, 2011)
5. Felder, R.M., Silverman, L.K.: Learning and teaching styles in Engineering education. Engr. Education 78(7), 674–681 (1988)
6. Hendriks, M., Kaliszyk, C., Van Raamsdonk, F., Wiedijk, F.: Teaching logic using a state-of-the-art proof-assistant. Acta Didactica Napocensia 3(2), 35–48 (2010)
7. Huertas, A.: Teaching and learning Logic in a Virtual Learning Environment. Logic Journal of IGPL 15(4), 321–331 (2007)
8. Koper, R., Miao, Y.: Using the IMS LD Standard to Describe Learning Designs. In: Handbook of Research on Learning Design and Learning Objects: Issues, Applications and Technologies, pp. 41–86. Information Science Publishing (2009)
9. Polsani, P.: Use and abuse of reusable Learning Objects. Journal of Digital information 3(4) (2006)
10. Ritzhaupt, A.D.: Learning Object Systems and Strategy: A description and discussion. Interdisciplinary Journal of E-Learning and Learning Objects (IJELLO) 6, 217–238 (2010)
11. Sieg, W.: The AproS Project: Strategic thinking & computational logic. Logic Journal of IGPL 15(4), 359–368 (2007)
12. Terrematte, P., Marcos, J., Galdino, T.: O Logicamente: A implementação colaborativa de Objetos de Aprendizagem de Lógica (IV WAPSEDI). In: SBIE - XX Simpósio Brasileiro de Informática na Educação (2009)
13. Vilela, G., Rosan, M., Marcos, J.: Implementação de um gerador e verificador de modelos finitos para a Lógica Clássica de Primeira Ordem. VIII ERMAC–R3 (2008)