

# The LOGICAMENTE: collaborative challenges and benefits of a tool for teaching and learning Logic \*

Patrick Terrematte<sup>1</sup>, Fabrício Costa<sup>1</sup>

<sup>1</sup> Federal University of Rio Grande do Norte (UFRN)

Department of Informatics and Applied Mathematics (DIMAp)

Group of Logic, Language, Information, Theory and Applications (LoLITA)

59078-970 – Natal – RN – Brasil

terrematte@ppgsc.ufrn.br, fabriciocsc@ppgsc.ufrn.br

**Abstract.** *Logic is a subject offered to many undergraduate courses with different focal points. For most of those courses, it is remarkable how Logic represents a pedagogic challenge for both teachers and students, and the recorded number of cases of failures and of discontinuity is often high. Given the need to provide a solid basis in the subject, we propose a set of Learning Object (LO) for teaching Logic Applied to Computer Science: the LOGICAMENTE<sup>1</sup>. The tools that integrate the system aim to illustrate fundamental concepts and algorithms from Logic, as well as to allow students to conduct interactive practical experiments involving the understanding of those concepts.*

**Key-words:** *Learning Objects. Teaching Logic. Logic.*

**Resumo.** *A disciplina de Lógica é oferecida por diversos cursos de graduação com diferentes focos de aplicação. Para a maioria desses cursos, é notável como a disciplina representa um desafio pedagógico para os professores e os alunos de graduação, assim como é frequentemente alto o número de trancamentos e reprovações registrados. Diante da necessidade de uma formação sólida dos alunos na disciplina, propomos um conjunto de Objetos de Aprendizagem (OA) para o ensino de Lógica Aplicada à Computação: o LOGICAMENTE. As ferramentas que integram o sistema visam a ilustrar conceitos e algoritmos fundamentais da Lógica e a permitir que os estudantes façam experimentos práticos envolvendo a compreensão desses conceitos de forma interativa.*

**Palavras-chave:** *Objetos de Aprendizagem. Ensino de Lógica. Lógica.*

## 1. Introduction: Main aspects of teaching and learning Logic

*“Logic: The art of thinking and reasoning in strict accordance with the limitations and incapacities of the human misunderstanding.”*

— AMBROSE BIERCE

*The Collected Works of Ambrose Bierce (1911), Vol. 7, The Devil’s Dictionary, 196.*

The logic has not been sufficient emphasis on computer courses. We note that there are some talents working in universities in that area. But why is it important to study logic and why it has not had their due importance? In [Barland et al. 2000] are mentioned

---

\*The authors acknowledge financial support from CNPq/CAPES. Thanks to all students of Computer Science and Computer Engineering who contributed to the project LOGICAMENTE for a few semesters in the discipline of Logic Applied to Computer on Federal University of Rio Grande do Norte. Indeed, thanks to Prof. João Marcos, for all his attention to making it possible, to create and advise this project.

<sup>1</sup>Is partiality available from <http://www.lolita.dimap.ufrn.br/Logicamente>.

two points that is happening in computer courses. The first is the myth that comes from students, which argue that knowledge of logic will not be useful in their professional careers. The second point is the part of teachers and educational institutions. Often they were not prepared, teaching in a superficial way the content of logic and in other cases the actual course curriculum has trouble putting the logic with a secondary issue within a matter of discrete mathematics, spending a few weeks to teach all content on logic.

Beyond those reasons, the logical concepts are not trivial to understand, then generating a digression on this subject. Aimed to gather some solutions [Barland et al. 2000] brings several points illustrating why we must study logic. The main ones are: Software developers must analyze the behavior of their programs; circuit minimization and equivalence checking Rely on the logical equivalence of designs; Establishing protocol correctness Requires a specification language and verification Corresponding framework. These are some of the many uses of logic.

Seen it, we have pointed out problems in the teaching logic, both attributed to disinterest and lack of proper training of educators. The project LOGICAMENTE has as main goal treat logic as a friendly discipline, bringing an interactive way to teach with content adaptation making it easier to learn logic. In this sense we see the power of the Semantic Web and ontologies with regard to learning objects.

We may have learning material tailored to the needs of each student through an ontology that establishes the connection between these needs and the learning materials. Among many challenges, we focus now on study materials for learning objects, seeking efficiency in search, recovery and adaptation of teaching materials. We aim to further that learning objects are based on ontologies. We believe that virtual environments can benefit from the use of semantic web based in ontologies.

### **1.1. Learning Logics Tools: Challenge, Benefits and Opportunities**

The project LOGICAMENTE is developed aiming to explore the advantages of web applications on teaching of logic, implementing some LOs to perform tasks such as: the automatic generation of formulas with the desired complexity, the configuration of a language and to develop new connectives, the translation between different syntaxes, the construction of truth tables and tableaux of formulas, the interactive presentation of formulas in the form of trees, the implementation of the method of resolution for classical logic, the search for models or counter-models and moreover, the use of an assistant for the demonstration of theorems.<sup>2</sup>

Thus, we expose a set of interactive LOs gathering tools to support teaching of Logic Applied to Computation. Our goal is to produce integrated instruments that illustrate fundamental concepts and algorithms of logic, allowing students to make practical and pedagogical experiments. Finally, we point out some methodological expectations and what we intend to perform further work to make the LOGICAMENTE a Virtual Learning Environment of Logic.

Currently, the conventional curricula in our universities prefer students with an intuitive learning style, verbal, deductive, reflective and sequential, as opposed to those stu-

---

<sup>2</sup>For a better understanding of the concepts of logic implemented in Logic, cf. [Barwise and Etchemendy 2002, van Dalen 2004].

dents that learn by other ways [Barland et al. 2000, Felder and Silverman 1988]. Adding this to the lack of solid basic knowledge of the begginer undergraduate students and the precarious learning resources available, is not surprising that there is a high failure rate and locking of the discipline of Logic Applied Computer in several universities and technology centers.

To solve thoses problems in the teaching of logic, we invested in a more sensory learning style, visual, inductive, active and comprehensive, aimed professional training able to meet more effectively the needs of the industrial sector and academic community.

**Table 1. List of learning tools Logic**

#	NAME	CONTENT	IMPLEMENTATION	WEB
1	<i>Akka</i>	Prop. and 1 <sup>st</sup> Order; Kripke Models	Java Applets	*
2	<i>Apros</i>	Prop. and 1 <sup>st</sup> Order; Natural Deduction	Java Web Start	*
3	<i>ASA-CalcPro</i>	Prop; Tableaux; Truth Table	Java	-
4	<i>Comp. Aristotelian Logic</i>	Aristotelian Logic	PHP	✓
5	<i>Expression Evaluator</i>	Prop. and 1 <sup>st</sup> Order; Expression Evaluator	Perl/cgi	✓
6	<i>JAPE</i>	Prop. and 1 <sup>st</sup> Order; Natural Deduction	Java	-
7	<i>LoTREC</i>	Prop. and 1 <sup>st</sup> Order; Modal	Java Applets	*
8	<i>Pandora</i>	Prop. and 1 <sup>st</sup> Order; Natural Deduction	Java Applets	*
9	<i>Possible World Creation</i>	Prop.; Modal	Flash	✓
10	<i>ProofWeb</i>	Web Assistant-Proof; Natural Deduction	Ocaml	✓
11	<i>ORGANON</i>	Prop. and 1 <sup>st</sup> Order; Semantic; Tableaux and Truth Table	Servlets	✓
12	<i>Tarski's World</i>	Prop. and 1 <sup>st</sup> Order; Semantic and Tableaux	Java	*
13	<i>Fitch</i>	Prop. and 1 <sup>st</sup> Order; Natural Deduction	Java	*
14	<i>Boole</i>	Prop. ; Truth Table	Java	*
15	<i>The Daemon Proof Checker</i>	Prop. and 1 <sup>st</sup> Order; Model Checking; Natural Deduction	C#	-
16	<i>Theorem Proving System</i>	Prop. and 1 <sup>st</sup> Order; Natural Deduction	Lisp	✓
17	<i>Tree Proof Generator</i>	Prop. and 1 <sup>st</sup> Order e Tableaux	JavaScript	✓
18	<i>Truth Table Constructor</i>	Prop.; Truth Table	Java Applets	*
19	<i>Tableau3</i>	Prop. and Tableaux	Java Applets	*
20	<i>DiagVenn1.0</i>	Prop. and 1 <sup>st</sup> Order; Venn Diagrams	Java Applets	*
21	<i>MAFIA</i>	Prop. and 1 <sup>st</sup> Order; Kripke Models	Java Applets	*
22	LOGICAMENTE	Prop. and 1 <sup>st</sup> Order; et al.	PHP	✓

\* Partially available on the Web.

Many support systems to the teaching of logic are available currently, such as those showed in the Table 1. However, most of them are desktop systems, not available on different operating systems, or they are Web applications based on *Java Applets* without store information about the user's session. These systems are restrictive because they need a Java Virtual Machine (JVM) and usually have a low performance execution. Another problem is that these systems are monothematics and cover a very restricted area of the content of Logic (only one or two contents are implemented by each system) and moreover have no challenging activities and does not stimulate the user.

## 2. The Virtual Learning Environment: LOGICAMENTE

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

A Virtual Learning Environment (VLE) is a computer system designed to support teaching and learning in an educational monitoring mechanism. A VLE should provide a set of tools, such as appraisals, interactive communication, transfer of content management, stu-

dent activities, tools, tracking exercises, collaborative entries, Wikis and learning modules specific to the content.

The prototype of LOGICAMENTE was initially implemented in 2006, under the idealization and guidance of Prof. João Marcos, as a final project of the discipline of Logic Applied to Computing. The goal was to implement a methodology of Problem-Based Learning (PBL) for the implementation of some key concepts of logic. In the following semesters, the project continued to be developed from the collaborative work of students and volunteers, with the opportunity to present papers that explored both the computational aspects of algorithms implementations as well aspects of computer technology applied to education [Vilela et al. 2008, Barros et al. 2008, Terrematte et al. 2009]. The prototype of the home screen can be seen in Figure 1.



Figure 1. LOGICAMENTE prototype available currently

## 2.1. Some Leaning Objects of LOGICAMENTE

Currently, we have nine modules, namely: the *Settings*, the *Generator*, the *Formula Reader*, the *Truth-Table Constructor*, the *Tree Interaction*, the *Substitution Master*, the *Prenex Converter*, the *Skolemizer* and the *Resolution Game*, which will be described below.

- *Settings*: is a module that we can choose which style of notation we want to use. Currently, we can choose the notation style *Infix*, *Polish* e *Functional*. The notation *Infix* is the default – represented by  $\alpha \wedge \beta$  –, while the notation *Polish* have connectives preceding the formulas in scope –  $K\alpha\beta$  – and, the latter case is the notation *Function* which represents connectives as functions –  $f(\alpha, \beta)$ . Furthermore, its possible to add new connectives by setting his symbol, arity and order of association. The goal is to create new connectives and save them in the database for define new logics. In the picture 2(a), we can find the prototype of this module.

- *WFF<sup>3</sup> Generator*: this module is responsible for generate arbitrarily well-formed propositional formulas, according to certain properties entered by the user (number of occurrences of connectives, sentential number of variables and others). With this module, sets of formulas are generate according to the desired complexity and might prepare some exercises automatically with generated formulas. Also the student can generate formulas,

<sup>3</sup> A WFF is a **Well Formed Formula**, that mean a sequence of generated according to a formal language.

put them on the clipboard and analyse them in other modules. In the picture 2(c), we observe the generation of three formulas with two connectors and three atoms.

- *Formula Reader*: is a module for inserting formulas by the user to the clipboard, to be analysed in others modules.

- *Truth-Table Constructor*: enables the construction of truth-table and the satisfiability of a collection of formulas, display of all possible states of sentential variables and the complete formula as a whole. Must support, in a second stage, the setting to define new operators for the user, as well for evaluation of formal semantic for certain non-classical logic that can be characterized by interpretations in terms of finite truth tables. In the Figure 2(b), we display a truth table for the formula  $((p \wedge \neg p) \rightarrow q)$ .

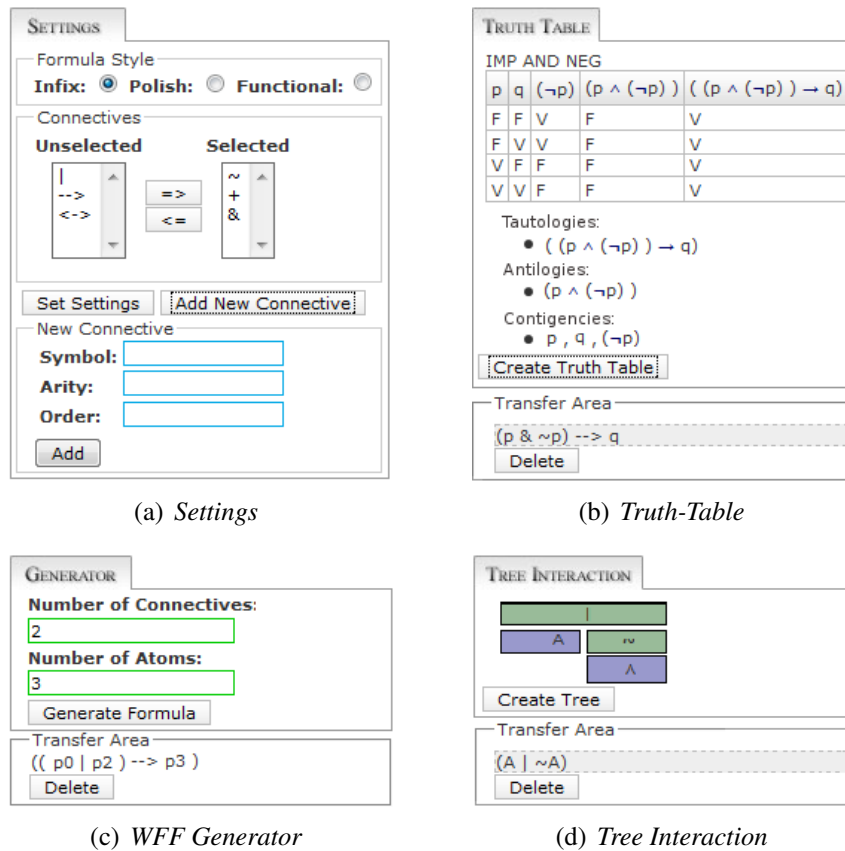


Figure 2. Some Learning Objects prototypes of LOGICAMENTE

- *Tree Interaction*: the formulas are data structures that corresponds to trees. This module makes these trees viewable, and allows the user to interact in various ways with their nodes and leaves of such trees. Many other uses are given in this module interact, view and interact with tableaux, resolution method and natural deduction. In the Figure 2(d), the module shows a generated tree representing the formula  $(A \vee \neg A)$ .

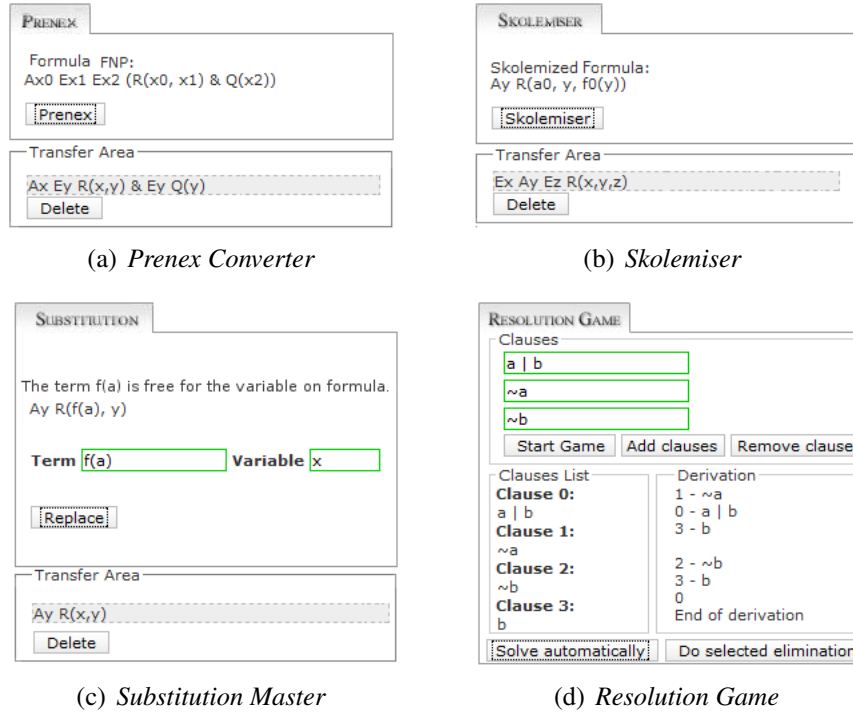
- *Substitution Master*: implements a function that takes a formula in first order, a variable to be replaced and a term, and first of all, by checking if the term is free for the variable in the formula received <sup>4</sup>. If so, the replacement is made. Otherwise, the user is consulted if

<sup>4</sup>A term  $t$  is *free* for a variable  $x$  in a wff  $\alpha$ , if there is no occurrences of  $x$  in  $\alpha$  in an scope of quantification, of  $\forall y$  or  $\exists y$ , such that  $y$  is a variable in  $t$ .

he desire to properly rename the bound variables of the formula to allow the replacement is performed. In the Figure 3(c), we display the replacing of the variable  $x$  by the term  $f(a)$  in the formula  $\forall y R(x, y)$ , resulting in  $\forall y R(f(a), y)$ .

- *WFF Converter*: this module performs some conversions of formula necessary for the method of resolution in first order logic, through the manipulation of quantifiers and connectives in the appropriate reductions. The module accompanies, step by step, the conversion of a formula for the Conjunctive Normal Form (CNF)<sup>5</sup> and interactively transforms formulas for Prenex Normal Form (PNF)<sup>6</sup>. The module also receives first order formulas in PNF and enables application of the method of Skolemization<sup>7</sup>.

- *Resolution Game*: implements the basic resolution method, assisting the student in eliminating complementary literals<sup>8</sup>. The Figure 3(d) shows a instance of the resolution method for  $\{\alpha \vee \beta, \neg\alpha, \neg\beta\} \mapsto \square$ , concluding that  $\{\alpha \vee \beta, \neg\alpha, \neg\beta\}$  is a set of contradictory formulas.



**Figure 3. Others Learning Objects prototypes of LOGICAMENTE**

<sup>5</sup>A formula is in Conjunctive Normal Form, if is a conjunction of subformulas such that each of the subformulas is a disjunction of literals and each literal is an atomic proposition, or the negation of an atomic proposition, i.e. has the form  $p$  or  $\neg p$ .

<sup>6</sup>A formula is in prenex normal form, if has the following form:  $Q_1 x_1 \dots Q_n x_n \alpha$  such that  $\alpha$  is a formula free quantified,  $x_1, \dots, x_n$  are different variables and each  $1 \leq i \leq n$ ,  $Q_i$  is  $\forall$  or  $\exists$ .

<sup>7</sup>The name of the method is in honor of the mathematician and philosopher Thoralf Skolem (1887-1963). Skolemization means the elimination of existential quantifiers in closed formulas in PNF, where the variables are replaced by functions of choice and new symbols are introduced constants and functions.

<sup>8</sup>The resolution method is used to decide whether a sentence has a formula in CNF that is a theorem. If  $\alpha$  and  $\beta$  are basic clauses, with  $p_k \in \alpha$  and  $\neg p_k \in \beta$ , then the result of the rule Complementary literal elimination of  $\alpha$  and  $\beta$  - concerning to  $p_k$  and  $\neg p_k$  - is  $(\alpha - \{p_k\}) \cup (\beta - \{\neg p_k\})$ , such that  $\gamma - \delta$  means the set of the difference between  $\gamma$  and  $\delta$ , i.e.  $\gamma - \delta \equiv \{x | x \in \gamma \text{ e } x \notin \delta\}$ .

Our immediate goal is to add the following LOs, already implemented but not yet integrated to the LOGICAMENTE:

- *Small Counter-Model Builder*: implements a finite search engine of counter-models, i.e. a bounded model checker, receiving as input a number limited of objects, a collection of assumptions and a goal formula of conclusion.
- *WFF Diagnoser*: tests whether a expression is a well-formed formula and, in a negative case, it suggests some fixes. The module verifies if there is some redundant parentheses in the formula, besides it exposes various informations of well-formed formulas: its complexity, its set of basic atoms of subformulas among others properties.
- *WFF Translator*: the module create syntax for formulas and might to represent the connectives in different ratings symbols, images, HTML codes, codes in ASCII or LaTeX. Also with this module, its possible to translate formulas of different applications.
- *Tableaux Constructor*: implements the method of tableaux of first order logic, allowing the student to see the refutation tree and choose which rule to use.
- *Semantic Consequence Tester*: here the student can insert sets of propositional formulas and test step by step, the relation of semantic consequence involving the same. The module generates a counter-model whenever possible.
- *Theorem Proving Web System*: implements a demonstration assistant in Natural Deduction. We intend to develop this module integrating to our project the ProofWeb<sup>9</sup>.

## 2.2. Modeling presentation

The LOGICAMENTE is implemented as a Web application in PHP language and AJAX technology, aiming that the system be available at anywhere and without require a previously installation. In Figure 4(c), we observe the model diagram currently implemented structures of propositional sentences, first-order propositions and their relationships between terms and functions.

All LOs presented above should cover both the classical first order logic as propositional logic. The set of LOs specified above must meet the following requirements:

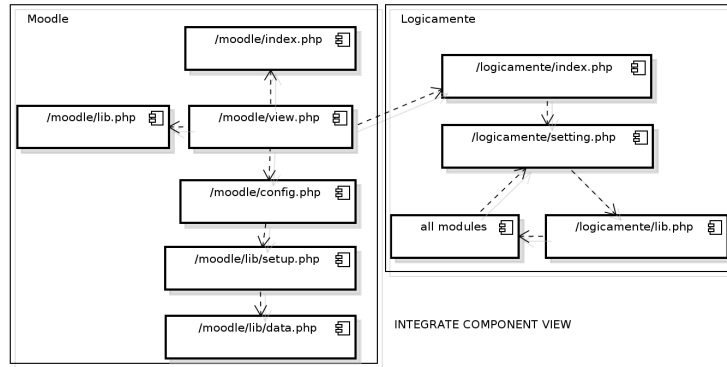
- User Interface: For each LO, a user-friendly interface is implemented to provide easier and enjoyable to use. Besides that, the system has a tutor for the novice user can use the program and answer a few questions about the system.
- Correction and optimization: we carefully analyse the implementation of all Learning Objects to maintain the correction of the implemented algorithms.

In the Figure 4(b), is shown the activity diagram of loading process of the Moodle, until the VLE is ready to be used. The activities take place in an order that begins with the connection to the database. The second step is to load the classes, note that the LOGICAMENTE is a module of Moodle and can be added to any course. In this point we assume that we have already loaded a course and registred his activities. After the Moodle and LOGICAMENTE are loaded, we can access the course in accordance with the kind permission of the user account. See Figure the activity diagram.

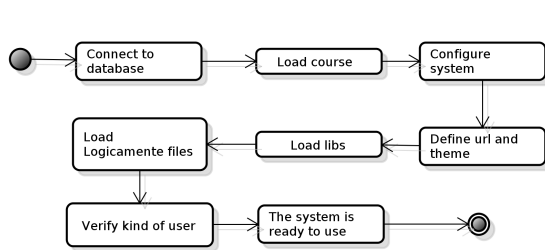
---

<sup>9</sup>The *ProofWeb* is an opensource software for teaching Natural Deduction which provide a interaction between some demonstrators theorems (Coq, Isabelle, Lego) and an interface Web. It is available in <http://proofweb.cs.ru.nl/>.

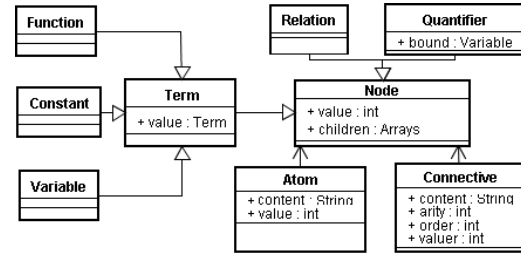
The diagram of Figure 4(a) presents the component view with the integration of Moodle and LOGICAMENTE. On the basis of Moodle we have the system with data access components, libraries, configurations and component view that calls the VLE. In the system we have an important component that is the setting that is requisite for this to work. Once set up the LOGICAMENTE, we will have access to all modules that access to the their libraries.



(a) Component Diagram



(b) Activity Diagram



(c) Model Diagram implementing formulas

Figure 4. Architecture Overview of LOGICAMENTE

### 3. Methodological Aspects: evaluation and monitoring of learning Logic

*It is by logic that we prove,  
but by intuition that we discover.*  
— JULES HENRI POINCARÉ

*In Science and Method (1908) translated by Francis Maitland (1914, 2007), 129.*

Following the line argued by Antonia Huertas [Huertas 2007] to the teaching and learning logic – identifying two paradigms of education, face-to-face and e-learning –, we agree that should be adopted teaching techniques for the course that use elements of both. For example, it is still difficult to have an evaluation method based only on the e-learning paradigm, then we must do activities in e-learning modality and any test in the old model with paper and pen.

However, the success of a model based on e-learning depends not only on techniques applied in most VLEs, after the adoption of this model is a constant need to improve the motivations to the student and the uses of process. The mains methodologies identified by [Huertas 2007] are: Instrumental, Interdisciplinary, Constructivism, Conceptual modeling, Personalization of the learning process, Workload adjustment and Continuous evaluation process. Our aim is to further examine these features applied into the project LOGICAMENTE.



In the LOGICAMENTE, the teacher have the control of students, classes, lessons and exercises. Besides, even the teacher have access to activity reports of students performance, also there is an agent for monitoring the learning modules and students activities of interactions with the OAS. In turn, the student is assessed by his performing with assigned lessons and have exercises related to LOs.

The LOs developed and presented aim to clarify the relationships between different approaches of logic, such as the semantic theory, the proof theory and others approaches of Logic applied. These different approaches will have their content associated with activities and challenges offered in Moodle, aiming a constructive process of teaching and learning Logic. The LOGICAMENTE, seeks to explore the potential offered by Web methodological, and also enable the implementation of different teaching styles of logic, without being restricted to particular styles of learning.

#### 4. Future Works and Final Remarks

*“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).*

Aiming to facilitate the reuse of LOs, the LOGICAMENTE use the standard IMS Learning Design (IMS-LD) to have their specified content and to ensure the sustainability of its resources [IMS Global Learning Consortium 2003]. IMS LD is compatible with Moodle and is a framework used for the planning of new courses, support activities, organization of learning environments [de Souza Dutra and Tarouco 2006]. On Figure 5 we present an overview of our system through the tool of editing and reuse of LOs, the ReCourse<sup>10</sup>.

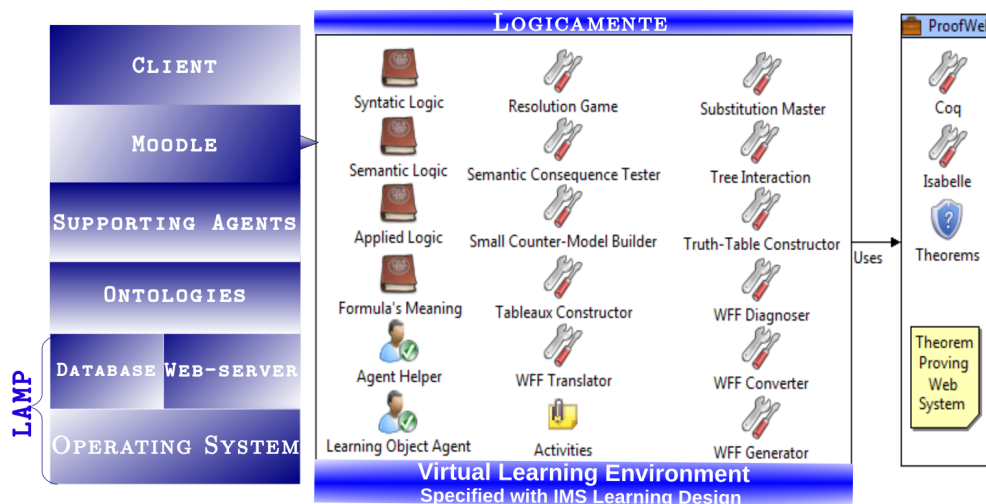


Figure 5. High level presentation of LOGICAMENTE

A second point identified in [Bittencourt et al. 2008] was that the teaching supported by computer can enjoy the benefits of semantic web. In this sense the LOGICAMENTE need to use Semantic Web techniques, to effectively manage organize and manage e-learning resources available according to peculiar necessities of both teachers and students, has been advocated by many authors. He presents the components of the Semantic Web-based

<sup>10</sup>Availiable from <http://www.tencompetence.org/ldauthor/>.

Educational Systems (SWBES) as described in [Bittencourt et al. 2008] are related to the roles mainly of users (teachers, learners, authors, groups, developers) to the educational resources, to the environment, to its interface and the functionalities it does offer. The SWBES should include: ontologies, tutoring/pedagogical agents, semantically Described tools and services (better if they are web 2.0 based). Accordingly identified as a future goal to create ontologies for intelligent agents that can organize and manage learning resources in accordance with the special needs of each user.

Currently, the area of teaching Logic has a lack for a computational tool to explore other styles of learning and teaching. Although there are available tools for teaching logic, these applications do not represent virtual learning environments and really integrated interactively. In order to fill this need, we present the project LOGICAMENTE. In the present work, we have explained how the project was initially conceived from the work collaborative and learning based on implementation problems, finally it was presented its learning objects implemented currently.

## References

- Barland, I., Felleisen, M., Fisler, K., Vardi, M. Y., and Kolaitis, P. (2000). *Integrating Logic into the Computer Science Curriculum*. Helsinki.
- Barros, T. M., Araújo, A. E. F. D., and Marcos, J. (2008). A implementação colaborativa de uma suíte de ferramentas on-line de apoio ao ensino de Lógica. *VIII ERMAC-R3*.
- Barwise, J. and Etchemendy, J. (2002). *Language, Proof and Logic*. Center for the Study of Language and Information.
- Bittencourt, I. I., Isotani, S., Costa, E., and Mizoguchi, R. (2008). Research directions on Semantic Web and education. *Scientia Interdisciplinary Studies in Computer Science*.
- de Souza Dutra, R. L. and Tarouco, L. M. R. (2006). Objetos de Aprendizagem: Uma comparação entre SCORM e IMS Learning Design. *Novas Tecnologias na Educação – CINTED-UFRGS*, 4(1):1–8.
- Felder, R. M. and Silverman, L. K. (1988). Learning and Teaching Styles In Engineering Education. *Engr. Education*, 78(7):674–681. <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf>.
- Huertas, A. (2007). Teaching and learning logic in a virtual learning environment. *Logic Journal of IGPL*, 15(4):321.
- IMS Global Learning Consortium, I. (2003). *IMS Learning Design Best Practice and Implementation Guide*, version 1.0 final specification edition. <http://www.imsglobal.org/learningdesign>.
- Terrematte, P., Marcos, J., and Galdino, T. (2009). O LOGICAMENTE: A implementação colaborativa de Objetos de Aprendizagem de Lógica. *IV Workshop de Arquiteturas Pedagógicas para Suporte à Educação a Distância mediada pela Internet (IV WAPSEDI)*. In *SBIE - XX Simpósio Brasileiro de Informática na Educação*.
- van Dalen, D. (2004). *Logic and Structure*. Springer.
- Vilela, G., Rosan, M., and Marcos, J. (2008). Implementação de um gerador e verificador de modelos finitos para a Lógica Clássica de Primeira Ordem. *VIII ERMAC-R3*.