SUPPORT STUDENTS' EXPERIMENTAL WORK IN ELECTRICAL ENGINEERING WITH VISUAL MODELING

ANCA DANIELA IONIȚĂ[1], ADRIANA OLTEANU[1]

Key words: Modeling languages, Web-based applications, Computer-Aided Education, measuring instruments.

Students' experimental work in electrical engineering laboratories is facilitated by two important features: visual representations of the systems existing in the laboratory, and their interpretation for various purposes, like simulation or acquisition. Environments like Simulink and LabVIEW are already consecrated in the academic world. However, there are a lot of laboratories that work with equipment having its specific modules for acquisition, interoperation, or processing. This paper shows how visual modeling is supported in a Web-based application that aids education in electrical engineering, by characterizing various measuring instruments and by increasing students' capacity of understanding. Taking into account the current inclination of the new generations towards visual representations, to the detriment of text, the use of models in various forms is expected to be an adaptation to the preferences and the way of thinking of the upcoming students, who grew up along with the Internet technology.

1. INTRODUCTION

In the traditional model of education, the quality depends on the teacher's level of knowledge and ability to share it. Computer-Aided Education introduces access to multiple sources of knowledge – collected, assembled, and sequenced by the teachers, or accessible through Internet. This creates a lot of additional possibilities of combining traditional methods with e-learning resources and activities, which may be assessed, used or recommended by teachers. Thus, the quality of education also depends on the quality of electronic knowledge sources and didactic materials [1].

Academic laboratories in electrical engineering work with a large variety of measuring instruments, characterized by specific physical quantities, units of measurement, settings, warnings etc. Generally, students have to understand practical work guidelines that explain new instruments and new experiments, and

[1] Automatic Control and Computers Faculty, University "Politehnica" of Bucharest, Splaiul Independentei 313, 060042, Bucharest, Romania, Tel +(4021) 4029113; E-mail: Anca.Ionita @ aii.pub.ro, Adriana.olteanu @ aii.pub.ro

Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **59**, *1,* p. 107–116, Bucarest, 2014

they have to assimilate the new information fast. This educational area may benefit from specific computer-aided environments, with requirements like:

- introducing simplified representations of instrument concepts, for increasing the performance of learning;
- generating lessons based on instrument characterizations, pre-existing in a model library;
- interpreting models for parsing data acquired from various instruments and integrating them in a central database;
- querying data for further processing with Computer-Aided Design environments;
- generating reports of the students' practical work.

For describing the measuring instruments used in electrical engineering laboratories, we used visual modeling languages, designed for a friendly communication, complementing and extending textual or spoken presentations. The power of modeling languages stands in summarizing the essential elements of a system, reducing its complexity, but also in representing it in a formalism that may be subsequently used for automated interpretation. Students in engineering are generally well accustomed to the symbolic style of learning [2], so introducing modeling elements into the learning content is expected to give abstract and easily comprehensible learning objects.

The EquiLAB tool, presented in this paper, is an extension of the Equipment Model Editor previously developed for integrating data in an electromagnetism laboratory [3]. The idea was to offer diagrammatic representations for the data models of various instruments, as they are generally recognized to be the most suggestive, from the point of view of methodologists, tool vendors and educators.

2. MODELING LANGUAGES

Models have been intensely used in all fields of engineering, in various forms: mathematical formulae, graphical representations with symbolic elements, structured text, informal drawings. From the large variety of modeling approaches, we chose two trends for our work:

- *object orientation* – grouping instances by classes, characterized by attributes and operations that modify them; this paradigm has dominated conceptual modeling since the 90s, starting with methodologies like the Booch method, OMT (Object Management Technique) and OOSE (Object-Oriented Software Engineering) and conducting to the adoption of the UML (Unified Modeling Language) standard [4];
- *representation through conceptual graphs* – considering concept nodes and relationships between them [5]; this formalism pertains to visual modeling, used for displaying the concepts graphically; software engineers

intensely use modeling, especially since the adoption of numerous MDA (Model Driven Architecture) standards, including UML.

Models are typically used for describing a physical or conceptual system, introducing a simplification, usually called abstraction, built with "an intended goal in mind" [6]; therefore, modeling is a relative, not an absolute approach. A system may have multiple models, sometimes correspondent to different points of views; in a survey on visual Model-Driven Development, Giese et al. analyzed approaches like MATLAB/Simulink, CHARON, HybridUML, SysML and they concluded that they generally support structure and behavior views, but they lack formal verification, code generation and support for integrating PSM (Platform Specific Model) elements, like legacy and reusable assets [7].

The completeness of a model depends on the language it is defined in, i.e. on its metamodel. The metamodel indicates the relevant concepts that allow the representation of a system through an abstract model [6], plus the relationships between them. It is therefore strongly linked to conceptual modeling, where each concept corresponds to an element of the real world, and also to domain engineering [8].

From the point of view of the scope covered by a modeling language, we also consider two important approaches:

- *general-Purpose Languages* (GPL) – aiming to fit for describing and designing any application domain;
- *domain Specific Languages* (DSL) – representing various aspects of the systems with the concepts familiar to the domain experts.

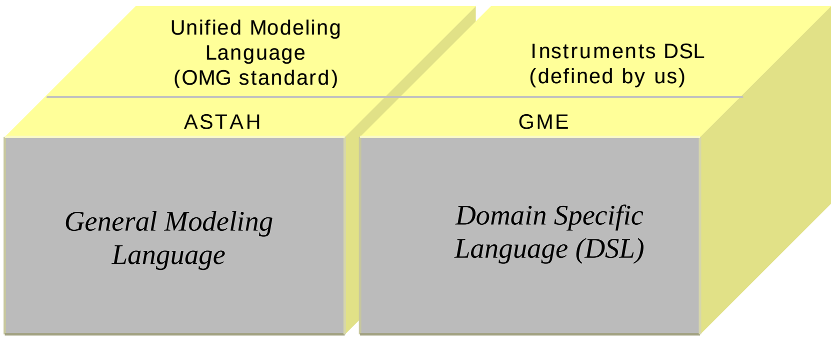


Fig. 1 – Modeling languages used in our approach.

Our purpose was to support both approaches, as represented in Fig. 1. For the former, the choice was Unified Modeling Language – the standard adopted by Object Management Group (OMG) for object-oriented analysis and design; we used Astah Community Edition [9] as modeling editor. For the latter, we developed our own Domain Specific Language (DSL), called *Instruments*, dedicated to the measuring instrumentation domain; we used Generic Modeling Environment

(GME) [10], a configurable toolkit for creating DSLs; we defined concepts and relationships specific to the measuring instruments domain, and we generated a dedicated model editor, with customized notations.

3. WEB-BASED EDUCATIONAL SUPPORT

The EquiLAB environment is based on creating a collection of equipment characterizations (Fig. 2), which are further used for generating Web pages representing various lessons and practical work guidelines. These lessons contain a common structure, where the information regarding the instrument is automatically filled in, and other specific elements, like particular tasks, or theoretical background, are separately edited. The equipment is associated to a set of actions, presented in Table 1.

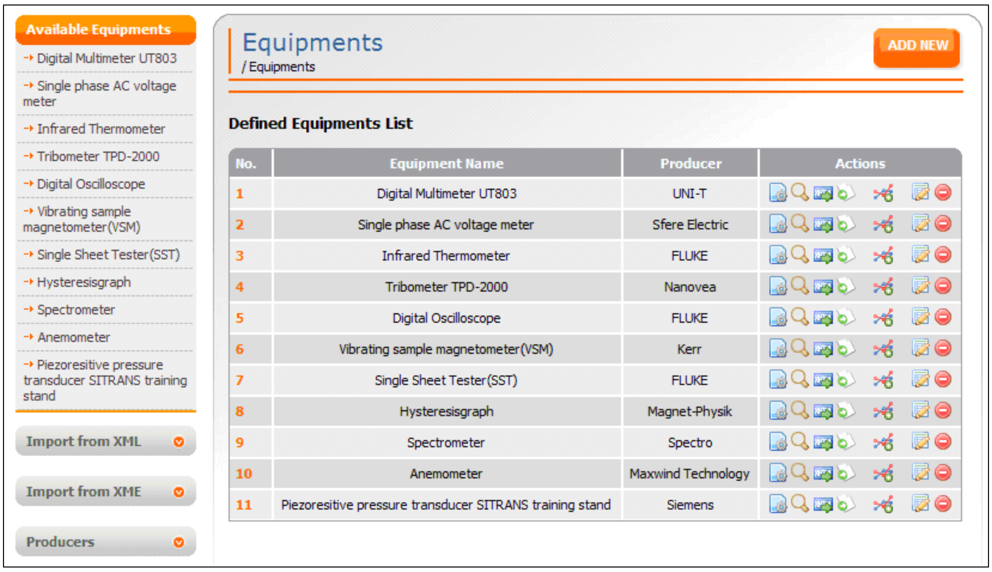


Fig. 2 – List of instruments supported by EquiLAB
and used in electrical engineering laboratories.

The idea that differentiates this system from other learning content managers is that the equipment / instrument characterizations are stored in a central database, according to well-defined criteria, so they can be transformed into semiformal visual models, conforming to UML and also to the *Instruments* language, developed in GME. The environment includes automatic transformations between these models [11]. For education in electrical engineering, we defined a library of models (as described in the next chapter), which can be further extended.

Table 1

Actions applied to measuring instruments

Icon	**Action**	**Description**
	Manage parameters	Manage the instrument parameters, distributed into the following groups: Measurement information, Measured object characterization, Acquired data, Measured functions, Experiment settings, Initial instrument setup, and Warnings.
	View equipment	Visualize all the information regarding the instruments, including their parameters, the correspondent measuring units, and the graphical models in UML and GME.
	Export to xml	Export the instrument characterization, consisting in its groups of parameters, to a file written in XML (Extensible Markup Language) in order to support a standard encoding format that is both understandable to humans and interpretable by other programs.
	Export to xme	Export the instrument characterization, consisting in its groups of parameters, to a file with .xme extension, containing an XML representation of GME projects used for interoperation
	Import measurement	Imports a particular measurement into the central database, by parsing the data and transforming their specific format for corresponding to the data model of EquiLAB.
	Edit	Supports the modification of instrument information, i.e. of its parameters.
	Delete	Deletes an instrument characterization from the central database.

4. MODEL LIBRARY

The equipment models created with EquiLAB are stored in several formats, each one useful for a particular purpose:

- in the database, according to its data model; this format is used for generating part of the lessons, but it can also be transformed into an interoperable XML (Extensible Markup Language) format, and into .xme files that can be imported in GME;
- in Unified Modeling Language, the standard object-oriented modeling language, because it is an intuitive way for representing a large variety of systems; the UML notations used in our models are described in Appendix 1.
- in the domain specific language called *Instruments* defined in the Generic Modeling Environment, in order to work with domain concepts exclusively; the GME *Instruments* notations are described in Appendix 2.

Several examples of visual models, represented with UML and GME-*Instruments* languages, are illustrated below. A digital multimeter, characterized by 20 parameters, has its general and domain-specific models illustrated in Fig. 3 and Fig. 5 respectively. A much simpler instrument, a single phase AC voltage meter, characterized by only 6 parameters, is also modeled in Figs. 4 and 6.

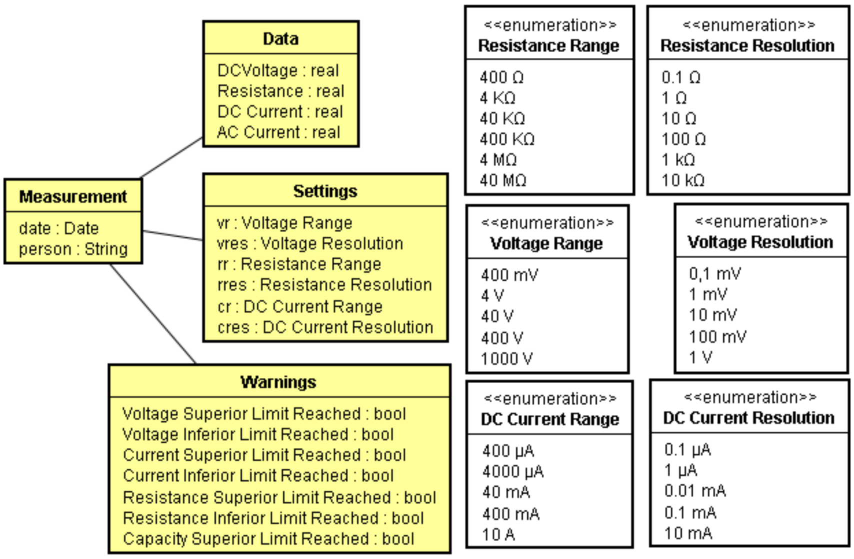


Fig. 3 – Model for a Digital Multimeter represented with the standard *UML* language.

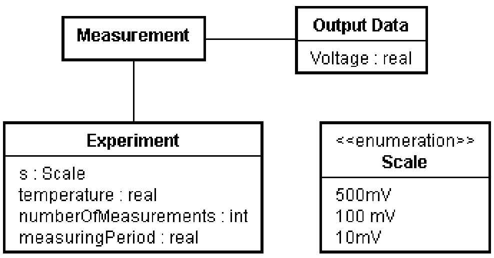


Fig. 4 – Model for a Single Phase AC Voltage Meter represented with the standard *UML* language.

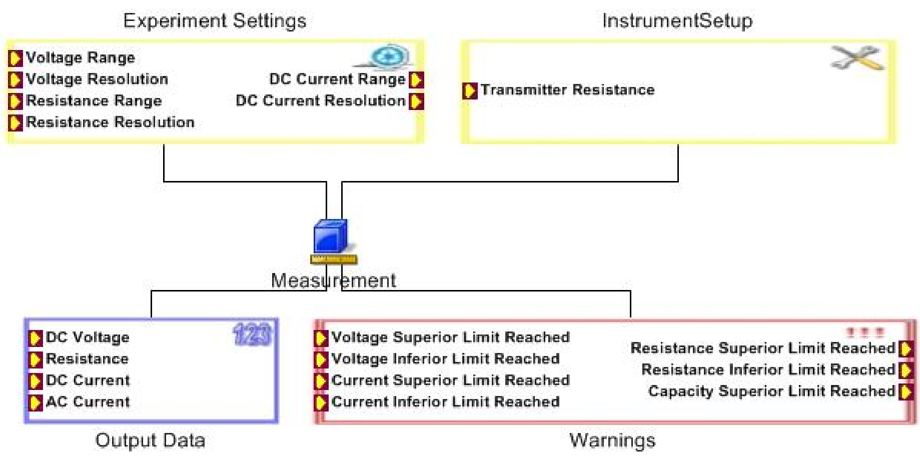


Fig. 5 – Domain Specific Model for a Digital Multimeter, realized with GME *Instruments* Language.

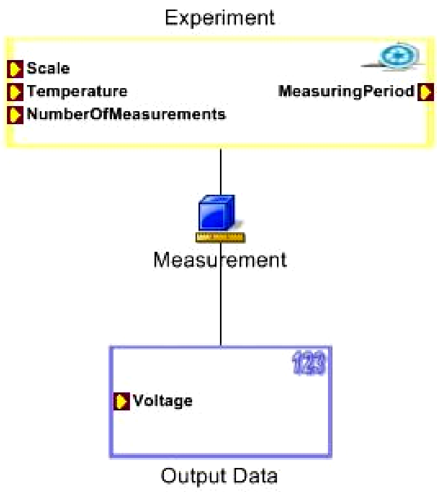


Fig. 6 – Domain Specific Model for a Single Phase AC Voltage Meter, realized with GME *Instruments* Language.

For UML, the modeler uses general concepts, like classes, and has to assign the proper class names according to the domain knowledge. The GME *Instruments* language already incorporates these concepts. Moreover, GME has a larger flexibility of hiding details, and of showing them when double clicking various elements of the model. Yet, a teacher is electrical engineering does not need to use modeling editors, which are not specific to his or her expertise. The visual models may be generated from a Web-based editor where parameters may be introduced by filling in several forms. EquiLAB generates then visual representations as those from Figs. 5 and 6.

6. RELATED WORK

EquiLAB belongs to the class of Learning Content Management Systems (LCMS) – used to create information objects assembled in learning objects with specific objectives; these objects may be sequenced or grouped for structuring larger components, like defining a course made up of multiple lessons. They are generally tagged with metadata, for allowing reuse; an example of standard for the interoperability of learning objects is Learning Objects Metadata (LOM), approved by IEEE-SA and also by ISO/IEC JTC1/SC36 [12]. LCMSs implement workflows for defining learning content, annotating it and saving it into a repository.

Besides, the Web pages generated by EquiLAB can be accessed from any kind of Learning Management System (LMS) that organizes the learning content, dividing courses into modules and lessons. Some well-known open source LMSs are Moodle, Docebo, Sakai. Other platforms for rapid development, integration, update and deployment of learning content are Elab and ELeap.

Related to the use of graphical representations of essential domain elements, an approach that is very close to ours is the use of concept maps for representing knowledge, and thus enhancing the students' capability to learn [13]. Fatemeh et al propose an interactive tool that assists students and teachers in generating concept maps from a given course material, by proposing lists of concepts and linking phrases [14].

Our approach may also be related to MEMOPS framework, built for managing data irrespective of scientific field. It uses UML for data models and generates code for multiple programming languages and storage formats [15]. In [16] one also discusses the use of IEC Common Information Model, containing data types for characterizing power systems equipment.

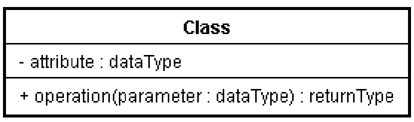
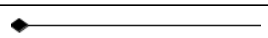
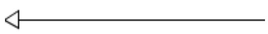
7. **CONCLUSIONS**

The paper presented a Web-based environment developed for helping teachers in Electrical Engineering to reuse instrument characterizations for generating lessons and supporting students' experimental work with new ways of presenting structured information. As the approach is interdisciplinary and it is mainly dedicated to engineers that are not specialists in modeling languages, we selected intuitive representations, such as to be comprehensible without any previous knowledge related to the syntax or semantics; even if models are expressed in semi-formal languages, their purpose is to outline the application domain concepts exclusively. The expertise of teachers in electrical engineering is expected to improve and extend the model library, and to define more experimental lessons based on visual representations of laboratory equipment.

APPENDIX 1. UML NOTATIONS

Table 2

Description of UML Notations

Notation	**Concepts**	**Description**
	Class	A set of objects with the same attributes and operations.
	Attribute	Represents the class structure.
	Operation	Represents the class behavior.
	Association	A relationship between classes.
	Role	A name of an association end.
	Multiplicity	The number of objects that may be linked: 0..1 (0 or 1), * (0 or more)
	Composition	A whole-part relationship between classes
	Generalization	A relationship between classes, where the class pointed by the triangle is the more general one

APPENDIX 2. GME INSTRUMENTS NOTATIONS

Table 3

Description of GME *Instruments* Notations

Notation	Concept	Description
	Measurement	A group of parameters regarding a particular measurement.
	Measured object	A group of parameters that characterize the measured object, sample or assembly
	Data	A group of parameters that signify the physical quantities for which data are acquired
	Experiment	A group of parameters that are set for performing a certain measuring experiment
	Setup	A group of parameters used for the initial instrument setup
	Warning	A group of warnings

Received on January 29, 2013

REFERENCES

1. E.B. Cohen, M. Nycz, *Learning Objects and E-Learning: an Informing Science Perspective*, Interdisciplinary Journal of Knowledge and Learning Objects, **2,** 2006.
2. M.N. Meeker, *The Structure of Intellect: Its uses and interpretation*, Columbus, OH: Charles Merrill, 1969.
3. V. Ionita, A.D. Ionita, *Architecture for Integrating Data Obtained by Advanced Characterization of Magnetic Materials* , Romanian Journal of Materials, ***38***, *1*, pp. 69–75, 2008.
4. *** *Unified Modeling Language (OMG UML), Superstructure*, Version 2.3., 2010.
5. J. Sowa, *Conceptual Structure: Information Processing in Mind and Machine*, Addison-Wesley, 1984.
6. J. Bézivin & O. Gerbé, *Towards a Precise Definition of the OMG/MDA Framework*, IEEE International Conference on Automated Software Engineering (ASE), USA, 2001.
7. H. Giese, St. Henkler, *A survey of approaches for the visual model-driven development of next generation software-intensive systems*, Journal of Visual Languages and Computing, **17**, pp. 528–550, 2006.
8. M. Simos, *Organization domain modeling and OO analysis and design: Distinctions, integration, new directions*, STJA’97, Technische Universität Ilemenau, Thüringen, Germany, 1997.
9. Astah Community, http://astah.net/editions/community
10. *** *A Generic Modeling Environment, GME 5 User’s Manual*, Version 5.0, Vanderbilt University,Institute for Software Integrated Systems, 2000–2005.

11. A.D. Ionita, A. Olteanu, T. Ionescu, L. Dobrica, *Automatic Transformations for Integrating Instrument Models across Technological Spaces*, Romanian Journal of Information Science and Technology, **14,** *1*, pp. 51–66, 2011.
12. *** *Information technology – Individualized adaptability and accessibility in e-learning, education and training – Part 1: Framework and reference model*, ISO/IEC *24751-1,* 2008
13. A.J. Cañas, R. Carff, G. Hill, M. Carvalho, M. Arguedas, Th. C. Eskridge, J. Lott, R. Carvajal, *Concept Maps: Integrating Knowledge and Information Visualizati*on, In: S.-O. Tergan, T. Keller (Edit.), *Knowledge and Information Visualization, Searching for Synergies*, Lecture Notes in Computer Science, **3426**, pp. 205–219, 2005.
14. H. Fatemeh, K. Ahmad, D.M. Mohammad, *ICMAP: An interactive tool for concept map generation to facilitate learning process*, Procedia Computer Science, ***3***, pp. 524–529 (2011).
15. R.H. Fogh et al., *A framework for scientific data modeling and automated software development,* Bioinformatics, **21***, 8*, pp. 1678–1684, 2004.
16. S. Vukmirovic, A. Erdeljan, I. Lendak, D. Capko, *Unifying the Common Information Model (CIM)*, Rev. Roum. Sci. Techn. – Electrotechn. et Energetique, **57**, *3*, pp. 301–310, 2012.