

Intelligent WBT: Specification and Architecture of the Distributed, Multimedia e-Learning System e-stat

Claus Möbus, Bernd Albers, Stefan Hartmann, Jochen Zurborg

Innovative Learning and Teaching Systems,
Department of Computing Systems, University of Oldenburg

Keywords. Architectures, Authoring Systems, Cognitive Approaches, Distributed Learning Environments, Intelligent Distance Learning, Internet Environments, Learning Environments, Content-Engineering, Intelligent Web Based Training (I-WBT), Specification of e-Learning Systems, Unified Modeling Language (UML), Extensible Markup Language (XML),

1. Introduction

Modern e-Learning Systems (eLS) are expected to be innovative not only concerning comprehensive representation of content enriched by multimedia, but also in the integration of learning situations in contexts suitable for students. Suitable, motivating contexts can be “fun” as found in strategic games or business simulations or of a more “serious” variety in the form of virtual data labs. In the new BMBF Project EMILeAstat (e-stat) 13 partners from different organisations are cooperating to construct such an innovative intelligent web based training (I-WBT) system for applied statistics. The German Federal Ministry of Education and Research finances e-stat by means of the NMB funding program “Neue Medien in der Bildung” (New Media in Education).

Special emphasis is placed on reuse and sharing of contents, clean separation of factual contents and its didactical motivated presentation, as well as the avoidance of proprietary solutions. E-stat is an attempt to go beyond the scope of existing WBT systems by using a strong integrating concept in combining content with a high diversity of methodical and didactical approaches. This ambitious approach creates the need for new research and evaluation. For example, a method for the presentation of coherent and user-adaptive content (learning objects) supplied by a variety of sources has to be found. The development of an eLS can be approached from different angles. This paper describes the architecture of e-stat from a knowledge and content engineering point of view, while applying cognitive-science criteria where necessary. All components were developed in a close cooperation with the statistical content providers.

2. The I-WBT-System “e-stat“

Since e-stat integrates different learning-methods, scenarios, and a consulting component into a knowledge landscape, existing methods of specification¹²³ have to be modified accordingly, to ensure a systematic method for production of content. For development and analysis purposes we make use of concepts and

notations supplied by object orientated analysis (OOA) and object orientated design (OOD)⁴. The specification will be realized on demand during the course of the project.

2.1 Specification of e-stat

In the beginning the wind rose⁵ (Fig.1) was used as a metaphor for the e-stat idea. It was meant to express e-stats ambition to supply applicable solutions with changing didactical demands⁶⁷ (e.g. instructional, cognitive and constructive): courses of differing levels of complexity for mathematicians, managers, psychologists and engineers but also for people with a special need of practical experience such as industrial technicians.

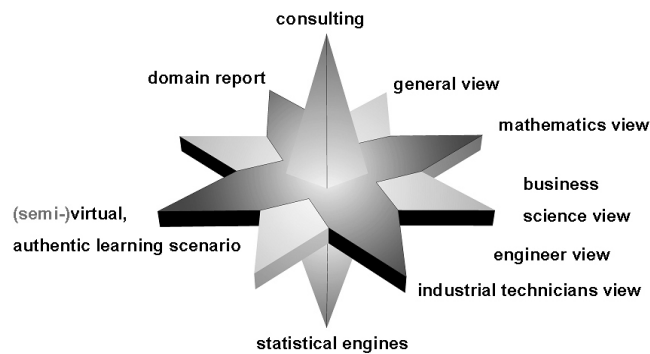


Figure 1: The Wind Rose

e-stat furthermore contains methods to integrate existing statistical engines, (semi-) virtual learning scenarios, an automated glossary, and the case based consulting component for the “hasty user”. The wind rose metaphor is a visualisation of the e-stat project suitable for people without expertise in computer science.

Next the wind rose was transferred into use-cases of the semiformal UML-Notation⁸ (Fig. 2). A use-case is the typical application of e-stat in a user-specific context. Due to the open nature of the e-stat system, the process of defining new use-cases has not been finalized.

The next step in the OOA constitutes the construction of the static system structure using a class diagram.

A class diagram represents classes with their attributes and operations, inheritance and associations between them. A class defines structure (attribute), behaviour (operations), and relations (associations and inheritance structures) for a collection of certain objects⁹. As can be seen in our class diagram, e-stat is a composite aggregate of views (Fig. 3). Views are shared aggregates of scenarios, courses, course units and concepts. Concepts have recursive structure. They can be built up by text blocks (text leafs) module frames, and/or concepts. This architecture ensures the representation of hierarchically organized lessons. Module frames are again composite aggregates of modules, which are the smallest building blocks of the e-stat landscape.

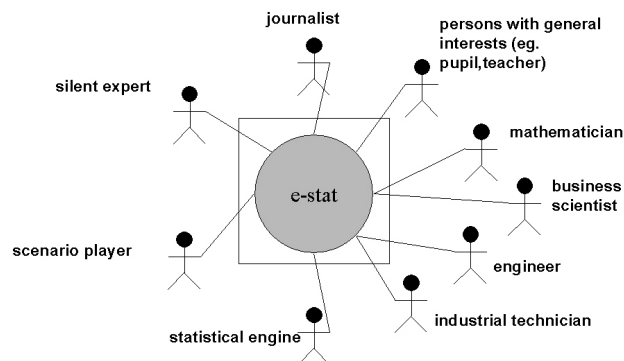


Figure 2: Use-Cases of e-stat

Scenarios are shared aggregates of situations, simulations, courses, and course units. Aggregation (shared aggregate) exists if whole-part relationship holds good between objects belonging to the corresponding classes. Objects can be part of several aggregate objects. Composition (composite aggregate) defines a stronger form of aggregation. In addition to the whole-part relationship there is the following constraint: each object belongs to only one aggregation object at one point in time⁹.

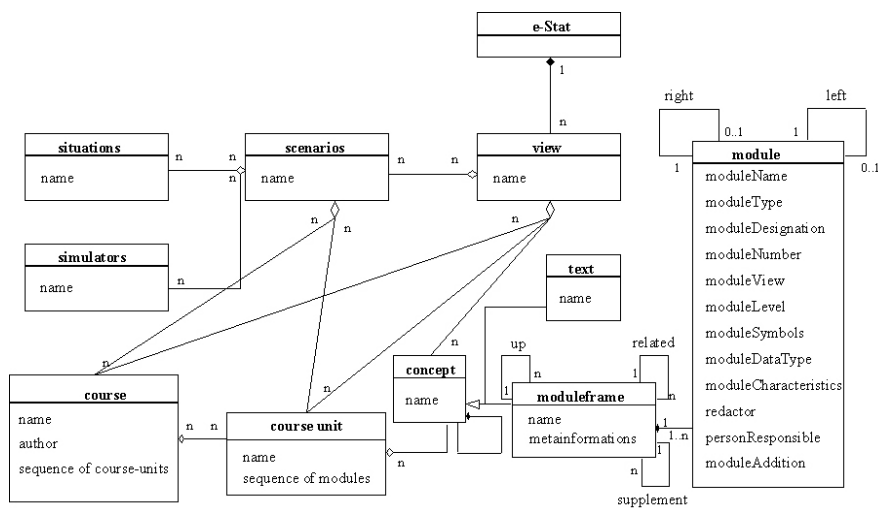


Figure 3: Class Diagram of the Content Aspect

Module frames (e.g. median) can be considered to be a single concept. Thus they can be part of any number of different views (e.g. psychological, mathematical or general view). Therefore, module frames can contain a number of different modules, which each have to be associated with a single view. To reuse modules (e.g. definitions) in a certain view they have to be referenced explicitly by that view. Modules constitute the smallest knowledge-units supplied by content providers. Modules can be of the following types: definition, expression, proof,

procedure, example, exercise, solution, comment, text, data, audio, video, XploRe, SPSS, R, etc. Types were provided to us by the statistical content providers, which are members of the mathematical department of our university. Inside module frames, modules are interlinked to define a partial order (e.g. „X depends on Y“). Modules can also point to external module frames or modules contained in external module frames. A conceptual map can be created automatically using this pointer structure. This ontology will be used for case based reasoning in the consulting component of e-stat.

2.2 Implementing e-stat using a 4-Tier Architecture

Most classes of the class diagram are implemented using the standardised XML language. XML allows to create semantic tags additional to syntactic tags (HTML). Authors receive the respective document type definitions (DTDs) to generate valid class objects (Fig. 4). We use DTDs to define elements that can exist in all documents, the attributes those elements can have, the hierarchical structure of elements within elements, and the order in which the elements will appear throughout the documents. The module structure is the result of an interactive process between domain experts, content provider, and developers.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- module: root element of module document -->
<ELEMENT module (moduleKeywords?, moduleContent, moduleAuthors, moduleSymbols, moduleBibliography?, moduleEnc
<!-- module attributes: identifying module-->
<ATTLIST module
  moduleName CDATA #REQUIRED
  moduleType (definition | theorem | proof | method | example | exercise | solution | comment | text | audio | applet | video | Xp
  moduleDesignation CDATA #REQUIRED
  moduleAddition CDATA #REQUIRED
  moduleNumber CDATA #REQUIRED
  moduleView CDATA #REQUIRED
  moduleLevel (A | B | C) #REQUIRED
  moduleDataType (ordinal | metric | nominal | none) #REQUIRED
  moduleCharacteristics CDATA #REQUIRED
>
<!-- moduleKeywords: used for searching-->
<ELEMENT moduleKeywords (#PCDATA)>
<!-- modulePosition: specifies position of module by right and left-->
<ELEMENT modulePosition (right?, left?)>
<ELEMENT right (targetModule)->
<ELEMENT left (targetModule)->
<!-- moduleContent: content (texts, images, etc) of module-->
<ELEMENT moduleContent (heading?, p*)>
----->
<!-- p: paragraph-->
<ELEMENT p (#PCDATA | b | i | u | br | center | cite | code | targetModule | a | p | table | ul | ol | formula | img | javaApplet | sa
<ATTLIST p
  id ID #IMPLIED
>
<!-- formula: only contains name of formula without '<format>'-->
<ELEMENT formula EMPTY>
<ATTLIST formula
  formulaName CDATA #REQUIRED
  id ID #IMPLIED
>
<!--img: like html img element-->
<ELEMENT img (caption?)>
<ATTLIST img
  src CDATA #REQUIRED
  align (top | middle | bottom) "middle"

```

Figure 4: The e-stat-DTD

DTDs for modules and module frames have been developed. Independent DTDs for courses (-units) and scenarios will be supplied. Modules are specified by following attributes taken from the class-diagram (Fig. 3): moduleName,

moduleType, moduleDesignation, moduleNumber, moduleView, moduleLevel, modulSymbols, moduleDataType, moduleCharacteristics, redactor, personResponsible, and moduleAddition. They are marked as required, which means their values are necessary to identify one module. Additionally, ontological links serve to construct the conceptual map of the modules, which is essential for the consulting-component.

Other optional attributes are moduleKeywords, moduleBibliography (corresponding to the BibTex-Format), moduleConnection, and modulePosition. The last two elements can both be used to define a partial order on modules presented to the students in the graphical user interface (GUI).

Figure 5 shows a screenshot of the student GUI, which is based on a sketch provided us by our statistical partners at our university. The example module presents the concept mean in German. The first row of the GUI shows the administration operations like edit (Bearbeiten), print (Drucken), options (Optionen), history (History), search (Suchen), and help functions (Hilfe). Below the administrating row is the name of the module frame: mean (Arithmetisches Mittel). The information about level, view and the current course follows. Then the partial order is presented in the form of tabs. All these rows show information about the current module. In fig.5 the tab description (Bezeichnung) is selected and the content part of the GUI contains the description of the concept mean. At the bottom of the window some external statistical engines can be chosen. The column on the left-hand side shows navigation capabilities like home, up, related and supplement. Visualizations, interactive exercises, other multimedia enrichments, and information about the authors of the module are further parts of the panel on the left hand side of the screen.

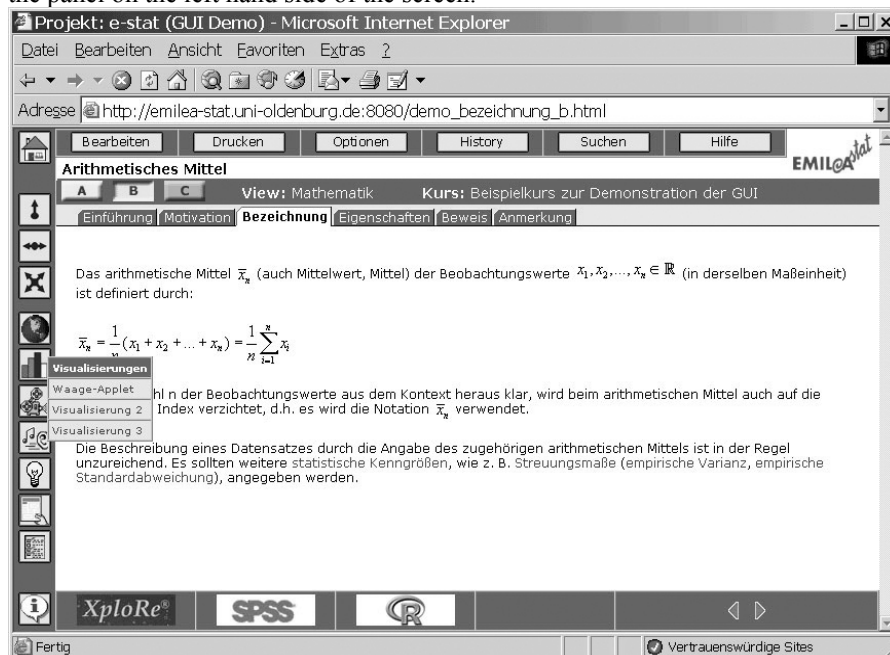


Figure 5: The Student GUI of e-stat

On the bottom right of the window the navigation for the course is located. Depending on the separation of content and layout the authors will also get a XSL-file, which is responsible for the layout. While developing content, authors have preview permanently.

e-stat is implemented using a 4-tier architecture. The presentation tier supplies content providers and students with suitable graphical user interfaces (GUIs). The view-author GUI will be powerful enough to enable authors to construct a course for his particular target group from the certified e-stat modules by means of a system similar to „shopping cart“ systems used by many e-shops. It should only be necessary to construct new modules in very special cases.

Authors can use current XML-editors (as seen in Fig. 4) instead of a special e-stat content-authoring GUI.

In the logic-tier we use an Apache server, which is installed in Oldenburg. On this server our e-stat-control-system is implemented. Interactions with our database and other statistical-engines will be managed in this tier, as well as the handling of the user-administration.

The native XML-database TAMINO is represented in the data-tier. Statistical engines (Xplore, SPSS, qs-stat) and scenario engines for simulation (e.g. handling business planning, production or stock exchange) are part of the application tier.

3. Contact

Prof. Dr. Claus Möbus, Innovative Learning and Teaching Systems, Department of Computing Systems, University of Oldenburg, D-26111 Oldenburg, Germany
<mailto:moebus@informatik.uni-oldenburg.de>

¹ Aviation Industry CBT Committee (AICC)

² Learning Technology Standards Committee (LTSC) mit den Work Groups IEEE P1484.1 Architecture and Reference Model WG: Learning Technology Systems Architecture (LTSA) und IEEE P1484.12 Learning Object Metadata WG: LOM Standards

³ Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE)

⁴ BALZERT, H., Lehrbuch der Objektmodellierung, Heidelberg: Spektrum Akademischer Verlag, 1999, ISBN 3-8274-0285-9

⁵ Förderantrag an das BMBF, Förderkennzeichen 08NM058A, Dezember 2000, Projektantrag 229

⁶ JANK, W. & MEYER, H., Didaktische Modelle, Frankfurt a. M.: Cornelsen Scriptor, 1994, ISBN 3-589-21012-5

⁷ BRUNS, B. & GAJEWSKI, P., Multimediales Lernen im Netz: Leitfaden für Entscheider und Planer, Berlin: Springer Verlag, 1999, ISBN 3-540-65428-3

⁸ BOOCH, G. & RUMBAUGH, J. & JACOBSON, I., The unified Modeling Language User Guide, Addison-Wesley, 1999, ISBN 0-201-57168-4

⁹ BALZERT, H., UML kompakt, Heidelberg: Spektrum Akademischer Verlag, 2001, ISBN 3-8274-1054-1