ISBEEM: Indoor Environment and Energy Management for School Buildings

Kostas Karatzas1, Anastasios Bassoukos1

Abstract
The present paper describes an informatics system that is being developed for indoor environment and energy management in school buildings. On the basis of a workflow analysis performed, actions and tasks related to school building maintenance and management have been recorded, while users involved have been identified (teachers, administrators, engineers). ISBEEM supports the actions and tasks that need to be undertaken towards the improvement of indoor air quality and for optimising energy usage in buildings. Schools have been selected as the studied building category due to their importance and specific characteristics when it comes to the interactions they have with children. The system under development contains technical information, practical advices and suggestions and guidelines, in order to address problems that are recorded on the basis of predefined, school-tailored questionnaires. Java-oriented, open source related technologies are used, that allow for flexible, adaptable and low cost applications to be developed. The system and its services are platform neutral, support access via the internet using PC’s, laptops, PDAs or mobile phones, and will be made available via http://isbeem.meng.auth.gr.

1. Introduction
Environmental Informatics should be realized as the combination of software and environments engineering methods and tools towards the solution of engineering problems that have a strong environmental dimension. To this end, Environmental Informatics may play an important role in the Mechanical Engineering domain, by contributing to the analysis of a Mechanical Engineering problem area of interest and to the design and development of related systems and services. The present paper addresses this area, focusing on management, knowledge engineering and decision making support in relation to indoor environment and energy management problems for schools.

2. The school building: a Mechanical Engineering problem domain
A school building can be considered as a structure that “hosts” equipment and is operated as a system, on the basis of predefined target values, via processes that take into account a large number of operational parameters. (Higgins et al., 2005). This set of parameters includes, among others, a large sub-set that falls within the domain of work of a Mechanical Engineer, like heating, cooling, ventilation, energy efficiency, indoor environment, etc. Due to the developments in the related knowledge domains and in technology, and the changes in management and administration of school buildings, modern Mechanical Engineering has increased requirements in terms of monitoring, accessing and processing of engineering information and data that sufficiently describe the operation of the school building system. Moreover, this information should be made available as widely as possible, to increase collaboration between engineers, communication of best practices and solutions to everyday problems, and enhance transparency in the management of school buildings towards the engineering and educational society. On the other hand, a large number of school building problems related to everyday operation and management are handled, at
a certain level, from non engineers, thus increasing the need for problem-focused applications that incorporate domain intelligence, and may handle multiple requests from multiple non expert users. These needs underline the importance of specialized informatics resources, appropriately tailored to user profile and needs, towards the decision making support in the technology/engineering domain. The current paper addresses the problem domain of indoor-environment and energy management in buildings, which is a multi-domain issue that stands high in the R&D agenda (Rey and Velasco, 2000), while it also touches upon relevant EU legislation (like Dir 2002/91). The application example is school buildings, due to the fact that parameters like thermal comfort, indoor air quality and energy savings are among the most important ones for this building category, a fact that has been identified from the scientific community (Daisey et al., 2003; Jalea et al., 2008; Butala and Novak, 1999; Chaloulakou and Morvidou, 2002). It should be noted that school buildings are used by a large number of people for a long time, but related indoor air quality issues have been much less studied than in other buildings. Schools frequently have serious indoor problems because of poor building construction, maintenance, cleaning and ventilation (high levels of VOCs and allergens are reported, for example). The Indoor Air Quality (IAQ) in schools can cause various short- and long-term negative health effects and discomfort, particularly in subjects with allergy. Nowadays, only a very few European countries have guidelines aimed at improving IAQ (Carrer et al., 2002).

3. Environmental Information Systems aspects in ISBEEM

Environmental Information Systems (EIS) are informatics systems concerned with the management of data about the status of the environment and related scientific, regulatory, legal, managerial, product oriented or other information. EIS are used by authorities, decision makers and the business sector alike for environmental monitoring, management, planning and coordination, environmental impact assessment, production chain support, information retrieval, etc. Due to the complexity of the processes involved, disparate data from a variety of sources must be combined. This “holistic nature” of environmental information systems leads to heterogeneity problems regarding the syntax, structure, and semantics of environmental data (Visser et al., 2001). Overcoming them requires, among others, the usage of modular and flexible architectures which are oriented towards the requirements of the end user (SOA, 2005).

In the frame of the ISBEEM project, a web-services portal is developed, for supporting the recording and management of problems related to energy and indoor environment in schools. For this purpose, a number of application modules are being built, on top of a modular platform that supports the formulation of questionnaires and the built-up of wizards, taking into account first level logic (IF-THEN-ELSE) and logical controls, in order to access a database of schools and their features. The basic idea is that the ISBEEM user, being either an expert (engineer) or not (school staff), will be guided through the application of tools and the completion of questionnaires, that will allow her/him to identify problems, specify a solution, and make use of suggestions towards its implementation. Yet, it is evident that such a system may require expertise in input data selection and preparation as well as in “running” and “post-processing” of the results. For this reason, a wizard-based approach has been adopted. A wizard is in its essence a series of screens or dialogue boxes that users follow through, guiding them to the completion of a task. Generally, each wizard screen asks users to enter information, either by making selections, or filling in fields. In the case of ISBEEM, the wizard approach helps the user in going through the complete energy and indoor air quality information collection, problem identification and management, providing a workflow scenario to be followed, accompanied by logical checks and support functions. To this end, wizards in ISBEEM should be considered application services which are designed in such a way that they:

- Help the user to apply the tools provided within ISBEEM in an easy way.
- Save time by remembering user’s previous actions and choices/decisions, when appropriate.
- Explain every step needed to continue until the final objective is accomplished.
4. System architecture

Following the principles of N-tier application architecture (URL 1) the ISBEEM system is implemented using a 4-tier architecture, where the tiers are (a) a Web Browser, (b) a Web Server, (c) an Application Server and (d) a Database. The application server in particular is composed of an ORM (object-relational mapping) layer (Torque) that eases working with relational data in an object-oriented environment.

1. an application framework (Turbine) that provides solutions to common web application tasks, such as templating, session management, authentication and authorization implementations, and component configuration.
2. a workflow engine, that directs and implements business objects between actors.
3. a templating engine (Velocity)
4. a servlet container (Tomcat) that hosts all the parts.

It should be noted that a full J2EE application server was not required, and as such we opted for a simple servlet container, thereby minimizing complexity at that level.

Fig. 1: ISBEEM system architecture

4.1 Components and technologies used

From the beginning of the software development phase of the project, several FOSS tools were considered for use, on the basis of the following criteria (Masouras and Karatzas, 2006):

1. Licensing of the components. Open-source implementations are preferred, both due to their (usually) low cost, and their openness, which allows us to inspect and adapt them to our requirements.
2. Popularity. In open-source components, the most popular ones have less bugs (due to the fact that they have been exposed to more environments, hence the bugs get detected early and there is more man-power to fix them). Popular projects usually also have better documentation, again due to more available man-power.
3. Prior fluency with the component and ease of use and integration.

On this basis, the following components and associated technologies were applied:
Web Server: Apache

Apache is the most widely used web server (as of February 2006 more than 66% of all web servers were Apache [URL 2]. Several factors contribute to this popularity:

- Apache is open-source, both Free as in cost and free as in libre.
- It is very easy to install and configure.
- Integrating other services into and existing apache server is quick and painless using the reverse proxy capabilities in the default Apache install.

On the contrary, the existing plethora of other web servers (IIS, Zeus, Sun Web Server to name a few) was not an adequate reason for adopting one of them, due to (a) cost (some of them are not open source) and (b) the fact that they require a familiarization phase that was costly for the specific development team.

Programming language: Java

Java is an excellent language for large and medium-sized projects, with an emphasis on component interoperability and a large base of open-source components. Especially in the web application space, Java has created a niche for itself for enterprise-level systems, with the advantage that several of those technologies scale down easily. Several large companies are providing services and solutions based on Java, with a healthy ecosystem existing on the smaller scale. While Java is not technically open-source, it is available free-of-charge and with a relatively open license for most platforms, and there are third-party JVMs available. Other languages that were initially considered but not selected for usage include: Perl (a good alternative with equivalent capabilities on the frame of ISBEEM), PHP (weak support of enterprise frameworks like workflows, which were considered as an essential part of ISBEEM), Ruby (upcoming, lacks of library support and third party components), and .NET (does not support adequately a Linux-based environment which was selected from the beginning as the development and operational environment of the system).

Platform: J2EE

For web-based Java applications, the J2EE spec is the industry standard. While the ISBEEM web application did not use all facilities of the specification, it is compatible with J2EE application servers. Other solutions exist, but they usually are based on subsets of the full J2EE spec, such as the OSGi-compatible plugin-based servlet container, that focuses more on mobile phone applications.

Servlet container: Tomcat

In a Java application/environment, the application server is responsible for all intra-application transactions and the management of EJB (Enterprise Java Beans) while the servlet container manages the process of JSP page generation on the basis of related requests by web clients. While an application server was not used in ISBEEM, Tomcat was naturally selected as the application container, as the development team had extensive previous experience with its use in past projects. As Tomcat is the reference implementation of several web application servlet containers, and has proven a robust choice for development and production use, the development team considered it to be an adequate choice. Other servlet containers (like Resin, WebSphere, JBoss, Jetty) were found to be more heavyweight exactly because they implement the full J2EE specification, and thus were not preferred.

JDBC

Java Database Connectivity is the API specification created by Sun Microsystems (the creator of the Java language and runtime) that allows database vendors to provide a single Java interface to their databases that programmers can code for instantly.
PostgreSQL
PostgreSQL was chosen as a database system, due to its popularity, availability as open source and robustness. Other candidate selections included:
- MySQL (current version, i.e. 4.1 was considered to suffer from some data integrity issues)
- Oracle (very good, yet non-open source and not appropriate for the application discussed)
- Microsoft SQL server (no open source, not a good fit with the development environment used in the project)
- Derby (although developed in Java, and thus closer than any other candidate to the development environment of the project, it is not considered to be mature and complete enough)

Axis
Axis is the SOAP development utilities for Java provided by the Apache software foundation, and are used for the development of web services interface to the applications.

Torque
This is an Object-Relational Mapping (ORM) tool (bridge). Other ORM solutions considered, but not selected, include Hibernate (http://java-source.net/open-source/persistence), which seems to have developed in the mostly acceptable ORM, yet it was not used due to the developer team’s unfamiliarity.

Turbine web application framework
Turbine is a servlet based framework that allows experienced Java developers to quickly build web applications. Turbine allows for personalization of web sites and for user logins to restrict access to parts of an application. Other Application frameworks considered but not selected include: Spring, Struts (not selected for avoiding conflicts with the Jacarta framework development environment). It should be noted that Turbine produces a more integrated solution without requiring third party solutions.

Maven:
This is the next generation built tool that has succeeded Ant, which was the de-facto build process automation tool for Java application development.

Development environment: Eclipse
Eclipse was chosen as the default IDE, and proved to be an extremely usable and productive development environment, especially where extensive automated (e.g. refactoring) or team (e.g. code review) operations where concerned. Other development environments considered, but not used, include: NetBeans, JBuilder, (both non open source), IDEA (equally powerful to Eclipse, yet costly). CVs was used for source code revision control

5. The ISBEEM concept
The application model consists of the following concepts:

Questionnaires, which are the top-level templates that correspond to a workflow and the associated data that will be collected. A Questionnaire is usually linked to a specific school and point-in-time. Questionnaires are defined by the site administrator and are composed of one or more sections, some of which may be optional, that are presented to the school administrator or the engineer during a workflow cycle. It should be noted that the ISBEEM system has been developed in such a way that questionnaires and other interactive sections can be created on the basis of a pallet of functions and utilities that have been developed (like multiple choice questions,
selection trees, tied selections, etc). This means that new questionnaires with a new structure can be added later on to the system.

[2] Sections are thematically connected set of questions with appropriate descriptions and instructions. A section is the building block of a questionnaire, and it can appear in several of them. Sections can have restrictions applied to them so that they are available only if some preconditions are met; for example, hiding a section about HVAC details if no HVAC is installed in the building.

[3] Questions collect from the user personal nuggets of data. Questions can be formatted in a variety of ways – Simple numbers, text strings, multiple-choice questions, free text, even rich text and images are supported.

[4] Restrictions are applied to Questions and/or Sections, limiting their visibility based on earlier answers. A restriction may be based on an answers existence or value, using either numeric ranges or regular expression matching on the contents.

The ISBEEM structure supports three main levels of indoor environment and energy related problems for school buildings

1. The elementary level, that includes the basics concerning energy efficiency of buildings, as addressed by the relevant EU legislation (Dir. 93/76/EU and Dir. 2002/91/EU). ISBEEM hosts a specialised questionnaire for covering this issue, that intends to collect information concerning the basic energy related operational parameters of the building.

2. An introduction level, where basic building related operational conditions are recorded, and on this basis guidelines are provided to the user of the ISBEEM system. These guidelines are based on a Boolean classification of responses and their parameters – associated values, and are provided following a predefined decision tree. This decision tree may be altered by the the administrator of the ISBEEM system, by introducing new logical checks, new sets of questions and answers, and interrelations between questions.

3. An Operational Level. This is where the ISBEEM user is able to combine responses coming from a decision tree, with the results of algorithms that are applied with the aid of the system. More specifically, the questionnaires that are used at this (operational) level, follow the structure of a wizard, i.e. a series of actions, based on the couple of question-answer, that users follow through the completion of the questionnaire. Each questionnaire focuses on one of the following thematic category of building management, using the classroom (or, generally speaking, a room), as the building unit of reference. Thus, the following areas of concern are addressed:
   o Thermal need calculations for the efficiency of central heating installations
   o Ventilation calculations, for estimating the need for interventions aiming at supporting existing ventilation solutions
   o Estimation of the capacity and efficiency of installed radiators, to verify that the installed central heating covers the operational needs of the specific building unit under investigation.

All these calculations are based on the technical norms and regulations that are applicable in EU, and take into account the climatic conditions of all Greek cities. It should be underlined that each question and each section within a questionnaire may be linked to another section or question of the same or even other questionnaire(s), and inherit the logical structure and dependencies of the latter. More importantly, the ISBEEM administrator has the capacity to modify existing questionnaires, add or edit sections and questions, in order to adapt it to the requirements of a specific application. In addition, all completed questionnaires are saved in the system, which allows for collaboration with business intelligence software modules, that are able to go through saved information (via SQL-based queries) to compile reports, statistics and graphs, and data mining algorithms may be applied to further investigate data and their interrelation and classification. Thus, ISBEEM is capable of supporting the formulation and fine tuning of
data models applied in various building units, for addressing indoor environment and energy management
problems. It should be noted that EPA has recently released (Jan 2006) a software tool to help school
districts establish and manage comprehensive school facility self-assessment program, the Healthy School
Environments Assessment Tool. HealthySEAT contains a fully integrated environmental health and safety
checklist and is designed to be easily customized to reflect state and local requirements and policies, yet it
does not move beyond the (very expensive and useful) checklists that it includes for the identification and
recording of any type of environmental problem related to school buildings, and is primarily targeted at
school districts and not school units (HealthySEAT, 2006). Another interesting initiative is the assessment
guidelines available under the Tools for Schools Action Kit for Canadian Schools (URL 3).

5.1 A scenario of use

The ISBEEM system is designed to be used by expert and not expert users alike, the latter presenting a
considerable challenge from the software engineering point of view. Let us consider the following
hypothesis. Suppose that in a lyceum, the board of staff members is discussing problems associated with
discomfort reports and fatigue concerning a certain class. It is also noted that this is the same class that
suffered more from absences caused by the latest flu, that seemed to struck the school lately. In order to
further investigate this issue, the board has nominated one of its members to use the ISBEEM system as a
first step of problem identification. The professor made use of the appropriate ISBEEM questionnaires,
that requires information like size and volume of the specific classroom, number and age of students,
available openings, windows and ventilation system, usage of laboratory materials in classroom, type and
operating schedule of the heating system, lights installed, etc. The ISBEEM system calculated ventilation
requirements and identified that there is a problem concerning proper ventilation of the classroom. In
addition, the fact that some of the chemicals were used in experiments conducted in the classroom, was
detected as potentially responsible for discomfort. Then, the system suggested that some measurements
of air temperature and humidity should be conducted in the classroom during its operation. When these
values were available and were fed to the system, the application calculated the discomfort index and
found that it was out of acceptable limits. In the last phase, the application provided some suggestions
like forced ventilation installation, or alternatively opening of windows more frequently.

6. Conclusions

School operation and management frequently has to deal with engineering problems like energy
consumption and savings from school buildings, and indoor environmental conditions. ISBEEM is
developing a web application that hosts a number of services, based on the wizard approach, for the
collection and processing of data, for the assessment of problems and the “production” of suggestions
based on sound engineering knowledge and experience. The ISBEEM developing environment is Java
powered, under a Code Versioning System supported by the Eclipse development environment, and with
the collaboration of various tools available via the Apache Jakarta project, for which a selection
justification is provided. First results of the ISBEEM system are very promising, and it is expected that it
will contribute to the maintenance energy usage optimisation in school buildings.

Acknowledgements

The authors wish to acknowledge the support of the Pythagoras Project, EPEAEK, 2005-6.
Bibliography


URL 1: http://www.webopedia.com/quick_ref/app.arch.asp


Copyright © Shaker Verlag 2006. ISBN: 978-3-8322-5321-9