Supporting Energy Efficiency Decisions with IT: Initial Experiences from the EnRiMa Project

Martin Henkel[1], Janis Stirna[1], Markus Groissböck[2], and Michael Stadler[2]

[1] Department of Computer and Systems Sciences, Stockholm University,
Forum 100, SE-16440, Kista, Sweden
`{martinh,js}@dsv.su.se`
[2]Center for Energy and innovative Technologies,
Doberggasse 9, 3681 Hofamt Priel, Austria
`{mgroissboeck,mstadler}@cet.or.at`

Abstract. IT solutions can aid decision makers in making informed decisions that lower the energy consumption in buildings. However, in order to design and implement an IT solution there are a number of issues that need to be resolved, for example, adequately handling sometimes contradicting goals of the decision makers and integrating the Decision Support System with the existing building IT infrastructure in the form of building management systems. In this paper we report on our initial experiences from implementing a decision support system for the management of energy consumption in public buildings. The experiences are based on our work with the EnRiMa project that aims to develop a state-of art decision support system for lowering the energy consumption and CO_2 emissions of public buildings. We divide our experiences into two areas, namely, business concerns and software architectural, and provide our initial solutions and lessons learned with respect to these areas. Furthermore, we discuss a number of challenges for future work in the area of IT support for energy efficiency.

Keywords: Sustainability, Green IT, Energy Efficiency, Decision Support System.

1 Introduction

Energy efficiency and reduction of CO_2 emissions is one of the main challenges of our society. This paper presents experiences from an ongoing EU funded FP7 project on IT solutions for improving energy efficiency of public buildings - EnRiMa (Energy Efficiency and Risk Management in Public Buildings).

The overall objective of EnRiMa is to develop a decision-support system (DSS) for operators of buildings and spaces of public use. By providing integrated decision support for handling conflicting goals such as cost minimization, meeting energy efficiency, and emission-reduction requirements, the DSS will enable operators to improve building energy efficiency in the most cost-effective manner based on their tolerances for comfort and risk. The DSS will aid the operators' in adjusting on-site

A. Kobyliński and W.A. Sobczak (Eds.): BIR 2013, LNBIP 158, pp. 315–326, 2013.

generation dispatch, and off-site energy purchases. The DSS will also enable long-term planning aimed at increasing energy efficiency. Specifically analysis of building retrofits and/or expansion of on-site energy systems, in order to meet forthcoming EU targets for reducing CO_2 emissions. Thus the system is aimed at the measurement and improvement phases [1] in a decision process.

The EnRiMa project has been running for 30 months and currently enters the user testing and validation phase of the DSS development. Hence, the objective of this paper is to present the overall business requirements and software architecture of the EnRiMa DSS as well as to discuss a number of emerging challenges pertinent to developing and running IT solutions for energy efficiency. The research findings are based on the requirements engineering and system development work performed in the EnRiMa project. We structure the description of our experiences into two areas - business concerns and software architecture. For each area we discuss the problems we have faced and how we have solved them. In order to make the solutions easier to apply in forthcoming projects, we define a set of *principles* that document the problems and the solutions. We do not claim that these are general or complete principles for building decision support systems in the area of energy efficiency, however we believe that the principles provide important input for projects with similar objectives.

The EnRiMa DSS will contain several novel features that are not present in current solutions proposed by academics or that existing in existing offers from the industry. To start with, existing building management systems, such as the Siemens Desigo system, relies on heuristic rules to optimize the use of installed equipment in order to lower a buildings energy use. The EnRiMa DSS will allow the users to also consider the effects of retrofitted equipment. Furthermore the EnRiMa DSS are capable of using advanced algorithm for handling long-term uncertainties, such as fluctuations in energy prices. This is made possible by a, for the domain of energy efficiency, novel combination of stochastic programing (SP) and a model of the long-term decision options that are available.

The remainder of the paper is organized as follows. Section 2 presents the business perspective of the EnRiMa DSS, that is, what kind of decisions making needs to be done and the types of users making the decisions. Section 3 presents the EnRiMa DSS software architecture, that is, the main software modules that the system consists of. The paper ends with a brief discussion about the challenges for adoption of the DSS.

2 The Business of Energy Efficiency – Decisions, Roles and Requirements

The key function of the DSS shall be to support the decision-maker (a person who actually makes the decisions) to find such decisions that best fits her/his preferences, expressed (usually implicitly) in terms of objectives (often called also outcomes, goals, criteria, indicators) used for evaluation of consequences resulting from implementation of a given set of decisions. The aim of the EnRiMa project is to provide the decisions makers with a DSS that could provide the best possible fundament for taking decisions that reduces the energy consumption in buildings.

Early in the project it was clear that there was several roles that made decisions that influenced the energy consumption; building owner, financial manager, operations manager, outsourced maintenance manager, energy service company, utility company (providing energy), energy consultants, policy makers, as well as energy auditors. Furthermore it was evident that these roles and their respective decisions ranged from short-term day-today decisions, such as how to best manage building's boilers and heat pumps, to more long-term decisions such as investments in alternative technologies. To cope with these different perspectives, the project early on decided to acknowledge them and treat them as equally important. We summarize this way of handling the differences in perspectives in the following simple principle:

Principle 1: Acknowledge that energy efficiency is both concerned with short-term and long-term decisions, thus, let the DSS tackle them both.

Applying the 1st principle to the EnRiMa DSS made us design an operational and a strategic module of the system. The application of the principle thus affected the way the system modeled the decision problem, this model is an important step in supporting decision making [2]. To ensure that these parts were not developed in isolation, we also followed principles for their integration. These principles will be presented later in this paper. In the following two subsections we briefly describe each of these parts.

2.1 Short-Term Operational Planning

The operational DSS module shall assist decisions on how to operate the existing building equipment. For example, this includes controlling a centralized Heating, Ventilation and Air Conditioning (HVAC) system as well as electricity and thermal storages (where available) of a building. The set of objectives includes [3]:

- Minimization of operational costs
- Minimization of environmental impact, in particular the CO_2 emissions
- Diverse indicators of the user comfort based on indoor temperature
- Analysis of the HVAC performance
- Economic dispatch of the dispatchable primary generating units. That is to start and stop or otherwise operate the installed technologies in the most economical way.

Among the parameters for operational planning typically are building facades, materials, and their thermal behavior, external temperature, external humidity, occupancy, internal temperature set-points, energy prices, as well as energy efficiency of installed equipment. In the EnRiMa DSS the given parameters are used on a daily basis to calculate the optimal way of managing the equipment for the next 24 hours. For example, by using weather forecast it is possible to estimate the solar irradiation and adjust the use of heating and cooling equipment accordingly.

2.2 Long-Term Strategic Planning

The strategic DSS module should support making long-term decisions to secure energy supply, minimize energy consumption costs and environmental impacts, and improve existing energy infrastructure. This part enables integration with elements of the operational module, which evaluates the performance of the energy system of the building in real time and triggers long-term decisions on installment of new and decommissioning obsolete technologies, retrofitting, etc. The strategic module takes into account at least the following: long-term evolution of equipment and activities; the long-term evolution of the energy consumption curves; availability of new technologies; as well as depreciation of available equipment. The set of the supported decisions is specific at each building site and typically includes:

- Selection of new devices
- Decommissioning of devices
- Selection of energy sources for electricity and heat generations, as well as for
- Cooling. This includes selecting the appropriate tariffs for buying energy the format/medium in which the energy is bought, for example, cooling can be done by district cooling or by electricity powering a HVAC.
- Financial instruments for coping with uncertainties and risks, such as futures, tariffs and inflation rates.

The following parameters are typically treated as inputs to the strategic DSS: existing building facades, materials and their thermal behavior; options for the building retrofitting; long-term climate parameters, long-term forecasts of prices and availability of energy carriers; long-term forecasts of energy loads/demand at the building; scenarios for changes of regulations affecting building operations; scenarios for changes in the energy markets.

The operational and strategic modules can run in isolation. There are however clear links between them, for example, the day-to day behavior of equipment can affect its long-term use. To strengthen this interconnection we devise the following basic principle:

Principle 2: Find links between the long-term and short term perspectives, and provide a design that supports this link in the DSS.

A concrete example of how this is done in the EnRiMa DSS is the use of profiles to capture equipment's day-to-day performance and feed this into the strategic model. For example, when making a long-term investment in photovoltaic (PV) panels it is interesting to estimate typical long-term parameters such as the evolution of their efficiency and price. However, short-term performance such as how much energy the PVs produce per day is also of interest. To express this short term data the EnRiMa DSS stores *profiles* that represent how much solar irradiation can be expected for each hour in a typical summer day, in a typical winter day etc. This short-term data is the link between the strategic model and the operational model.

Having the DSS handle information on the existing and future equipment both on the strategic and operational level can result in that the decision maker needs to know and enter a lot of information in the system. Manually entering information is cumbersome, and also makes future automation more difficult. In the EnRiMa project we therefore aim to follow a third principle:

Principle 3: As the DSS for energy efficiency handles large set of data, aim to support automation and reduce the manual work.

We follow this principle by providing a set of default values. For example, we will produce valid default profiles that describe the hourly solar irradiation over a year, and likely price development of equipment. This can substantially simplify the process of system configuration for the user, as there are predefined profiles they can choose from.

3 EnRiMa DSS – Internal and External Software Modules

To cater to the need of the decision makers, the EnRiMa DSS is constructed as a web based system where the users can access the DSS using a standard web browser. Internally the software is structured into a set of software modules; this structure constitutes the software's *architecture*. In this section we report on the experiences of creating the software architecture and its implementation. Just as for the business perspective we report on a set of *principles* that are fundamental for the way we build the system. To structure the presentation we divide it into a general overview of the software modules, a discussion about the languages used for the modules, and a presentation of the integration with external software modules.

3.1 General Software Modules

The overall architecture of the EnRiMa DSS follows the common approach of having three principal layers [4]: presentation, domain logic, and data source. In Figure 1 these layers are realized by the user interface (presentation), DSS Engine kernel (domain logic) and database (data source). Figure 1 illustrates the overall EnRiMa DSS architecture using the Unified Modeling Language (UML) component diagram notation [5].

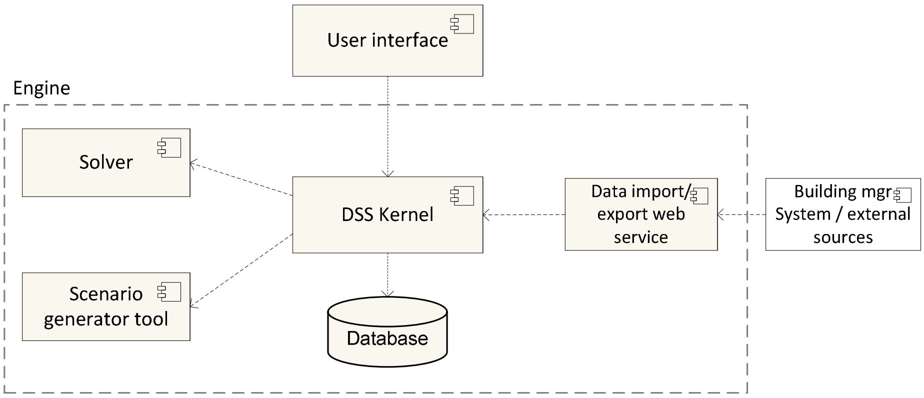


Fig. 1. Overview of the architecture

The role and contents of each part in the architecture are summarized below, a complete description can be found in [6].

The graphical *user interface* (GUI) consists of the modules responsible for the interaction with the user. The user will access the DSS via a Web browser. The user interface consists of regular and custom-made interface elements, such as buttons, tables, tabs etc. To aid in the presentation of these elements the user interface module make use of the Vaadin user interface framework, and a graph tool in the form of Google visualizations. Due to the use of these tools it is possible to create a composite application that combines various user interface elements (see example in figure 2). Currently the GUI of the DSS contains about 40 screens.

The *engine* (shown as a dotted rectangle in figure 1) is the backbone of the DSS, providing services, each through one of the DSS modules, namely the kernel, solver, and scenario generator tool.

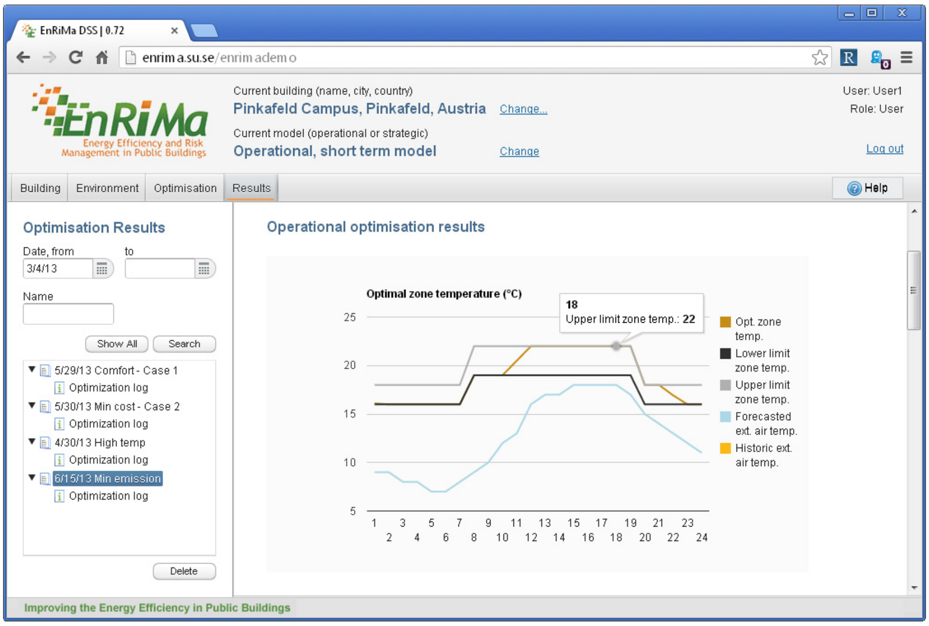
The *kernel* is responsible for providing functionality needed to manage the system data and to run the *scenario generator tool* and *solver*. The kernel also supports user handling, access control, and performing queries of the result data. The kernel is interfacing the database via a Java Persistence API (JPA) compliant object-relational mapper (ORM).

The *scenario generator tool* creates a scenario tree representing possible decisions that affect the building management. The scenario tree is based on the configuration values as set by the user via the GUI. The output from the scenario generator is fed into the solver manager that produces the solution that is shown to the user.

The *solver* is responsible for managing the calculations of the optimal use of energy and equipment. Essentially, it extracts the needed information from the database, runs an optimization and puts the result back. Upon request of the user the solver is started by the *kernel*. The time it takes to create a solution varies depending on the input parameters. The current version of the operational module, working with datasets for the next 24 hours, takes about two minutes to execute.

The data *import/export* module manages the import and export of information to external systems. This module is implemented as a web service, making it remotely accessible for external systems. External systems include both the buildings IT infrastructure as such, and external service providers. The building management systems contain information on the buildings, such as current in- and outdoor temperatures. External service providers provide information about weather forecast and energy prices.

The *user interface*, *kernel*, *solver* and *scenario generator tool* are deployed on the same server, while the *database* due to administrative reasons is deployed on a dedicated database server.



Current building (name, city, country)
Pinkafeld Campus, Pinkafeld, Austria

Current model (operational or strategic)
Operational, short term model

User: User1
Role: User

Date, from 3/4/13 to

Name

- 5/29/13 Comfort - Case 1
  - Optimization log
- 5/30/13 Min cost - Case 2
  - Optimization log
- 4/30/13 High temp
  - Optimization log
- 6/15/13 Min emission
  - Optimization log

Fig. 2. Screen capture displaying results of running optimizations for the short term operational planning module

As stated earlier the architecture is built on the commonly accepted three layer architecture. Even though the architecture follows common design principles it still contains elements that are special for the EnRiMa project. One of these special elements, or problems, is that the software modules are very different in the way they handle information and their functionality. For example, the GUI manages the data in quite a straight-forward way, i.e. the user creates and updates entities (such as description of existing building equipment) and the corresponding entities are stored in the database. However, the Scenario generator can, based on a few input entities, create thousands of data items to describe the possible scenarios. Likewise, while the GUI request is usually handled in less than a second, running the solver can take from several minutes up to several hours to run. These kinds of differences in the modules create problems in their interconnection. To cater to these problems we describe a forth principle:

Principle 4: Clearly separate the module-internal risks/problems from the inter-module problems in the software architecture.

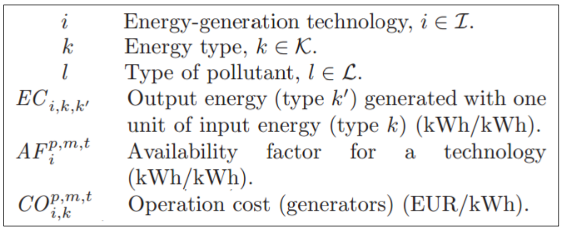
In essence this principle is useful to highlight that time need to be spent addressing the inter-module module issues. That is, time need to be spent on the module integration. For example, the integration of the kernel with the solver requires that the call from the kernel to the solver is asynchronous; in order to not keep the kernel waiting while the solver is running. Likewise the large amount of data (typically close to 100 000 entities) created by the scenario generator need to be written to the database with highly efficient code.

3.2 Languages of Use

As described earlier, the EnRiMa DSS consists of several modules that can be quite different in nature and thus attention was paid to their integration (Principle 4). However the difference in the modules goes deeper than that, and thus required the project to consider the *languages* used for designing and implementing the modules. By language we mean a set of coherent concepts, their relations and how they are represented during the design and implementation. For example, Figure 1 is expressed using a (part of) the UML language. Another example is that the DSS engine is implemented in the Java programming language. Two languages can differ in the used concepts (abstract syntax [7]), their graphical representations (concrete syntax [7]) and in their real-world meaning (semantics). Different languages have their particular area of use. While it is possible to use a few languages that have relatively little differences in a system development project (such as UML class diagrams and a relational database model), we found out that we needed to combine languages with substantially bigger differences. Before going into an example of the need to use different languages in the project we define a fifth principle to capture this need:

Principle 5: Build on the integration capabilities to leverage the power of different languages.

One of the challenges concerning the use of different languages was the difference in how the optimization problem was expressed in the solver module, compared to how it was expressed in the engine. The task of the solver is to use the huge amount of input data to find an optimal solution or solutions. A language especially equipped to express these kinds of problems is symbolic model specifications [8]. A symbolic model specification contains a mathematical representation of the input data (referred to as parameters), the output data (referred to as decision variables) and mathematical formulas (referred to as constraints) that express the desired optimal solution. An example of the parameters can be found in figure 3. Note that the relationships between parameters are defined using indices, for example the output energy type (EC) is related to a technology (i) and two energy types (k).



i	Energy-generation technology, $i \in \mathcal{I}$.
k	Energy type, $k \in \mathcal{K}$.
l	Type of pollutant, $l \in \mathcal{L}$.
$EC_{i,k,k'}$	Output energy (type k') generated with one unit of input energy (type k) (kWh/kWh).
$AF_i^{p,m,t}$	Availability factor for a technology (kWh/kWh).
$CO_{i,k}^{p,m,t}$	Operation cost (generators) (EUR/kWh).

Fig. 3. A subset of the model specification parameters (from [9])

In figure 4 a small subset of one of the constraints is shown. In essence this constraint calculates the technology operation cost (*CO*) for all types of energy output (*z*) during the optimization period (*DM*). The indices *p*, *m* and *t* refer to the time periods used for the calculation, essentially year (*p*), season (*m*) and hour (*t*).

$$\sum_{m \in \mathcal{M}} DM^{p,m} \cdot \sum_{i \in \mathcal{I}, k \in \mathcal{K}, t \in \mathcal{T}} CO_{i,k}^{p,m,t} \cdot z_{i,k}^{p,m,t}$$

Fig. 4. A small subset of the model specification constraints (from [9])

While the symbolic model specification is an efficient language to express an optimization problem, it is less suited when designing and implementing a graphical user interface. The issue here is that while the model specification contains a list of parameters, a well-structured user interface needs to have the information partitioned, for example to have all fields related to a certain technology on a certain screen. For this purpose languages such as class diagrams and the relational model is better suited as a foundation for the user interface design. A small example of a class diagram covering the same area as the model specification in figure 4 is shown in figure 5.

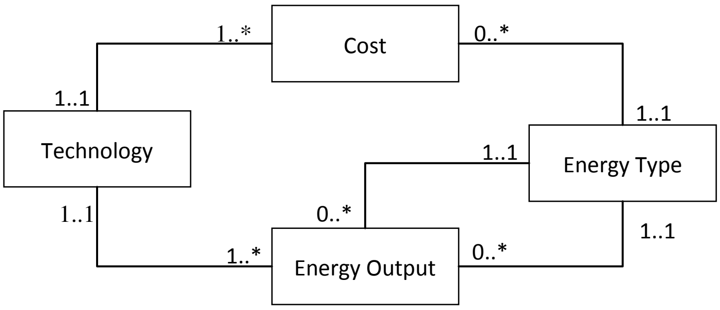


Fig. 5. Simplified UML Class diagram

3.3 Integration Requirements

The EnRiMa DSS should support the integration of both external data sources, such as data coming from building controls systems, and software modules, such as software for running optimizations and presenting results. As the EnRiMa system evolves, it is likely that the data formats that need to be handled will need to be extended, and that the number of systems that would like to make use of the EnRiMa DSS outputs will rise. We thus define the following principle:

Principle 6: A DSS for building energy efficiency cannot be run in isolation, thus look for flexible means for integration.

This principle is reflected in the system, where integrations with external weather forecasting services and building management systems (BMS) are implemented. In order to be easily extended the design of the EnRiMa DSS system considers the following integration requirements:

Protocols. The EnRiMa DSS should make use of well documented and standardized protocols for communication between software modules. For example, for integration purposes the use of Web Service protocols such as SOAP and interface descriptions such as WSDL should be used.

Data import. There are requirements for the EnRiMa DSS to import data from the Honeywell MCR building management system used at one test site, and a Siemens Desigo system used at another. Thus, a requirement on the EnRiMa DSS is that it should be built to be easily extended to allow for the import of different data formats.

Data export. The EnRiMa DSS will have built-in data representations, for example various forms of diagrams. However, there might be a need to further analyze the results from the DSS. In these cases it is important that the system supports export of the resulting data in easily accessible format. Currently the systems at the project test sites supports export of data in the form of Microsoft Excel files. Similar functionality should likely exist in the EnRiMa DSS.

External software modules. There are existing software modules that need to be a part of the EnRiMa DSS. For example the DSS will use solvers and data presentation software. These software modules should be integrated in such a way that a) they do not dictate the internal structure of the EnRiMa DSS, or limit its functionality, and b) that the software modules can be exchanged, for example when the use of a new solver arises.

4 Discussion and Future Challenges

While the presented principles cover some of the aspect the project has ran into, the project continues and we expect to define new problems and solutions. Thus, the list of principles is likely to expand. One area of particular interest is looking at new potential groups of stakeholders. The development process of the DSS has been iterative with active stakeholder involvement. While there is general interest to reducing CO_2 emissions, the main motivating factor for change still is cost reduction. Hence the DSS and similar solutions in this field should take into account the need to consolidate a multitude of stakeholder wishes and objective. Concerning the adoption of the EnRiMa DSS we foresee the following challenges:

- Integration and use of large amounts of data, (e.g. sensor, weather, occupant), information and knowledge. Sensor information is readily available in most modern buildings. E.g. the smart city project in Stockholm will have built in sensors in the newly built houses being about to generating large amounts of data, but at the moment there is no clear vision how this data could be used to reduce the carbon footprint and increase energy efficiency. This requires new theories, algorithms, and approaches for energy efficiency and environmental sustainability.
- User friendly adaptation, customization and evolvement of decisions over time. In a real building usually a diverse group of stakeholders are affected and information provided by them to be taken into account, e.g. market data and trends, technology developments, business strategies of involved actors, environmental considerations, security issues, occupant needs, as well as operator goals. The challenge is to integrate all this data into a model that can be used for decision making.
- Amalgamation of decision making based on mathematical models with business strategy development based for example on conceptual models and balanced scorecards. This would make the strategic decision making more available for people with less experience in DSS, as well as it would stimulate the learning process of the consequences of different decisions.

Furthermore, as demonstrated in section 3, the DSS should be connected to the BMS to import data and to export temperature set points. This requirement is valid for all solutions in this area, i.e. new energy efficiency solutions should be seen as add-ons to the existing BMS. What make this challenging, however, is that many BMS are proprietary systems with limited integration possibilities.

5 Conclusion

In this paper we have presented a prototype DSS for improving energy efficiency in buildings developed in EU supported FP7 project EnRiMa. Furthermore we have presented six high-level principles that reflect our experiences of developing the prototype system. These principles concerns two perspectives - business concerns and software architecture. The business concerns are covered by our recommendation to consider long and short term issues (principle 1), and their integration (principle 2). Furthermore based or the project experiences that a DSS for energy efficiency needs to handle large amounts of data we argued for the use of automatization and default values (principle 3). In the paper we also outlined the software architecture of the system and its constituents in form of software modules. Based on the experiences from the project we pointed towards the importance to tackle the differences in the software modules (principle 4), as well as the need to use different languages (principle 5). A decision support system that is clearly affected by external data sources, such as Building Management Systems in the case of the EnRiMa DDS, also needs to consider flexible means for system integration (principle 6).

The next stage of the project will test the DSS in real life settings, we thus expect to uncover more principles as the project continues.

Acknowledgements. This work has been performed in course of the European project EnRiMa ”Energy Efficiency and Risk Management in Public Buildings” and is supported by the European Commission, within the 7th Framework Programme (FP7).

Center for Energy and innovative Technologies (CET) was supported by the Austrian Federal Ministry for Transport, Innovation and Technology through the Building of Tomorrow program as well as by the Theodor Kery Foundation of the province of Burgenland in course of EnRiMa.

We also want to thank the University of Applied Science at Pinkafeld and University of Applied Science at Vienna (ENERGYbase) for their great support of the EnRiMa project. The authors also thank all other EnRiMa partners: University College London, International Institute for Applied Systems Analysis, Universidad Rey Juan Carlos, Minerva Consulting and Communication, SINTEF Group, Tecnalia, and Hidrocantábrico Energía (EDP Group) for their valuable input and support.

References

1. Loos, P., Nebel, W., Marx Gómez, J., Hasan, H., Watson, R.T., vom Brocke, J., Recker, J.: Green IT: a matter of business and information systems engineering? Business & Information Systems Engineering 3(4), 245–252 (2011)
2. Turban, E., Sharda, R., Delen, D., Efraim, T.: Decision support and business intelligence systems. Pearson Education (2007)
3. IIASA, SU, UCL, URJC, SINTEF, CET, HCE, and TECNALIA, Requirement Analysis, EnRiMa Deliverable D4.1, European Commission, FP7 Project Number 260041 (2011)
4. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley (2003)
5. OMG, OMG Unified Modeling Language (OMG UML), Superstructure, version 2.4.1, Document Number: formal/2012-05-07 (2012), `http://www.omg.org`
6. SU, IIASA, SINTEF, URJC, and CET. Draft Specification for Services and Tools. EnRiMa Deliverable D5.1, European Commission FP7 Project Number 260041 (2012)
7. OMG, OMG Unified Modeling Language (OMG UML), Infrastructure, version 2.4.1, Document Number: formal/2012-05-06 (2012), `http://www.omg.org`
8. Geoffrion, A.: Integrated modeling systems. Computer Science in Economics and Management 2, 3–15 (1989)
9. Cano, E., Groissböck, M., Moguerza, J., Stadler, M.: A Strategic Optimisation Model for Energy Systems Planning. IEEE Transactions on Power Systems (submitted, 2013)