24

Design and Development of a Compound DSS for Laboratory Research

Tomáš Hujer
University of West Bohemia in Pilsen
Czech Republic

1. Introduction

Using of decision support systems is today far away from being only the domain of top business management. DSSs are successfully applied in many areas of human activities, from traditional finance, financial forecasting and financial management, through clinical medicine, pharmacy, agronomy, metallurgy, logistics and transportation, to maintenance of machinery and equipment.

Despite this, the use of decision support systems in the domain of laboratory research is still relatively unexplored area. The main idea behind the application of DSS in this particular domain is increasing the quality and shortening the duration of research, together with reducing costs. To achieve these objectives, making the right decisions at the right time using the right information is needed. Unfortunately, the disadvantage of decision support in the field of laboratory research is mainly the lack of historical data. The rules for decision-making are still nascent during the research. This makes the issue of applying DSS for laboratory research very interesting.

It is obvious that requirements for computer support of laboratory research will vary from case to case, sometimes even substantially. On the other hand, there is a characteristic common to all laboratory research. The laboratory research consists of series of tests and measurements which generates data and knowledge as their outputs. To make a research effective, it is necessary to apply an appropriate process control to diagnostics, as well as knowledge acquisition techniques and knowledge management tools. Moreover, knowledge is very often hidden in the relationships between measured data and has to be discovered by using sophisticated techniques, such as Artificial Intelligence.

There are several options for building DSS application. This chapter is focused on in-house development as the best way to develop DSS application with maximal possible compliance with user's demands and requirements. Especially evolutional prototyping enables rapid development and deployment of the system features and functions according to the actual user's requirements. On the other hand, in-house development puts certain requirements on IT skills, which may be an intractable obstacle in some cases.

The objectives of this work are not to describe the universal, ready to use DSS, but to reveal possibilities, means and ways, to describe the methodology of design and in-house development of DSS for laboratory research with the most possible fit to user's requirements.

2. Types of DSS suitable for laboratory research

There are several classifications and taxonomies of DSS applications. A brief overview of DSS classifications brings French (French et al., 2009). For this work, the AIS SIGDSS[1] classification is used. It is today the most common and widespread DSS classification, popular with many authors, such as Power (Power, 2000) or Turban (Turban et al., 2008). This classification divided DSS into five main categories as follows:

- **Data-driven DSS**, which are primarily based on the data and their transformation into information,
- **Model-driven DSS**, which puts the main emphasis on the use of simulation and optimization models,
- **Knowledge-driven DSS**, characterized by the use of knowledge technologies to meet the specific needs of decision-making process,
- **Document-driven DSS**, that helps users acquire and process unstructured documents and web pages, and
- **Communications-driven and group DSS**, which includes all systems using communication technologies to support collaboration of user groups.

Of course, all listed categories can be combined to create compound or hybrid systems. For laboratory research purposes, the combination of data- and knowledge-driven DSS seem to be the best solution. Laboratory research is based on diagnostics. Important attributes with which the diagnostic processes work are data, information and knowledge. Knowledge transforms data into information and information (diagnosis) is the output of diagnostic process (Tupa, 2008). The diagnostic process is schematically illustrated in Figure 1.

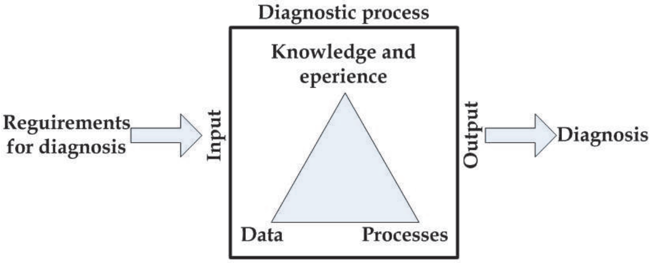


Fig. 1. Process view of diagnostics (Tupa, 2008)

As already mentioned, Data- and Knowledge-driven decision support systems are the most appropriate types of DSS for laboratory research. The next two subchapters are focused on definition, characteristics, and the key features of these two types.

2.1 Data-driven DSS

A Data-driven decision support system is defined as an interactive computer-based system that helps decision-makers use large database. Data-driven DSS primarily rely on data and their processing into information, along with the presentation of this information to a decision maker (Turban et al., 2008). The main goal of such systems is to help users transform data into information and knowledge. Users of these systems can perform

[1] Association for Information Systems Special Interest Group on Decision Support System

unplanned or ad hoc analyses and requests for data, process data to identify facts and draw conclusions about the data patterns and trend. Data-driven DSS help users retrieve, display, and analyze historical data.

This broad category of DSS help users "drill down" for more detailed information, "drill up" to see a broader, more summarized view, and "slice and dice" to change dimensions they are viewing. The results of "drilling" and "slicing and dicing" are presented in tables and charts (Power, 2000). With this category of DSS are mainly associated two technologies – Data Warehousing and Online Analytical Processing (OLAP).

Data warehouse is defined as subject-oriented, integrated, time-variant, non-volatile collection of data in support of user's decision making process. Subject-oriented means it is focused on subjects related to examined activity. Integrated means the data are stored in a consistent format through use of naming conventions, domain constraints, physical attributes and measurements. Time-variant refers to associating data with specific points in time. Finally, non-volatile means the data do not change once they are stored for decision support (Power, 2000).

OLAP and multidimensional analysis refers to software for processing multidimensional data. Although the data in data warehouse are in multidimensional form, OLAP software can create various view and more dimensional representations of data. OLAP usually includes "drill down" and "drill up" capabilities. This software provides fast, consistent, and interactive access to shared multidimensional data.

The data-driven DSS architecture involves data store, data extraction and filtering component, end user query tool, and end user analysis and presentation tool. The data store consists of database or databases built using relational, multidimensional, or both database management systems. The data in data store are summarized and arranged in structures optimized for analysis and fast retrieval of data. The data extraction and filtering component is used for extract and validate the data taken from so called operational database or from external data sources. It selects the relevant records and adds them to the data store in an appropriate format. The end user query, analysis and presentation tools help users create queries, perform calculations, and select the most appropriate presentation form. The query and presentation tools are the front-end to the DSS.

Data-driven DSS are usually developed using general development approaches called System Development Life Cycle (SDLC) and Rapid Prototyping (see Section 4), depending on the size of the resulting system.

2.2 Knowledge-driven DSS

Knowledge-driven DSS is overlapping term for the decision-making support systems using artificial intelligence technologies. These systems are usually built using the expert system shells and data mining or knowledge discovery tools. Knowledge-driven DSS era interactive programs that made recommendations based on human knowledge. This category of DSS helps users in problem solving, uses knowledge stored as rules, frames or likelihood information. In addition, these systems may have capabilities to discover, describe, and predict knowledge hidden in data relations and patterns.

According to Turban (Turban et al., 2008), expert systems (ES) are computer-based system that use expert knowledge to attain high level decision performance in a narrow problem domain. ES asks questions and reasons with the knowledge stored as part of the program about a specialized subject. This type of program attempts to solve a problem or give advice (Power, 2000).

ES consists of three elements – knowledge base, inference engine, and user interface. Knowledge base contains the relevant knowledge necessary for understanding, formulating and solving problem. Typically, it includes two elements – facts that represent the theory of the problem area, and rules or heuristics that use knowledge to solve specific problem. Inference engine is the control structure or the rule interpreter of ES. It is a computer program that derives answers from knowledge base and formulates conclusions. Inference engine is a special case of reasoning engine, which can use more general methods of reasoning. User interface is a language processor that provides user-friendly, problem-oriented communication between the users and the expert system.

The aim of data mining (DM) is to make sense of large amounts of mostly unsupervised data in some domain (Cios et al., 2007). Data mining techniques can help users discover hidden relationships and patterns in data. They can be used either for hypothesis testing or for knowledge discovery. According to Power (Power, 2000), there are two main kinds of models in data mining – predictive and descriptive. Predictive models can be used to forecast explicit values, based on patterns determined from known results. Descriptive models describe patterns in existing data, and are generally used to create meaningful data subgroups.

Data mining software may use one or more of several DM techniques. These technique and DM tools can be classified based on the data structure and used algorithm. The most common techniques are:

- **Statistical methods**, such as regression, correlations, or cluster analysis,
- **Decision trees**, that break down problems into increasingly discrete subsets by working from generalization to increasingly more specific information,
- **Case Based Reasoning**, that uses historical cases to recognize patterns (see Section 3.2),
- **Intelligent agents**, that retrieve information from (especially) external databases, and are typically used for web-based data mining,
- **Genetic algorithms**, that seek to define new and better solution using optimization similar to linear programming,
- **Neural computing**, that uses artificial neural networks (ANN) to examine historical data for patterns and applying them to classification or prediction of data relationships (see Section 3.4),
- **Other tools**, such as rule indication and data visualization, fuzzy query and analysis, etc.

Knowledge-driven DSS are usually built using several proposed Rapid Prototyping approaches.

3. Diagnostic DSS

Diagnostic decision support systems are mainly associated with the domain of clinical medicine. They are developed since the early seventies, and are designed to provide expert support in diagnosis, treatment of disease, patient assessment and prevention. Between the medical and technical diagnosis is an obvious similarity. In technical diagnosis, the examined subject is also analyzed to obtain a diagnosis. On its basis, corrective and preventive actions can be proposed and taken, and examined subject's condition can be monitored, evaluated and predicted.

Although the use of diagnostic decision support systems overlaps with other fields, such as maintenance planning and management (Liu & Li, 2007), or the prediction of product life cycle (Lolas & Olatunbosum, 2008)(Li & Yeh, 2008), the vast majority of available publications dealing with application in clinical medicine. Vikram and Karjodkar (Vikram & Karjodkar, 2009) briefly summarize the history of so called clinical decision support systems (CDSS) development. According to their findings, there are four main types of CDSS, which are based on:

- **Rule Based Reasoning** (RBR) – use of the principle of cause and effect (if-then),
- **Case Based Reasoning** (CBR) – decision making based on the principle of analogy with already resolved cases,
- **Bayesian believe networks** (BBN) – the use of probability theory,
- **Artificial neural networks** (ANN) – computational model inspired by the structure and functional aspects of biological neural networks.

Apart from these, there are few examples of the use of uncommon techniques, such as heuristic algorithms, fuzzy logic (Lingaard et al., 2007), or game theory (Lin et al., 2009). The following subchapters briefly describe the four main techniques used in diagnostic DSS and provide examples of use and suitability of these technologies for their use in laboratory research.

3.1 Rule based reasoning (RBR)

The rule based reasoning uses notation of rules in the "if-then-else" form. This notation can be extended to define the probability of suitability of the proposed actions. Rule-based reasoning is mainly used by expert systems that analyze the base of facts and apply the appropriate rules on the solved situation. The main disadvantage of RBR is laborious and time-consuming creation of a quality knowledge base.

Kumar (Kumar et al., 2009) describes a hybrid approach, combining case and rule based reasoning for branch independent CDSS for the intensive care unit (ICU). DSS that use rule based reasoning are usually limited to use in a specific area, such as cancer, poisoning, cardiology, etc. It is significantly limiting for multidisciplinary applications such as the DSS for ICU. The rigidity of the system was eliminated by combination with case based reasoning. CBR has been chosen as the main decision support technique. CBR uses RBR subsystem with knowledge base containing common rules for all medical disciplines needed at the intensive care unit.

In the area of research, the use of rule based reasoning seems to be unusable. The process of research is characterized by exploring and revealing relationships and rules within the survey data, therefore it is difficult, or impossible, to define the rules beforehand. Especially in the early stages of research is the use of rule based reasoning as a tool for decision support absolutely inconceivable.

3.2 Case based reasoning (CBR)

Case based reasoning is technique of computer-based decision-making which uses the principle of analogy with already resolved cases. It is based on the premise that the newly solved problems are often similar to previously solved cases, and therefore the previous solution can be used in current situation. The fundament of CBR is case repository, so called case library. It contains a number of previously solved cases, which are used for decision support. CBR is often used in medical DSS. One possible reason is that the reasoning based on previous cases is psychologically more easily acceptable than reasoning based on rule model (Turban et al., 2008).

Ting (Ting et al., 2010) describes a DSS integrating case based reasoning and association rules mining for decision support in prescribing of medications. According to him, CBR, unlike Bayesian networks and ANN, does not have a tendency to generalize too much, which results in superior accuracy when proposing a solution derived from the memorized cases. Association rules mining is a technique used for extraction of significant correlations of frequent patterns, clusters, or causal structures among database items.

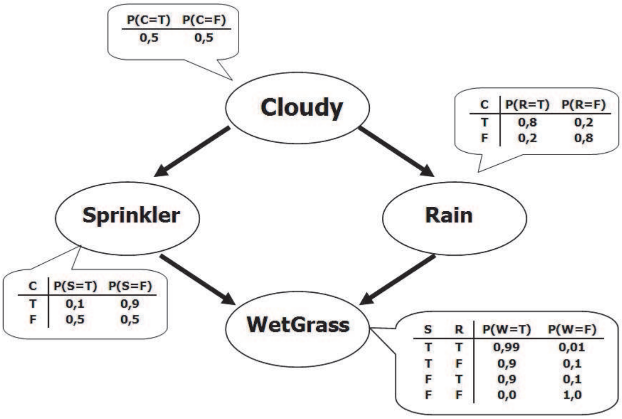
Zhuang (Zhuang et al., 2009) describes a new methodology of integrating data mining and CBR for intelligent decision support for pathology ordering. The purpose of the integration of data mining and CBR is gathering of knowledge from historical data using data mining, and retrieve and use these data for decision making support.

As the knowledge base for rule based reasoning, case library is a fundamental building block for decision support systems that use case based reasoning. For similar reasons as for rule based reasoning, CBR technique is not eligible for application in laboratory research.

3.3 Bayesian believe networks (BBN)

Bayesian believe network is a directed acyclic graph *G* = (*V*, *E*, *P*), where *V* is a set of nodes representing random variables, *E* is a set of edges representing the relationships and dependencies between these variables, and *P* represents associated probability distributions on those variables. It is a graphics model capable of representing the relationships between variables in a problem domain.

In other words, BBN is directed graphical model where an edge from A to B can be informally interpreted as indicating that A "causes" B. The example of a simple BBN is in Figure 2. Nodes represent binary random variables. The event "grass is wet" (W=true) has two possible causes: either the water sprinkler is on (S=true) or it is raining (R=true). The strength of this relationship is shown in the table below W; this is called W's conditional probability distribution (NNMI Lab., 2007).



P(C=T)	P(C=F)
0,5	0,5

C	P(R=T)	P(R=F)
T	0,8	0,2
F	0,2	0,8

C	P(S=T)	P(S=F)
T	0,1	0,9
F	0,5	0,5

S	R	P(W=T)	P(W=F)
T	T	0,99	0,01
T	F	0,9	0,1
F	T	0,9	0,1
F	F	0,0	1,0

Fig. 2. A simple Bayesian network (NNMI Lab., 2007)

Bayesian networks represent good combination of the data and prior expert knowledge. Relationships between the variables in a problem domain can be interpreted both causally and probabilistically. BBN is able to cope with the situations where some data are missing, and unlike the rule based systems allow the capture of a broader context.

The application of Bayesian probability theory in diagnostic tasks addresses Lindgaard (Lindgaard et al., 2007). He claims that the Bayesian algorithm is a suitable alternative for the diagnosis of disease, and thus to provide effective decision support in medical diagnostic systems. The result of Bayesian analysis is a set of hypotheses associated with the probability distribution. To develop decision rules, these probabilities are combined with information about the nature of possible decisions, their significance and relevance. Probabilities of Bayesian network nodes are then reviewed and further refine in each of the next iteration with new information set.

Liu and Li (Liu & Li, 2007) bring the example of using diagnostic DSS outside the field of clinical medicine. They used Bayesian networks to build a DSS for machinery maintenance, using BNN's suitability for applications in fault diagnostics. Described DSS supports the strategy of proactive maintenance based on monitoring and diagnosis of machines, and forecasting and prevention of disorders.

Using of Bayesian believe networks for the decision support poses two major problems. The first is the need of mastering the Bayesian probability theory, which means managing relatively large and robust mathematical apparatus. The second is the considerable computation complexity of algorithms for learning BBN from data, respectively difficult inference in large models.

Although Bayesian networks seem to be an appropriate technology for decision support in the field of laboratory research, above mentioned difficulties make the development of DSS applications only through own forces very difficult or even impossible for many users.

3.4 Artificial neural networks (ANN)

Artificial neural network is a computational model derived from the way of information processing performed by human brain. ANN consists of simple interconnected elements for data processing - artificial neurons. These elements process the data in parallel and collectively, in a similar way as the biological neurons. Artificial neural networks have some desirable properties similar to biological neural networks, such as learning ability, self-organization, and fault tolerance (Turban et al., 2008).

The basic element of the artificial neural network is an artificial neuron. Artificial neuron is a computation unit with inputs, outputs, internal states and parameters. This unit processes the input data (signals) and generates appropriate outputs. There are several types of artificial neurons, which vary according to the type of neural network. Generic type of artificial neuron, so called formal neuron, is shown in Figure 3.

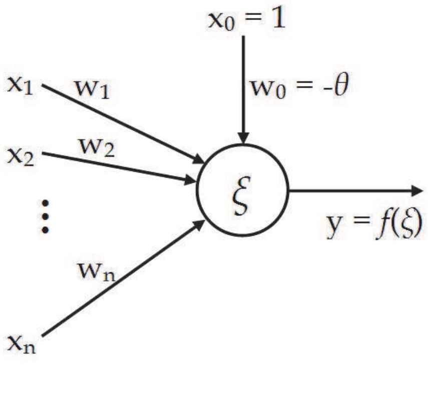


Fig. 3. Formal neuron

Formal neuron consists of several inputs ($x_1,\ldots, x_n$) and their connection weights ($w_1,\ldots, w_n$), formal input (x_0) and its connection weight, so called bias or threshold (w_0), neuron's own body, where the computation of the output is made, and one output (y), which can be further branching. The neuron function itself is divided into two steps. First, postsynaptic potential (ξ), i.e. weighted sum of all inputs, including formal input, is calculated. In the second step, so called activation function is applied to postsynaptic potential and its result is the value of the neuron output.

The connection weight is a key element of every ANN. It expresses the relative importance of each neuron input or, in other words, the degree of influence of the particular input to the output. Connection weight represents storage of patterns learned from the input data. The ability of learning lies exactly in the repeated refinement of the connection weights.

Because of their ability to learn and generalization, artificial neural networks are used in many applications of prediction and data classification. Aburas (Aburas et al., 2010) use neural networks to predict the incidence of confirmed cases of dengue fever. This prediction is based on the observations of real parameters, such as average temperature, average relative humidity, total rainfall, and the number of reported cases of dengue fever as a response to these parameters.

Faisal, Ibrahim and Taib (Faisal et al., 2010) also deal with the issue of dengue fever disease. They proposed a non-invasive technique to predict the health risks to ill patients via combination of self-organizing maps and multilayer feed-forward neural networks. Combining these techniques, they achieve 70% accuracy of forecasts.

Gil et al. (Gil et al., 2009) describe the use of artificial neural networks in the diagnosis of urological disorders. To suppress the main neural networks drawbacks, so called over-learning or over-fitting, they use a combination of three different ANN architectures, two unsupervised and one supervised. This combination has provided decision support with verified accuracy of almost 90%.

Both, Faisal (Faisal et al., 2010) and Gil (Gil et al., 2009), also mentioned the possibility of increasing the accuracy of their systems by combining artificial neural networks with fuzzy inference techniques. This approach uses Kannappan (Kannappan et al., 2010) for design the system for prediction of autistic disorders using fuzzy cognitive maps with nonlinear Hebb learning algorithm. Fuzzy cognitive maps combine the strengths and virtues of fuzzy logic and neural networks.

As mentioned earlier, artificial neural networks are often used for prediction, or to predict the probable progression of examined data. High-quality and credible prediction can be derived only on the basis of a sufficiently large volume of data. Generally, the more relevant input data is available, the more accurate is the prediction of their progression. A common epiphenomenon of materials research, as well as the development of new products, is very limited amount of data. In such cases, the neural networks must be appropriately modified to achieve an acceptable accuracy of prediction based on small data sets.

Lolas and Olatunbosun (Lolas & Olatunbosum, 2008) used ANN to predict reliability behavior of an automotive vehicle at 6000 km based solely on information from testing the prototype. To this propose, they drawn up a three-phase optimization methodology for neural network development. The proposed network can detect degradation mechanism of the vehicle and use this knowledge to predict the trend of reliability throughout its life cycle. The overall error of the whole neural network and the three output parameters were less than 9%.

Li (Li & Yeh, 2008) deals with the prediction of a product life cycle already in the initial stages of manufacturing. For work with the small data sets, they developed nonparametric learning algorithm named Trend and Potency Tracking Method (TPTM). This algorithm looks for the data trend by considering the occurrence order of the observed data and also quantifies the potency for each of the existing data by computing the TP value. It was experimentally verified that this algorithm helps to improve the performance of neural network prediction. This training mechanism conducts an incremental learning process and it is practical to earn the knowledge in dynamic early stages of manufacturing.

Li and Liu (Li & Liu, 2009) also deal with the use of neural networks for small data sets. They developed an unique neural network based on the concept of monitoring of a central data location (CLTM) for determining the network weights as the rules for learning. The experimental results confirmed the higher performance of prediction of the new network, especially in comparison with the traditional back-propagation neural networks.

Artificial neural networks offer very flexible technology broadly usable in the decision support systems. With the possibility of modification, neural networks are useful for applications which process a very limited amount of data. Laboratory research is one of the areas that are characterized by producing small data sets. One of the positive aspects of building a DSS for laboratory research using artificial neural networks is the fact that there are many commercial and free software tools for the design, development and implementation of the ANN.

3.5 Summary

The vast majority of above mentioned diagnostic DSSs are complex and robust tools with well-defined purpose, which processing huge amounts of data, representing hundreds or thousands of incidents and events, and having tens to hundreds of users. In contrast, DSS designed for use in research and development should be used by individuals with a maximum amount of data corresponding to tens of thousands events. The purpose of such system should be flexible in a certain manner, with the possibility of its definition according to the main objective of the research.

Artificial neural networks and Bayesian networks can be certainly considered as the appropriate technologies for development of DSS for laboratory research. Due to the relative inputs certainty of such system, the use of fuzzy logic seems to be somewhat excessive. Using of rule and case based reasoning logically seems to be inappropriate, mainly due to the absence or small number of already done cases, and the lack of rules for reasoning, especially in the early stages of research.

4. In-house DSS development overview

As was already mentioned in the introduction, this work is, among others, focused on in-house development as the best way to design, develop and implement DSS application with maximal possible compliance with user's demands and requirements. This part brings the short overview of in-house application development approaches and possibilities.

According to Turban (Turban et al., 2008), there are three basic approaches to DSS application development. They are:

1. build the system in-house,
2. buy an existing application,
3. lease software from application service provider (ASP).

Because of uniqueness of every research project, e.g. data types and data sources, amount of users, sharing and security requirements etc., there is basically only one solution to DSS development – building the system in-house. In the case of laboratory research, buying an existing application brings requirements for adaptation to specific demands and requirements. In some cases, the effort to adapt the application could be comparable or even higher than the effort expended to the development on your own. Finally, lease application from the third party can provide satisfactory compliance with the requirements, but with higher financial costs.

There are two possibilities of building DSS application in-house – building from scratch, or building from components. Building from scratch is suitable for specialized applications. This option provides the best compliance with the user's requirements, but can be time-consuming and expensive. Building from components uses available, commercial, freeware or open source components, and creates the required application via their integration.

In-house application development includes several development approaches and techniques. The three most common are System Development Lifecycle (SDLC), Rapid Application Development (RAD) techniques, such as Prototyping, and End-User Development (Turban et. al., 2008).

SDLC is the traditional method of application development, mostly used for large DSS projects. It is a structured framework consisting of the follow-up processes by which an application is developed. Traditional SDLC consists of four basic phases – planning, analysis, design, and implementation (PADI), which lead to deployed system. Scheme of traditional SDLC is in Figure 4.

Rapid Application Development are methodologies for adapting SDLC, so the system can be develop quickly and some of the system functionalities can be available to users as soon as possible. It is an incremental development with permanent feedback from potential users. RAD methodology breaks a system into a number of versions that are developed sequentially. Each version has more features than the previous one, so the system is developed in steps.

The most widely used methodology of RAD is Prototyping. This methodology involves performing the stages of analysis, design, and implementation concurrently and repeatedly. After each increase in development, the system is presented to the potential users.

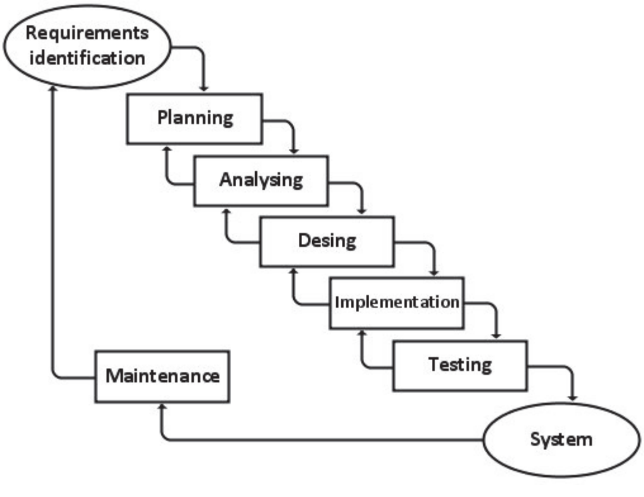


Fig. 4. Traditional application development approach - SDLC

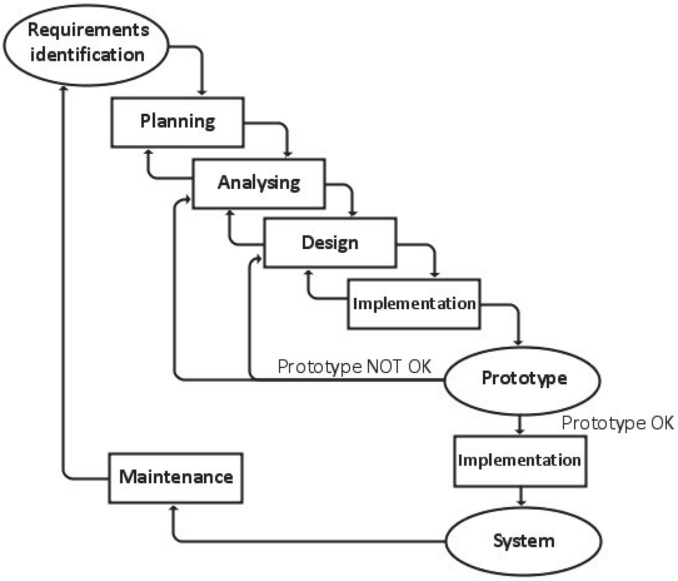


Fig. 5. Rapid Application Development methodology – Prototyping

Based on their response, further improvement takes place and the system is presented to the potential users again. After several iterations, no further improvements are proposed and the system is finally deployed. Scheme of Prototyping is in Figure 5.

End-User Development is a special case of DSS application development. The decision support tool is build by the users themselves. The advantages of this approach are undoubtedly the speed of development, the minimum cost and full compliance with the user's needs. On the other hand, there are considerable disadvantages associated with non-standard procedures of this type of development, such as inadequate documentation, improper use of development tools, the use of inappropriate technologies, or poor data security (Power, 2000).

5. Design and development of laboratory research DSS

The motivation for design and development of the decision support system for laboratory research is the need of practical tool for management and support of the research of organic semiconductors and their application in the field of vapor and gas sensors. The main aim of the proposed DSS is shortening the duration of research by revealing hidden knowledge in the measured data. Another goal is to make the research more efficient by the system's ability of manage, process, and properly present the measured data.

In the early stage, several system requirements were identified. The key requirements are: 1) requirement for presentation of measured data in a graphical form, 2) requirement for classification of the measured specimen, and 3) requirement for the prediction of material parameters of the sensitive layer according to the required response parameters.

There are also some requirements for the development. The first is to use in-house development. The second is to use only free or open source applications. Finally the third is requirement for quick adjusting and implementing of the system functions. These requirements lead to selecting Rapid Prototyping as the development approach.

5.1 Design, architecture, and development possibilities

The common structure of DSS consists of four components – data management subsystem, model management subsystem, knowledge management subsystem, and user interface subsystem. The data management subsystem includes database and database management system (DBMS). It is usually connected to the data repository, such as data warehouse. The model management subsystem is software that includes quantitative models with analytical capabilities, and an appropriate software management. It can also include modeling languages for building custom models, and can be connected to the model storage. The knowledge management subsystem provides intelligence to facilitate the decision-making process. This subsystem is optional, but highly recommended. The user interface subsystem mediates communication between the users and the system. A Schematic view of the common DSS structure is shown in Figure 6.

According to requirements and based on the common structure of DSS, the architecture of proposed DS for laboratory research was designed. Architecture of proposed DSS consists of four interconnected components – database server, application server, web server, and graphical user interface (GUI). Some of these components include additional functional units. The architecture of proposed DSS is shown in Figure 7. Functions of single components and their roles are described below.

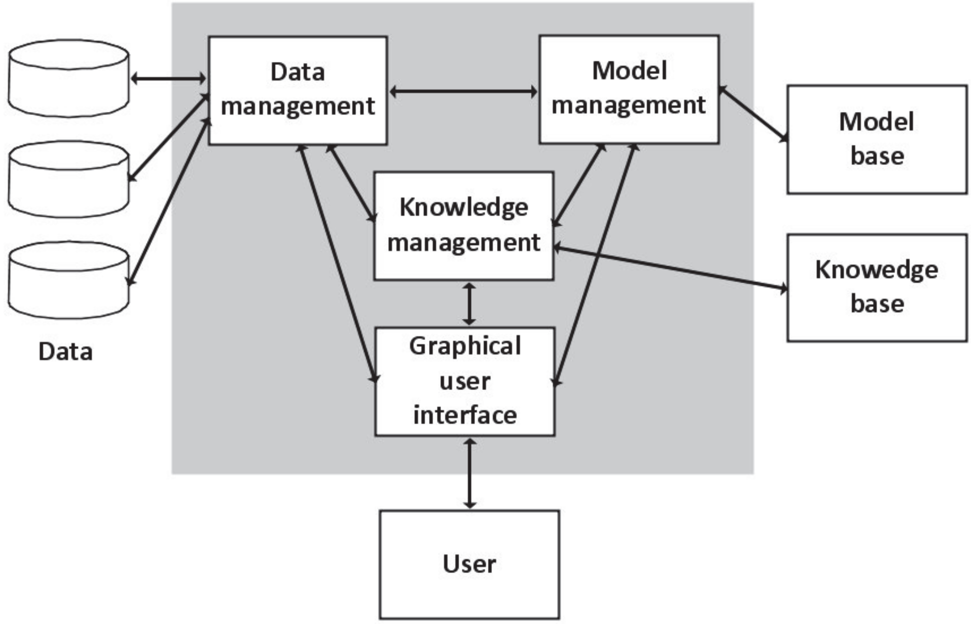


Fig. 6. A schematic view of DSS

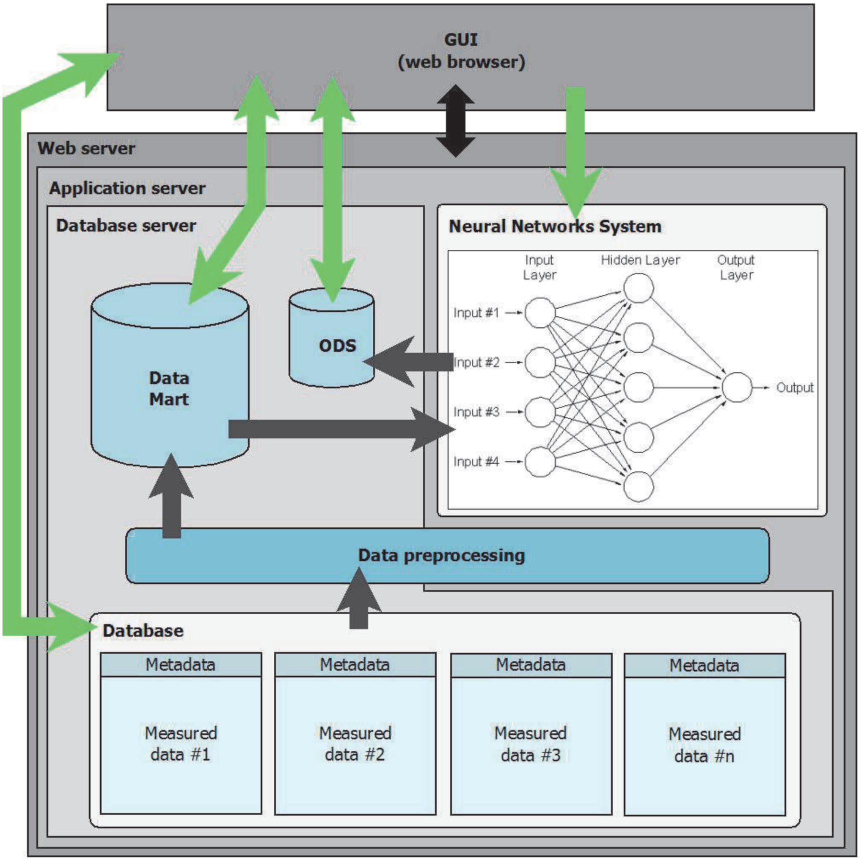


Fig. 7. Architecture of proposed compound DSS

5.1.1 Database server

The database server is a computer program that provides database services to other programs and computers. There are a number of free database servers. The most frequently used database servers are PostgreSQL, Sybase ASE Express Edition, MS SQL Server Express Edition, and Oracle Database 10g Express Edition, also known as Oracle XE. The basic criteria for choosing a database server are the dependence or independence on the platform (Windows, Unix, Mac OS), support of scripting and programming languages and standards, and the size and accessibility of the developer's community.

Based on these criteria, Oracle XE was selected as the database server for the proposed DSS. This solution provides a platform-independent, high quality and reliable database system, easy administration, and support for PHP, Java, .NET, XML and open source application. Oracle XE also involves PL/SQL language, a powerful tool for creating, storing and executing procedures applicable for the analysis and reporting. Due to the fact that Oracle is the database applications market leader, there is also large developer's community, providing support via the web discussion forums.

Within the database server, four functional units are built: relational database, data preprocessing unit, data mart (DMT) and operational data store (ODS). The main task of the relational database is to store data and metadata of all measurements and allows users to display information about the measurements and the measurements results in the form of graphs and charts. The data represent a set of measured values. The metadata contain information about particular measurement, such as the date and time of measurement, code of specimen, measurement conditions, used equipment, etc.

The data preprocessing unit, commonly known as ETL (Extract, Transform, and Load), is responsible for collecting and retrieving data from the relational database, their cleaning and adjusting into the desired form, and their subsequent loading into the data scheme of the data mart. The data have to be preprocessed for two reasons. First, the data have to be aggregated for further analysis, for instance using OLAP; second, the data have to be refined and consolidate for use in the neural network system.

Data preprocessing unit can be, with advantage, built using PL/SQL procedures. However, there are a lot of freeware ETL applications, such as Clove ETL, Pentago, Spago BI, KNIME, and more. In the case of using some of the open source applications, this functional unit will be move from the database server to the application server. It largely depends on the technology, respectively on the programming language of the application. This is also the cause of the unit overlapping between the database and application server in the diagram of the proposed DSS architecture, shown in Figure 7.

Data mart is a single subject data warehouse. Using a data warehouse technology allows performing quick ad hoc analysis, e.g. displaying data from different perspectives and in different contexts, such as used organic material, method of specimen preparation, measurement method, measurement conditions, etc. Operational data store (ODS) is the last functional unit of the database server. Its task is to store the data from the output of the neural network system and allows users to executing queries on these data.

5.1.2 Application server

The application server is a software framework designed for the effective implementation of procedures (programs, routines, scripts) to support building of applications. The task of application servers is to integrate heterogeneous environment when using multilayer architecture, and fully support the access to the various data sources. Major part of the application servers is based on the Java 2 Platform Enterprise Edition (J2EE) standard. The most popular open source application servers are Zope, JBoss, JOnAS, and GlassFish server OSE, which is a free version of the application server directly supported by Oracle.

Integration of the application server into the proposed DSS architecture mainly depends on the types and technologies of tools integrated in the neural networks system. The possibility of omitting the application server is strengthened by the fact that today's web and database servers taking over the function of application servers and allow direct integration of applications.

The main function of neural networks system is generation, preparation and application of neural networks for classification of measured data and prediction of their parameters and trends. For this purpose, it is necessary to choose a tool or tools that contain appropriate types of artificial neural networks, and are open-source enough to allow the modification of learning algorithms. There are many open source tools for creating neural networks, such as Fanny NuClass7, Joon, Encog, Neuroph, NNDef and more. Many of these are in the form of a library of some programming languages, like C + +, Perl, Python, .NET, or PHP.

5.1.3 Web server

Web server is the computer system responsible for processing requests from the client (Web browser) and transmitting the data in a network environment (Intranet, Internet), typically using HTTP (HyperText Transfer Protocol). Processing request means sending the web pages in the form of an HTML (HyperText Markup Language) document. Transmitted data can be static (prearranged data files, so called static content) or dynamic (dynamic content). Dynamic content is created on the client's request at the server side using different technologies (Perl, PHP, ASP, ASP.NET, JSP, etc.).

The proposed DSS will solely use the dynamic content. Technology used for creating the dynamic content will depend on the functions of the selected web server. Without a doubt, the most common open source web server is Apache. Other commonly used web servers are Roxen, Savant Server and nginx. There are also a number of web servers based on Java[2].

5.1.4 Graphical user interface

GUI is a very important part of the application. The quality of the user interface, its simplicity and clarity, significantly affects user's acceptation or rejection of the application. For creation of a graphical interface of proposed DSS, dynamic web pages displayed using web browser will be used. This solution is a logical consequence of the requirement for accessibility of DSS via intranet or internet.

Good possibility for building the whole DSS application is using of Integrated Development Environment (IDE), such as Eclipse or NetBeans. Both of these IDEs are Java based and support multiple programming languages. Eclipse can be used to develop applications in Java, C/C++, COBOL, Perl, PHP, Python, Ruby, Scala, Clojure, and Scheme. NetBeans includes all the tools needed for creation of professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, Python, JavaScript, Groovy, Ruby, and others.

5.2 An overview of DSS for organic semiconductors research

The laboratory research of organic semiconductors, as well as any other research, produces relatively large amount of data in the form of measured values, information about measurement's properties and conditions and information about observed subjects. On the other hand, it also produces knowledge, either explicit or directly visible, or hidden in the data. To make the research efficient and effective, the data and knowledge have to be managed and used in an appropriate form. These reasons lead to development of a web-enable compound DSS which combines elements and characteristics of data-driven and knowledge-driven decision support system.

The objectives for development of the decision support system for laboratory research were build the system in-house and use only free or open source applications and tools. That leads to investigation and testing of several applications which were supposed to be suitable for DSS development. The most of them are already mentioned in previous sections. An overview of applications, tools, and techniques used for development of a compound DSS for organic semiconductors research is given below.

[2] For more details see http://Java-source.net/open-source/web-servers

The data-driven part of the DSS is based on Oracle XE and uses PL/SQL for data processing routines. This part is responsible for storing the raw measured data and their metadata as well as their preprocessing such as filtering, computing, transforming etc. It is also responsible for executing user's queries and analysis.

Due to relatively expensive and time-consuming measurements associated with organic semiconductors research, the main attention was paid to the knowledge-driven part of the DSS. The main aim of this part is satisfactory prediction of observed parameters based on several initial measurements. The secondary aim is discovering of hidden relationships and patterns in measured data.

For these purposes, the artificial neural network approach was chosen. The neural network system of the DSS is build using Neuroph – a Java neural network framework. This open source product allows development of common neural network architectures. Neuroph also provides GUI neural network editor and includes own IDE based on NetBeans platform.

For prediction tasks of the DSS, mainly the multilayer Perceptron (MLP) architecture with backpropagation training algorithms is used. The neural networks are deployed as Java applications. These ANN applications are integrated and executable using GlassFish Server OSE. It is an open source Java EE compatible application server. It is the free version of Oracle GlassFish Server. It provides more or less the same functionality with a broad support and developer community.

Undoubtedly, the best approach for building decision support system for laboratory research is evolutionary prototyping. Laboratory research projects are not static and it is clear that the requirements for decision-making support will still appear and change during all phases of research. Evolutionary prototyping approach enables developers to flexibly respond to current needs and requirements of users. In this way, the new functions and functionalities of the system could be implemented on demand and the system itself could be constantly up to date and satisfactory.

6. Conclusion

The usage of decision support systems in the field of laboratory research is still relatively unexplored area. The main aims of deployment of DSS for research purposes are shorten the duration of research and make the research more efficient. These objectives can be successfully achieved using artificial neural networks. Using DSS also brings the advantages in managing and processing of related data.

Such a system need to be built in the shortest possible time, and precisely tailored to the user's requirements. For these reasons, the in-house application development using evolutionary prototyping has been chosen as the most satisfactory approach. The architecture of proposed DSS consists of four interconnected components – database server, application server, web server, and graphical user interface. The application server is more or less optional, dependent mainly on functions of database and web server, and on the requirements of the neural network system.

This work proposes the approach to building decision support system for laboratory research. Based on characteristics, properties, and demands of laboratory research, the appropriate DSS types are discussed. Selection of applicable technology is derived from the capabilities of the four main categories of diagnostic DSS, used mainly in clinical medicine.

7. References

Aburas, H. M.; Cetiner, B. G. & Sari, M. (2010). Dengue confirmed-cases prediction: A neural network model. *Expert Systems with Applications*, Vol. 37, No. 6, (June 2010), pp. 4256-4260, ISSN 0957-4174

Cios, K. J.; Pedrycz, W.; Swiniarsky, R. W. & Kurgan, L. A. (2007). *Data Mining, a knowledge discovery approach*, Springer Science + Business Media, LLC, ISBN 978-0-387-33333-5

Faisal, T.; Ibrahim, F. & Taib, M. N. (2010). A noninvasive intelligent approach for predicting the risk in dengue patients. *Expert Systems with Applications*, Vol. 37, No. 3, (15 March 2010), pp. 2175-2181, ISSN 0957-4174

French, S.; Maule, J. & Papamichail, N. (2009). *Decision behaviour, analysis and support*, Cambridge University Press, ISBN 978-0-521-70978-1, Cambridge, UK

Gil, D.; Johnsson, M.; Chamizo, J. M. G.; Paya, A. S. & Fernandez, D. R. (2009). Application of artificial neural networks in the diagnosis of urological dysfunctions. *Expert Systems with Applications*, Vol. 36, No. 3, Part 2, (April 2009), pp. 5754-5760, ISSN 0957-4174

Kannappan, A.; Tilarasi, A. & Papageorgiou, E. I. (2010). Analyzing the performance of fuzzy cognitive maps with non-linear hebbian learning algorithm in predicting autistic disorder. *Expert Systems with Applications*, Vol. 38, No. 3, (March 2011), pp. 1282-1292, ISSN 0957-4174

Kumar, K. A.; Singh, Y. & Sanyal, S. (2009) Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU. *Expert Systems with Applications*, Vol. 36, No. 1, (January 2009), pp. 65-71, ISSN 0957-4174

Li, D. & Liu, C. (2009). A neural network weight determination model designed uniquely for small data set learning. *Expert Systems with Applications*, Vol. 36, No. 6, (August 2009), pp. 9853-9858, ISSN 0957-4174

Li, D. & Yeh, C., (2008). A non-parametric learning algorithm for small manufacturing data sets. *Expert Systems with Applications*, Vol. 34, No. 1, (January 2008), pp. 391-398, ISSN 0957-4174

Lindgaard, G.; Pyper, C.; Frize, M. & Walker, R. (2009). Does Bayes have it? Decision Support Systems in diagnostic medicine. *International Journal of Industrial Ergonomics*, Vol. 39, No. 3, (May 2009), pp. 524-532, ISSN 0169-8141

Liu, Y. & Li, S. (2007). Decision Support for Maintenance Management Using Bayesian Networks. *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, ISBN 978-1-4244-1311-9, Shanghai, September 2007

Lin, C.; Lin, C. M.; Lin, B. & Yang, M. (2008). A decision support system for improving doctors' prescribing behavior. *Expert Systems with Applications*, Vol. 36, No. 4, (May 2009), pp. 7975-7984, ISSN 0957-4174

Lolas, S. & Olatunbosun, O. A. (2008). Prediction of vehicle reliability performance using artificial neural networks. *Expert Systems with Applications*, Vol. 34, No. 4, (May 2008), pp. 2360-2369, ISSN 0957-4174

NNMI lab. (2007). Bayesian networks, In: *NNMI Laboratory research*, 2010, Available from: < http://nnmi.kaist.ac.kr/res_contents/bayesian_eng.html>

Power, D. J. (2000). *Decision Support Systems Hyperbook*, Cedar Falls, IA: DSSResources.COM, Retrieved from <http://dssresources.com/dssbook/>

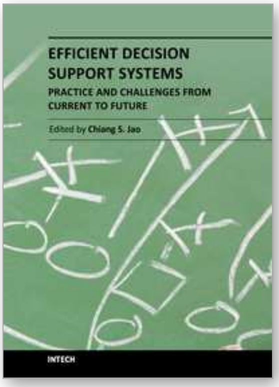
Ting, S. L.; Wang, W. M.; Kwok, S. K.; Tsang, A. H. C. & Lee, W. B. (2010). RACER: Rule-Associated Case-based Reasoning for supporting General Practitioners in prescription making. *Expert Systems with Applications*, Vol. 37, No. 12, (December 2010), pp. 8079-8089, ISSN 0957-4174

Tupa, J. (2008). *Control of diagnostics processes* (habilitation thesis), University of West Bohemia in Pilsen, Czech Republic

Turban, E.; Aronson, J. E.; Liang, T. & Sharda, R. (2008). *Decision Support and Business Intelligence Systems* (eight edition), Pearson Education, ISBN 0-13-158017-5, Upper Saddle River, New Jersey

Vikram, K. & Karjodkar, F. R. (2009). Decision Support Systems in Dental Decision Making: An Introduction. *Journal of Evidence Based Dental Practice*, Vol. 9, No. 2, (June 2009), pp. 73-76, ISSN 1532-3382

Zhuang, Z. Y.; Churilov, L.; Burstein, F. & Sikaris, K. (2009). Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners. *European Journal of Operational Research*, Vol. 195, No. 3, (16 June 2009), pp. 662-675, ISSN 0377-2217



Efficient Decision Support Systems - Practice and Challenges From Current to Future

Edited by Prof. Chiang Jao

ISBN 978-953-307-326-2
Hard cover, 542 pages
Publisher InTech
Published online 09, September, 2011
Published in print edition September, 2011

This series is directed to diverse managerial professionals who are leading the transformation of individual domains by using expert information and domain knowledge to drive decision support systems (DSSs). The series offers a broad range of subjects addressed in specific areas such as health care, business management, banking, agriculture, environmental improvement, natural resource and spatial management, aviation administration, and hybrid applications of information technology aimed to interdisciplinary issues. This book series is composed of three volumes: Volume 1 consists of general concepts and methodology of DSSs; Volume 2 consists of applications of DSSs in the biomedical domain; Volume 3 consists of hybrid applications of DSSs in multidisciplinary domains. The book is shaped upon decision support strategies in the new infrastructure that assists the readers in full use of the creative technology to manipulate input data and to transform information into useful decisions for decision makers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tomáš Hujer (2011). Design and Development of a Compound DSS for Laboratory Research, Efficient Decision Support Systems - Practice and Challenges From Current to Future, Prof. Chiang Jao (Ed.), ISBN: 978-953-307-326-2, InTech, Available from: http://www.intechopen.com/books/efficient-decision-support-systems-practice-and-challenges-from-current-to-future/design-and-development-of-a-compound-dss-for-laboratory-research

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.