VSB – Technical University of Ostrava

Faculty of Mechanical Engineering

The Department of Control Systems and Instrumentation

A DIAGNOSTIC EXPERT SYSTEM AS A TOOL FOR TECHNOLOGY IMPROVEMENT SUPPORT

Doctoral Dissertation

Student: *Ing. David Plandor*

Study program: *Mechanical Engineering*

Field of study: *Automation of Technology Processes 3902V010*

Supervisor: *Doc. Ing. Lenka Landryová, CSc.*

Submission: *08/2017*

Ostrava 2017

STUDENT AGREEMENT

I certify that this doctoral dissertation including attachments is a result of my individual research and development under the guidance of my supervisor and I have stated all resources used for its completion.

Ostrava 24th August 2017

David Plandor

ACKNOWLEDGEMENTS

My deepest gratitude is due to my supervisor doc. Ing. Lenka Landryová, CSc., who made me interested in the breathtaking world of expert systems and their applications and who got me involved in FutureSME project. Hereby I would like to express my appreciation for being a patient advisor, for supporting me with constructive criticism and new ideas.

As part of work has been done during several visits to Department of Design, Manufacture & Engineering Management (DMEM); University of Strathclyde Glasgow, UK, I also would like to express my most sincere gratitude to professor Umit Bititci for his invitation and Dr. Catherine Maguire for cooperation in development of Capability Diagnostic.

Special thanks go to Garry Smith, a CEO of Sirius Concepts Ltd., for his invitation to Ayr in Scotland, valuable assistance and support he provided me during implementation of Capability Diagnostic into the FutureSME portal.

Finally, I am forever indebted to my family for their understanding, endless patience and encouragement when it was most required.

There must be a lot of important people I have not thanked yet – thank you very much!

ABSTRACT

This dissertation focuses on design, modelling and development of a diagnostic expert system, which was implemented as a tool (named Capability Diagnostic) in FutureSME project (and web portal) as one of the tools for self-diagnostic of SMEs, where according to analysis of current state the output data were generated. These data were used for creation of an action plan, which serves as a list of improvements that need to be done to solve crucial processes of the company. After improvements in company processes were completed, a new diagnostic process was initiated and a comparison with previous results was performed. This tool was evaluated by companies partnering to the project as one of the most contributive results of the project. Design, modelling and development of the system were focused on general use of the diagnostic system for company processes and improvement support.

KEYWORDS

diagnostic system, expert system, tool for SME, general diagnostic system, self-assessment

ABSTRAKT

Tato disertační práce se zabývá návrhem, modelováním, vývojem a realizací obecného diagnostického expertního systému, který byl jako nástroj (nazván jako Capability Diagnostic) nasazen v rámci projektu (a portálu) FutureSME jako jeden z nástrojů pro diagnostiku malých a středních podniků, kde na základě analýzy aktuálního stavu generoval výstupní data, která byla použita pro vytvoření akčního plánu, na základě kterého byly provedeny zásahy do chodu firmy, které měly napomoci k řešení klíčových procesů jejího fungování. Po zavedení opatření byla opakovaně provedena diagnostika a výsledek byl porovnán s původním či předchozím stavem. Tento nástroj byl partnery projektu (majiteli či řediteli firem) hodnocen jako jeden z nejpřínosnějších v tomto projektu. Návrh, model i vývoj systému byl zaměřen na obecné využití i v jiných oblastech (nejen průmyslových) pro zlepšování firemních procesů.

KLÍČOVÁ SLOVA

diagnostický systém, expertní systém, diagnostika procesů podniku, hodnocení vlastních procesů

CONTENTS

LIST OF ABBREVIATIONS

ACID	Atomicity, Consistency, Isolation, Durability
AI	Artificial Intelligence
API	Application Programming Interface
CD	Capability Diagnostic
CLI	Common Language Infrastructure
CLS	Common Language Specification
CMS	Content Management System
CTS	Common Type System
CRUD	Create, Read, Update and Delete
DDB	Dependency Directed Backtracking
DM	Decision Maker
DMEM	Department of Design, Manufacture & Engineering Management
DP FMEA	Dynamic Positioning Failure Mode Effect Analysis
EA	Evolutionary Algorithm
EFQM	European Foundation for Quality Management
ER	Evidential Reasoning
ERP	Enterprise Resource Planning
ES	Expert system
EU	European Union
FCL	Framework Class Library
FMA	Failure Mode Avoidance
FMEA	Failure Mode Effect Analysis
FOL	First Order Logic
GA	Genetic Algorithm
GDS	General diagnostic system
GPS	General Problem Solver
GUI	Graphic User Interface
HTML	Hypertext Markup Language
IIS	Internet Information Services
IoS	Internet of Services
IoT	Internet of Things
IP	Internet protocol
IS	Information System
JESS	Java Expert System Shell

KPI	Key Performance Indicators
LAMP	Linux, Apache, MySQL, PHP
LAN	Local Area Network
LISP	LISt Programming
MCDA	Multi-Criteria Decision Analysis
MCDM	Multi-Criteria Decision Making
MOCO	Multi-Objective Combinatorial Optimization
MSIL	Microsoft Intermediate Language
MSSQL	Microsoft SQL
MySQL	Open-source relational database management system
PDA	Personal Digital Assistants
PHP	Personal Home Page, PHP: Hypertext Pre-processor
R&D	Research & Development
RDBMS	Relational Database Management System
SAW	Simple Additive Weighing
SME	Small and Medium Enterprise
SOA	Service Oriented Architectures
SQL	Structured Query language
TCO	Total Cost of Ownership
TDS	Tabular Data Stream
TQM	Total Quality Management
UDT	User-Defined Type
UML	Unified Modelling Language
URL	Uniform Resource Locator
WSM	Weighted Sum Model
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language
YACC	Yet Another Compiler-Compiler

1 INTRODUCTION

Micro, small and medium-sized enterprises (SMEs) are the engine of the European economy. They are an essential source of jobs, create entrepreneurial spirit and innovation in the EU and are thus crucial for fostering competitiveness and employment. That is why EU began to support European SMEs, a major step towards an improved business environment for SMEs and aims at promoting entrepreneurship, investments and growth. In the enlarged European Union of 28 countries, some 23 million SMEs provide around 75 million jobs and represent 99% of all enterprises. Main advantages offered by SMEs to big business are personalized service, better access to market information, greater flexibility to adapt to market changes, more rapid and timely decisions or increased worker productivity. However, they are often confronted with market imperfections. SMEs frequently have difficulties in obtaining capital or credit, particularly in the early start-up phase. Their restricted resources may also reduce access to new technologies or innovation. Therefore, support for SMEs is one of the European Commission's priorities for economic growth, job creation and economic and social cohesion.

The Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering, VSB – Technical University of Ostrava was involved in an EC FP7 funded project, FutureSME, investigating the sustainability and competitiveness of European SMEs. The project began in January 2009 and finished in December 2012. It involved 26 partners that included 13 SMEs and 13 R&D partners representing 8 European countries. The main goal of our department was to collaborate with FutureSME partners in research and development, supporting the development of web portal and its content providing tools for SMEs.

This dissertation summarises a part of the project and development of the methodology and tool implementing into practice accompanied with educational material helping SMEs in reinforcement of their position on the market without a need of hired advisors, additional significant and repetitive investments. The main goal of the research is to design and develop a software platform that would be useful for technology improvement – processing input data and generating results. First opportunity for real utilisation is the Capability Diagnostic (CD) model designed according to methodology of Department of Design, Manufacture & Engineering Management (DMEM); University of Strathclyde Glasgow, UK. Furthermore, the research leads to evaluation of general parameters and restrictions of the system and suggests types of real applications.

2 CURRENT STATE

The Capability Diagnostic model designed by DMEM originates from an analysis of current diagnostic systems focused on specific area of utilisation. All of them are single-purpose - used in industry for bug localisation [TIHHOMIROV et al., 2013], in medicine as knowledge-based medical diagnostic and treatment advice [ILIIF, 1997] or for diagnostic consultation in psychiatry [COHEN et al., 2011]. More flexible systems assist an operator in finding the cause of a machine's decreased performance. They are general in the meaning of using it for several machines [SOVARONG et al., 1997]. A methodology for development of model-based diagnostic system describes a way of diagnostic based on comparing real system with a model [CHANTLER et al., 1995].

Diagnostic systems principles are mostly based on conventional probability theory [SOVARONG et al., 1997], rule based methods [ABDULLAH et al., 2009], multiple-criteria decision analysis [MULLINERA et al., 2013], using query-based neural networks [CHANG, 2005], hybrid neural networks [KIM et al., 1997] or semantic web approach [YACHIKU et al., 2009].

In order to design a self-assessment diagnostic system a deeper study of organisational evaluation methods is a prerequisite as well as overall diagnostic process and its pitfalls. Self-assessment against the European Foundation for Quality Management (EFQM) Excellence Model as typically outlined by [HAKES, 1999] can be considered to be part of the general class of organisational evaluation problem. Most of the leaders on a market with business diagnostic tools offer strategy development for a company during a consultancy session with the help of advisers [RapidBI, 2007], recruiting and retaining skilled workers [The Building Small Business Human Resources (HR) Quiz, 2008], or may be oriented specifically at a given industry in order to deal with the highly dynamic circumstances in which industries around the world now operate [ANDERSEN, 2005]. Others are evaluating business performance going through a number of simple questions ranging from sales turnover, productivity and employee performance figures [MCANDERSON, 2010]. The results are offered in a form of an individual or team report supplied during a follow up meeting with a full explanation and advice for advancements, opportunities and clarification. [Small Business Diagnostic Tool, 1992] [GADD, 1995] [RITCHIE et al., 2000] [Business Diagnostics, 2004]

The main difficulty is the requirement for attributes to be potentially independent and quantitative. Measurements of qualitative attributes are highly subjective and imprecise and, as such, it is more effective to evaluate those using subjective judgements with uncertainty. This is a typical problem - self-assessment against the criteria of the EFQM Excellence Model - in which the process of scoring is influenced by: the background and prejudices of the assessors; how they interpret an organisation's approach; deployment, and assessment and review; their level of understanding of the TQM/excellence/continuous improvement philosophy and the sub-

criteria under consideration; and how they interpret the detail in an organisation's submission document. As a consequence there is some degree of subjectivity and uncertainty in the scoring activity and, as a result, the process is not scientifically robust with respect to numerical evaluation. The evidential reasoning approach provides an alternative way to deal with such uncertain synthesis problems by means of evidence combination for multiple attributes. [SIOW et al., 2001]

Diagnostic systems are used in most modern tools for process or quality improvement i.e. for value stream analysis in lean management, analysis in six sigma or Total Quality Management (TQM). "Risk management" appears in the index as a result of suggestions from the interviewees, since they thought that there should be a standardised way of working with a safer and better delivery. There were a lot of references to the usage of tools such as Failure Mode Avoidance (FMA) and Failure Mode Effect Analysis (FMEA). Both these aspects are significant to take into consideration since they both emphasises the importance of doing the right things in the right way. The costs to adjust errors and mistakes increases with time and there is a lot of money to be saved by implementing standardised work routines that can prevent errors in later stages. [ERICSSON et al., 2012]

FMEA includes gathering and review relevant information, evaluating the risk and prioritizing and assigning corrective actions. All these steps are logical parts of general diagnostic system. Guidelines provided for dynamic positioning FMEA's sometimes point towards complicated requirements of FMEA which may require formatted text, worksheets, test cases and their results, and lists of questions & their answers, recommendations and operations advice. There is also the need for a document management system to manage different versions of FMEA document and related versions of design documentation. In near future, there may also be the need to integrate electronic templates and sample data provided by various classification societies and forums. Basic, but essential, software tools such as Microsoft Excel and Word may not be sufficiently equipped to cater to all these needs. A DP FMEA specific software will help the FMEA experts to focus on creating content rather than worry about structure and acquiring advanced excel and word skills. [KAUSHIK et al., 2014]

This dissertation will research common methods and algorithms to find a most suitable set for general purposes of designed diagnostic system. As each and every method has its own advantages and restrictions, it leads to have a set of methods or blend methods to profit from all of them.

3 OBJECTIVES

The dissertation is based on research of current computational methods used in diagnostic expert systems. Initial identification of algorithms leads to modelling a general diagnostic system, encapsulation of a diagnostic model into the system and its implementation as well as tuning and testing with end-users. The diagnostic model (scheme) that has to be used with the system was being developed in cooperation with Department of Design, Manufacture & Engineering Management (DMEM); University of Strathclyde Glasgow, UK based on a team know-how developed during this project cooperation. Diagnostic model is used for Capability Diagnostic – process improvement tool used by SMEs for diagnosing their current state in several fields on business and helping them to improve by summarising the results and creating action plan or other ways i.e. continuous monitoring of key performance indicators (KPI). The final implementation on the server was launched and provides users with remote access via web portal. Overall system functionality was approved by partnering academics and mainly representatives of European SMEs within the project. The system and the model were being improved in a cycle of comprehensive analysis of all reviews and comments. Acquired diagnostic data are high-valued source for subsequent processing so additional techniques are thus implemented to support generation of KPIs or action plan. The requirements for development of computational methods and algorithms are focused on generality and flexibility for any other instance of the diagnostic system. The dissertation will also include verification of correct system solution after being implemented in practice and its convenient features for technology optimisation.

The final dissertation objectives are:

- Research of current state in decision making systems working in environments dealing with uncertainty type of data acquired from production processes. Research of methodology used for designing a system capable to process standard as well as non-measurable data considering uncertainty and tacit knowledge.
- Identification of computational methods suitable for processing manually-constructed taxonomies of data in knowledge-based or expert systems.
- Design of a general diagnostic knowledge-based or expert system for processing input data and output data generation with a separated functional environment (interpreter) and a diagnostic model (scheme) part, with consideration of the rules for a model template to be able to be implemented.
- Modelling an instance of the diagnostic system with designed data acquisition; a rule based data processing triggered by a user, and a report generation design.

- Development of a server-client solution with web presentation and its pilot implementation. Live operation - testing with end-users in practice, improving the system on the basis of testing and reviews.
- Data post-processing techniques design according to the needs or requirements from pilot implementation regarding commonly used and modern computational methods.
- Generalisation of the designed system according to practical exploitation and results for technological process optimisation.
- Summarising the contribution of designed solution to current diagnostic methods and propound a future development.

4 METHODOLOGY

All methodology used in this dissertation is divided into 5 sections dealing with description of expert systems, their architecture, functionality, key features and ways of their designing; processing methods used for subsequent design of my own original diagnostic expert system; object oriented programming as a standard for development including available programming environments and database systems for storing expert system data. All of these were finally used for system design, modelling and development.

4.1 EXPERT SYSTEMS

Expert system (ES) is a computer system that emulates the decision-making ability of a human expert. [JACKSON, 1998]. Expert systems are designed to solve complex problems by reasoning about knowledge, represented primarily as if–then rules rather than through conventional procedural code [Pcmag.com, 2013]. The first expert systems were created in the 1970s and then proliferated in the 80's. Expert systems were among the first truly successful forms of Artificial Intelligence software.

Reasoning, where deduction proceeds from general to specific, induction proceeds from specific to general:

- **Deductive reasoning** – deals with exact facts and exact conclusions
- **Inductive reasoning** – not as strong as deductive – premises support the conclusion but does not guarantee it, works with hypothesis under uncertainty conditions. [MISIASZEK, 2007]

An expert system has two parts or sub-systems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging capabilities.

4.1.1 KNOWLEDGE BASE

It contains domain-specific and high-quality knowledge. Knowledge is required to exhibit intelligence. The success of any ES majorly depends upon the collection of highly accurate and precise knowledge. [Tutorials' point, 2015]

Knowledge Base is collection of facts. The information is organized as data and facts about the task domain. Data, information, and past experience combined together are termed as knowledge.
The knowledge base of an ES is a store of both, factual and heuristic knowledge.

- **Factual knowledge** – is the information widely accepted by the knowledge engineers and scholars in the task domain.
- **Heuristic knowledge** – is about practice, accurate judgement, one's ability of evaluation, and guessing.

Knowledge representation is the method used to organize and formalize the knowledge in the knowledge base. It is in the form of rules, semantic networks, systems architecture, frames and ontologies semantic.

- **Primitives –** According to [BRACHMAN, 1985] semantic networks were one of the first knowledge representation primitives. Also, data structures and algorithms for general fast search. In this area there is a strong overlap with research in data structures and algorithms in computer science. In early systems the Lisp programming language which was modeled after the lambda calculus was often used as a form of functional knowledge representation. Frames and rules were the next kind of primitive. Frame languages had various mechanisms for expressing and enforcing constraints on frame data. All data in frames are stored in slots. Slots are analogous to relations in entity-relation modeling and to object properties in object-oriented modeling. Another technique for primitives is to define languages that are modeled after First Order Logic (FOL). These environments can validate logical models and can deduce new theories from existing models. Essentially they automate the process a logician would go through in analyzing a model.

- **Meta-representation –** This is also known as the issue of reflection in computer science. It refers to the capability of formalism to have access to information about its own state. An example would be the meta-object protocol that gives developers run time access to the class objects and enables them to dynamically redefine the structure of the knowledge base even at run time. Meta-representation means the knowledge representation language is itself expressed in that language. In rule-based environments the rules were also usually instances of rule classes.

- **Incompleteness** – Traditional logic requires additional axioms and constraints to deal with the real world as opposed to the world of mathematics. Also, it is often useful to associate degrees of confidence with a statement. This was one of the early innovations from expert systems research which migrated to some commercial tools, the ability to associate certainty factors with rules and conclusions. Later research in this area is known as Fuzzy Logic. [BIH, 2006]

- **Definitions and universals vs. facts and defaults**. Universals are general statements about the world such as "All humans are mortal". Facts are specific examples of universals such as "Socrates is a human and therefore mortal". In logical terms definitions and universals are about universal quantification while facts and defaults are about existential quantifications.

All forms of knowledge representation must deal with this aspect and most do so with some variant of set theory, modeling universals as sets and subsets and definitions as elements in those sets.

- **Non-monotonic reasoning** – Non-monotonic reasoning allows various kinds of hypothetical reasoning. The system associates facts asserted with the rules and facts used to justify them and as those facts change updates the dependent knowledge as well. In rule based systems this capability is known as a truth maintenance system. [ZLATARVA, 1992]

- **Expressive adequacy** – The standard that Brachman and most AI researchers use to measure expressive adequacy is usually First Order Logic (FOL). Theoretical limitations mean that a full implementation of FOL is not practical. Researchers should be clear about how expressive (how much of full FOL expressive power) they intend their representation to be. [LEVESQUE et al., 1985]

- **Reasoning efficiency**. This refers to the run time efficiency of the system. The ability of the knowledge base to be updated and the reasoner to develop new inferences in a reasonable period of time. In some ways this is the flip side of expressive adequacy. In general the more powerful a representation, the more it has expressive adequacy, the less efficient its automated reasoning engine will be. Efficiency was often an issue, especially for early applications of knowledge representation technology. They were usually implemented in interpreted environments such as Lisp which were slow compared to more traditional platforms of the time.

Knowledge Acquisition

The success of any expert system majorly depends on the quality, completeness, and accuracy of the information stored in the knowledge base. The knowledge base is formed by readings from various experts, scholars, and the knowledge engineers. The knowledge engineer is a person with the qualities of empathy, quick learning, and case analysing skills.

He acquires information from subject expert by recording, interviewing, and observing him at work, etc. He then categorizes and organizes the information in a meaningful way, in the form of IF-THEN-ELSE rules, to be used by inference machine. The knowledge engineer also monitors the development of the ES.

Knowledge acquisition is a crucial stage in the development of an expert system. As a process, it involves eliciting, analyzing, and interpreting the knowledge that human expert uses when solving a particular problem and then transforming this knowledge into a suitable machine representation.

Knowledge acquisition involves the following:

Employing a technique to elicit data (usually verbal) from the expert, interpreting these verbal data (more or less skillfully) in order to infer what might be the expert's underlying knowledge and reasoning processes, using this interpretation to guide the construction of some model or language that describes (more or less accurately) the expert's knowledge and performance. Interpretation of further data is guided in term by this evolving model. [MARCUS, 2013] [ZIMMERMANN, 2012] [KIDD, 2012]

Knowledge domains

Domains in class of application software, management and marketing do not have a stable, well-agreed language for representation or reasoning. They lack a coherent underlying theory. For example when an application engineer is required to develop a new electronic mail system, he first has to develop his own theory followed by a model of electronic mail before writing the software to realize this theory in his idiosyncratic way. As a result, such domains present the worst problems for expert system development because knowledge acquisition must involve the creation of a new language (and therefore an implicit theory) that will adequately support reasoning within the domain. This new language is likely to be the subject of continuous negotiation with both experts and users even when the expert system is developed. Surprisingly, however, this class of domain is proving increasingly popular for expert system development; presumably this is because such domains present the biggest problems to human reasoning simply because they lack an agreed language or any underlying theory.

Defining the problem that the expert system is designed to solve

The aim of an expert system is not only to capture a static representation of some knowledge domain but to simulate a particular problem-solving task carried out within that domain. A generic set of such tasks has been identified by [HAYES-ROTH, 1984] and is now widely referenced.

This list comprises the following:

Interpretation	Inferring situation description from sensor data
Prediction	Inferring likely consequences of a given situation
Diagnosis	Inferring system malfunctions from observables
Design	Configuring objects under constraints
Planning	Designing actions
Monitoring	Comparing observations to plan vulnerabilities
Debugging	Prescribing remedies for malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Diagnosing, debugging, and repairing student behavior
Control	Interpreting, prediction, repairing and monitoring system behaviors

Actions are naturally based on fundamental knowledge and not on analytical processes. The analytical process is a part of a diagnosing process but not part of a deciding activity.

Ontogenetic maps are a conceptual general problem solver (GPS) to generate solutions in the field of business considered as human adaptive systems. They describe the ontological algorithm of a function or the whole system providing the steps that have to be followed to influence business. Ontogenetic maps are basic to design business architecture to develop diagnoses, strategies, scenarios, business models, business processes and business objects. [BELOHLAVEK, 2012]

4.1.2 INFERENCE ENGINE

Use of efficient procedures and rules by the inference engine is essential in deducting a correct, flawless solution. In case of knowledge-based ES, the inference engine acquires and manipulates the knowledge from the knowledge base to arrive at a particular solution.

In case of rule based ES

- Applies rules repeatedly to the facts, which are obtained from earlier rule application (Deductive vs Inductive reasoning).
- Adds new knowledge into the knowledge base if required (monotonic systems).
- Resolves rules conflict when multiple rules are applicable to a particular case (non-monotonic systems).

Monotonic and non-monotonic systems

Traditional systems based on predicate logic are monotonic where number of statements known to be true increases with time. New statements are added and new theorems are proved, but the previously known statements never become invalid. [Artificial Intelligence Blog, 2010]

In monotonic systems there is no need to check for in consistencies between new statements and the old knowledge. When a proof is made, the basis of the proof need not be remembered, since the old statements never disappear. But monotonic systems are not good in real problem domains where the information is incomplete, situations change and new assumptions are generated while solving new problems.

Non-monotonic reasoning is based on default reasoning. Assumption is assumed to be true as long as there is no evidence to the contrary. A computational description of default reasoning must relate to the lack of information on X to conclude on Y.

Default reasoning is defined as follows:

Definition 1: If X is not known, then conclude Y.

Definition 2: If X cannot be proved, then conclude Y.

Definition 3: If X cannot be proved in some allocated amount of time then conclude Y.

It is to be noted that the above reasoning process lies outside the realm of logic. It concludes on Y if X cannot be proved, but never bothers to find whether X can be proved or not. So the default reasoning systems cannot be characterized formally. Even if one succeeds in gaining complete information at the moment, the validity of the information may not be for ever, since it is a changing world. What appears to be true now, may not be so at a later time (in a non-monotonic system).

One way to solve the problem of a changing world is to delete statements when they are no longer accurate, and replace them by more accurate statements. This leads to a non-monotonic system in which statements can be deleted as well as added to the knowledge base. When a statement is deleted, other related statements may also have to be deleted. Non-monotonic reasoning systems may be necessary due to any of the following reasons:

- The presence of incomplete information requires default reasoning.
- A changing world must be described by a changing data in database.
- Generating a complete solution to a problem may require temporary assumptions about partial solutions, validated and verified later.

Non-monotonic systems are harder to deal with than monotonic systems. This is because when a statement is deleted as "no more valid", other related statements have to be backtracked and they should be either deleted or new proofs have to be found for them. This is called dependency directed backtracking (DDB). [KAMBHAMPATI, 1996] In order to propagate the current changes into the database, the statements on which a particular proof depends, should also be stored along with the proof. Thus non–monotonic systems require more storage space as well as more processing time than monotonic systems.

Forward and backward chaining

To recommend a solution, the inference engine uses the following strategies –

- Forward Chaining
- Backward Chaining

Forward Chaining

It is a strategy of an expert system to answer the question, "What can happen next?" Here, the inference engine follows the chain of conditions and derivations and finally deduces the outcome. It considers all the facts and rules and sorts them before concluding to a solution. This strategy is followed for working on conclusion, result or effect. For example, prediction of share market status as an effect of changes in interest rates.

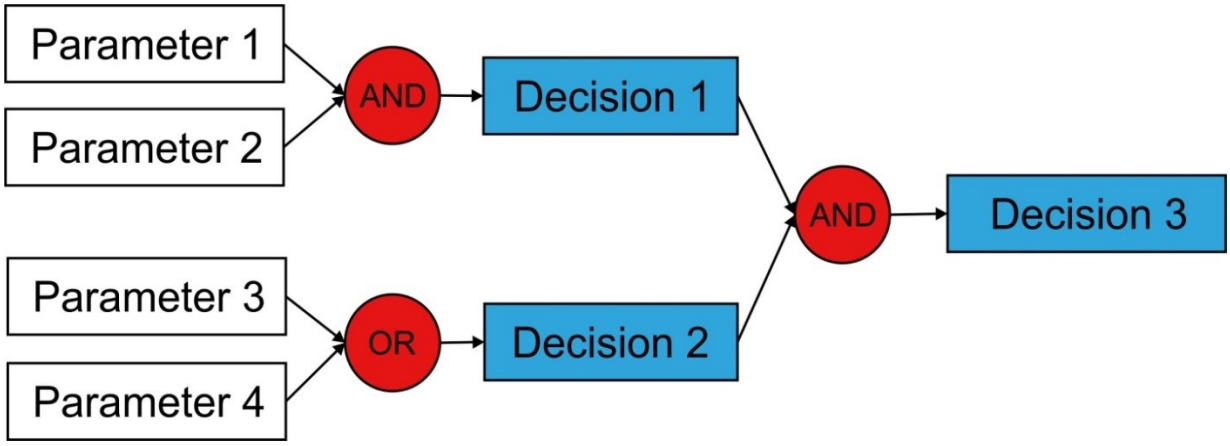

FIGURE 1 – FORWARD CHAINING PROCESS

Backward Chaining

With this strategy, an expert system finds out the answer to the question, "Why this happened?" or confirms previously determined hypothesis from acquired data. On the basis of what has already happened, the inference engine tries to find out which conditions could have happened in the past for this result. This strategy is followed for finding out cause or reason i.e. diagnosis of blood cancer in humans.

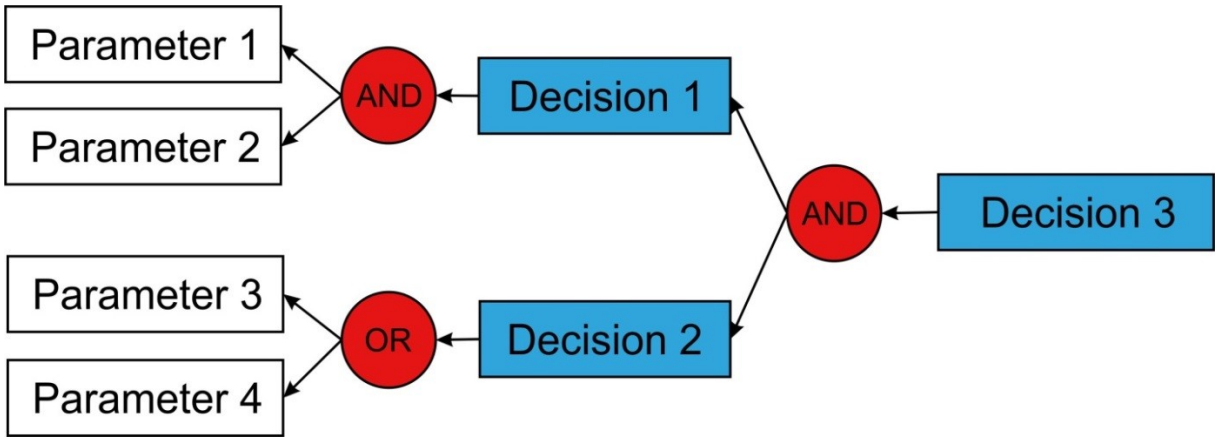

FIGURE 2 – BACKWARD CHAINING PROCESS

4.1.3 User Interface

User interface provides interaction between user of the ES and the ES itself. It is generally Natural Language Processing so as to be used by the user who is well-versed in the task domain. The user of the ES needs not be necessarily an expert in Artificial Intelligence.
It explains how the ES has arrived at a particular recommendation. The explanation may appear in the following forms:

- Natural language displayed on screen.
- Verbal narrations in natural language.
- Listing of rule numbers displayed on the screen.

The user interface makes it easy to trace the credibility of the deductions.

Requirements of Efficient ES User Interface

- It should help users to accomplish their goals in shortest possible way.
- It should be designed to work for user's existing or desired work practices.
- Its technology should be adaptable to user's requirements; not the other way round.
- It should make efficient use of user input.

4.1.4 EXPERT SYSTEM TECHNOLOGY

There are several levels of ES technologies available. Expert systems technologies include –

- **Expert System Development Environment** – The ES development environment includes hardware and tools:
  - Workstations, minicomputers, mainframes.
  - High level Symbolic Programming Languages such as **LIS**t **P**rogramming (LISP) and **PRO**grammation en **LOG**ique (PROLOG).
  - Large databases.
- **Tools** – reduce the effort and cost involved in developing an expert system to large extent.
  - Powerful editors and debugging tools with multi-windows.
  - They provide rapid prototyping.
  - Have Inbuilt definitions of model, knowledge representation and inference design.
- **Shells** – A shell is nothing but an expert system without problem specific knowledge base. A shell provides the developers with knowledge acquisition, inference engine, user interface, and explanation facility. For example, few shells are given below –
  - Java Expert System Shell (JESS) that provides fully developed Java API for creating an expert system.
  - Vidwan, a shell developed at the National Centre for Software Technology, Mumbai in 1993. It enables knowledge encoding in the form of IF-THEN rules.

4.1.5 DEVELOPMENT OF EXPERT SYSTEMS: GENERAL STEPS

The process of ES development is iterative. Steps in developing the ES include:

1. **Identify Problem Domain**
   - The problem must be suitable for an expert system to solve it.
   - Find the experts in task domain for the ES project.
   - Establish cost-effectiveness of the system.

2. **Design the System**

   - Identify the ES Technology
   - Know and establish the degree of integration with the other systems and databases.
   - Realize how the concepts can represent the domain knowledge best.

3. **Develop the Prototype**

From Knowledge Base: The knowledge engineer works to:

   - Acquire domain knowledge from the expert.
   - Represent it in the form of If-THEN-ELSE rules.

4. **Test and Refine the Prototype**

   - The knowledge engineer uses sample cases to test the prototype for any deficiencies in performance.
   - End users test the prototypes of the ES.

5. **Develop and Complete the ES**

   - Test and ensure the interaction of the ES with all elements of its environment, including end users, databases, and other information systems.
   - Document the ES project well.
   - Train the user to use ES.

6. **Maintain the ES**

   - Keep the knowledge base up-to-date by regular review and update (i.e. by machine learning).
   - Cater for new interfaces with other information systems as those systems evolve.

4.1.6 Benefits, limitations and application of Expert Systems

Benefits

- **Availability** – ES are easily available due to mass production of software.
- **Less Production Cost** – Production cost is reasonable. This makes them affordable.
- **Speed** – ES offer great speed. They reduce the amount of work an individual puts in.
- **Less Error Rate** – Error rate is low as compared to human errors.
- **Reducing Risk** – ES can work in the environment dangerous to humans.
- **Steady response** – ES work steadily without getting motional, tensed or fatigued.

Limitations

No technology can offer easy and complete solution. Large systems are costly; require significant development time and computer resources. ES have their limitations which include:

- Limitations of the technology
- Difficult knowledge acquisition
- ES are difficult to maintain
- High development costs

Applications

The following table shows where ES can be applied.

TABLE 1 – APPLICATIONS OF EXPERT SYSTEMS

Application	Description
Design Domain	Camera lens design, automobile design.
Medical Domain	Diagnosis Systems to deduce cause of disease from observed data, conduction medical operations on humans.
Monitoring Systems	Comparing data continuously with observed system or with prescribed behaviour such as leakage monitoring in long petroleum pipeline.
Process Control Systems	Controlling a physical process based on monitoring.
Knowledge Domain	Finding out faults in vehicles, computers.
Finance/Commerce	Detection of possible fraud, suspicious transactions, stock market trading, Airline scheduling, cargo scheduling.

Expert systems have played a large role in many industries including in financial services, telecommunications, healthcare, customer service, transportation, video games, manufacturing, aviation and written communication. Two early expert systems broke ground in the healthcare space for medical diagnoses: Dendral, which helped chemists identify organic molecules, and MYCIN, which helped to identify bacteria such as bacteremia and meningitis, and to recommend antibiotics and dosages. [LINDSAY et al., 1980]

4.2 PROCESSING METHODS

My analysis began by identifying the current state of diagnostic expert systems, their description and development, and continues in this chapter by research of internal processing methods. The following methods are considered for implementation into the system.

4.2.1 *RULE BASED DECISION MAKING*

Experts make better decisions than beginners. One reason for this is that they have better pattern recognition in their area of expertise than the beginner. They utilize the available information in different ways because of the experiences they have.

The idea behind rule-based decision tools is to take the knowledge available in literature and the skills that exist only in the heads of a few experts and make it available to others. Hence they are called expert systems or knowledge-based systems. This knowledge is then applied to the information in a particular situation so as to allow others to make more effective choices, and even how to make faster decisions. [HEYWOOD, 2012]

Initially, all potential decisions or outcomes that could reasonably occur in a situation are established. Then questions are set out about specific circumstances or conditions that may or may not be present. Finally, a set of rules is established with 'if this - then that' scenarios. The whole thing can then be mapped out in written form or as a decision tree.

The decision maker simply gathers information from his own experience and enters it to reach a rational and informed decision. It would also be consistent because many people putting in the same information would arrive at the same decision.

The advantages of rule-based decision tools include:

- They are very structured.
- Allow people to learn and make decisions 'like the experts'.
- No special skills required so are easy to use.
- Because it is documented, it's easy to keep records and this allows for accountability.

Difficulties include:

- Reactive: It can be boring and repetitive filling in data with little real learning occurring.
- Inflexibility: As expertise develops, user may 'outgrow' the tool.
- Limiting: Only prescribed decisions are allowed.
- They are only as useful as the overall design and the questions asked.
- They do not include the human being.

4.2.2 *MULTIPLE-CRITERIA DECISION MAKING (MCDM)*

Multiple-criteria decision-making or multiple-criteria decision analysis (MCDA) is a sub-discipline of operations research that explicitly considers multiple criteria in decision-making environments. Whether in our daily live or in professional settings, there are typically multiple conflicting criteria that need to be evaluated in making decisions. Cost or price is usually one of the main criteria. Some measure of quality is typically another criterion that is in conflict with

the cost. In purchasing a car, cost, comfort, safety, and fuel economy may be some of the main criteria we consider. It is unusual to have the cheapest car to be the most comfortable and the safest. In portfolio management, we are interested in getting high returns but at the same time reducing our risks. Again, the stocks that have the potential of bringing high returns typically also carry high risks of losing money. In a service industry, customer satisfaction and the cost of providing service are two conflicting criteria that would be useful to consider.

In our daily lives, we usually weigh multiple criteria implicitly and we may be comfortable with the consequences of such decisions that are made based on only intuition. On the other hand, when stakes are high, it is important to properly structure the problem and explicitly evaluate multiple criteria. In making the decision of whether to build a nuclear power plant or not, and where to build it, there are not only very complex issues involving multiple criteria, but there are also multiple parties who are deeply affected from the consequences.

Structuring complex problems well and considering multiple criteria explicitly lead to more informed and better decisions. There have been important advances in this field since the start of the modern multiple criteria decision making discipline in the early 1960s. A variety of approaches and methods, many implemented by specialized decision-making software have been developed. [WEISTROFFER et al., 2005] [MCGINLEY, 2012]

There are different classifications of MCDM problems and methods. A major distinction between MCDM problems is based on whether the solutions are explicitly or implicitly defined.

- *Multiple-criteria evaluation problems* – These problems consist of a finite number of alternatives, explicitly known in the beginning of the solution process. Each alternative is represented by its performance in multiple criteria. The problem may be defined as finding the best alternative for a decision maker (DM), or finding a set of good alternatives. One may also be interested in "sorting" or "classifying" alternatives. Sorting refers to placing alternatives in a set of preference-ordered classes (such as assigning credit-ratings to countries), and classifying refers to assigning alternatives to non-ordered sets (such as diagnosing patients based on their symptoms). Some of the MCDM methods in this category have been studied in a comparative manner in [TRIANTAPHYLLOU, 2000].
- *Multiple-criteria design problems (multiple objective mathematical programming problems)* – In these problems, the alternatives are not explicitly known. A solution can be found by solving a mathematical model. The number of alternatives is either infinite and not countable (when some variables are continuous) or typically very large if countable (when all variables are discrete).

Whether it is an evaluation problem or a design problem, preference information of DMs is required in order to differentiate between solutions. The solution methods for MCDM

problems are commonly classified based on the timing of preference information obtained from the DM.

There are methods that require the DM's preference information at the start of the process, transforming the problem into essentially a single criterion problem. These methods are said to operate by "prior articulation of preferences." Methods based on estimating a value function or using the concept of "outranking relations," analytical hierarchy process, and some decision rule-based methods try to solve multiple criteria evaluation problems utilizing prior articulation of preferences. Similarly, there are methods developed to solve multiple criteria design problems using prior articulation of preferences by constructing a value function. Perhaps the most well-known of these methods is goal programming. Once the value function is constructed, the resulting single objective mathematical program is solved to obtain a preferred solution.

Some methods require preference information from the DM throughout the solution process. These are referred to as interactive methods or methods that require "progressive articulation of preferences." These methods have been well-developed for both the multiple criteria evaluation and design problems. [GEOFFRION et al., 1972]

Multiple criteria design problems typically require the solution of a series of mathematical programming models in order to reveal implicitly defined solutions. For these problems, a representation or approximation of "efficient solutions" may also be of interest. This category is referred to as "posterior articulation of preferences," implying that the DM's involvement starts posterior to the explicit revelation of "interesting" solutions.

When the mathematical programming models contain integer variables, the design problems become harder to solve. Multi-Objective Combinatorial Optimization (MOCO) constitutes a special category of such problems posing substantial computational difficulty. [EHRGOTT et al., 2003]

Simple Additive Weighing

In decision theory, the simple additive weighing (SAW) or weighted sum model (WSM) is the best known and simplest multi-criteria decision making (MCDM) method for evaluating a number of alternatives in terms of a number of decision criteria. It is very important to state here that it is applicable only when all the data are expressed in exactly the same unit. If this is not the case, then the final result is equivalent to "adding apples and oranges."

In general, I suppose that a given MCDM problem is defined on m alternatives and n decision criteria. Furthermore, let me assume that all the criteria are benefit criteria, that is, the higher the values are, the better it is. Next suppose that w_j denotes the relative weight of importance of the criterion C_j and a_{ij} is the performance value of alternative A_i when it is

evaluated in terms of criterion C_j. Then, the total (i.e. when all the criteria are considered simultaneously) importance of alternative A_i, denoted as $A_i^{SAW-score}$, is defined as follows:

$$A_i^{SAW-score} = \sum_{j=1}^{n} w_j a_{ij}, for\ i = 1, 2, 3, \dots, m. \qquad (1.1)$$

For the maximization case, the best alternative is the one that yields the maximum total performance value. [TRIANTAPHYLLOU, 2000]

Evidential reasoning approach

In decision theory, the evidential reasoning (ER) approach is a generic evidence-based multi-criteria decision analysis (MCDA) approach for dealing with problems having both quantitative and qualitative criteria under various uncertainties including ignorance and randomness. It has been used to support various decision analysis, assessment and evaluation activities such as environmental impact assessment [WANG et al., 2006] and organizational self-assessment [SIOW et al., 2001] based on a range of quality models.

The evidential reasoning approach is recently developed on the basis of decision theory in particular utility theory, artificial intelligence in particular the theory of evidence, statistical analysis and computer technology. It uses a belief structure to model an assessment with uncertainty, a belief decision matrix to represent an MCDA problem under uncertainty, evidential reasoning algorithms to aggregate criteria for generating distributed assessments, and the concepts of the belief and plausibility functions to generate a utility interval for measuring the degree of ignorance. A conventional decision matrix used for modeling an MCDA problem is a special case of a belief decision matrix.

Similar to decision matrix, a belief decision matrix is used to describe a multiple criteria decision analysis (MCDA) problem in the Evidential Reasoning Approach. If in an MCDA problem, there are *m* alternative options and each need to be assessed on *n* criteria, then the belief decision matrix for the problem has *m* rows and *n* columns or *m* x *n* elements, as shown in the following table. Instead of being a single numerical value or a single grade as in a decision matrix, each element in a belief decision matrix is a belief structure. A conventional decision matrix is a special case of belief decision matrix when only one belief degree in a belief structure is 1 and the others are 0.

TABLE 2 - BELIEF DECISION MATRIX

	Criterion 1	…	Criterion n
Alternative 1			
…		x_{ij}={ (Decision 1, 0.6), (Decision 2, 0.4)}	
Alternative m			

The use of belief decision matrices for MCDA problem modelling in the ER approach results in the following features:

- An assessment of an option can be more reliably and realistically represented by a belief decision matrix than by a conventional decision matrix.
- It accepts data of different formats with various types of uncertainties as inputs, such as single numerical values, probability distribution, and subjective judgments with belief degrees.
- It allows all available information embedded in different data formats, including qualitative and incomplete data, to be maximally incorporated in assessment and decision making processes.
- It allows assessment outcomes to be represented more informatively.

4.2.3 GENETIC ALGORITHM

In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

To evolutionary-computation researchers, the mechanisms of evolution seem well suited for some of the most pressing computational problems in many fields. Many computational problems require searching through a huge number of possibilities for solutions. In biology the enormous set of possibilities is the set of possible genetic sequences, and the desired "solutions" are highly fit organisms – organisms well able to survive and reproduce in their environments. Evolution can also be seen as a method for designing innovative solutions to complex problems. The fitness criteria continually change as creatures evolve, so evolution is searching a constantly changing set of possibilities. Searching for solutions in the face of changing conditions is precisely what is required for adaptive computer programs. Furthermore, evolution is a massively parallel search method – rather than work on one species at a time, evolution tests and changes millions of species in parallel. Finally, viewed from a high level the "rules" of evolution are remarkably simple – species evolve by means of random variation (via mutation, recombination, and other operators), followed by natural selection in which the fittest tend to survive and reproduce, thus propagating their genetic material to future generations. [MITCHELL, 1999]

Genetic algorithms find application in bioinformatics, computational science, engineering, economics, manufacturing, mathematics, physics, and other fields.

4.2.4 SEMANTIC WEB APPROACH

This approach comes from raw binary and octal numeric machine languages, through various symbolic assemblies, scientific, business and higher level languages. Programming languages have increasingly adopted notations that are more natural and meaningful to a human user. The important characteristic of this trend is the elevation of the level at which instructions are specified from the low level details of the machine operations to high level descriptions of the task to be done leaving out details that can be filled in by the computer. The ideal product of such continued evolution would be a system in which the user specifies what he wants done in a language that is so natural that negligible mental effort is required to recast the specification from the form in which he formulates it to that which the machine requires. [WOODS, 1978]

The logical choice for WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet's structure makes it a useful tool for computational linguistics and natural language processing. [MILLER, 1995]

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms - strings of letters - but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus do not follow any explicit pattern other than meaning similarity. [FELLBAUM, 2010]

4.3 OBJECT ORIENTED PROGRAMMING (OOP)

Object-oriented programming (OOP) is based on the concept of objects, which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods. A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated. In OOP, computer programs are designed by making them out of objects that interact with one another. There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which typically also determine their type.

Many of the most widely used programming languages are multi-paradigm programming languages that support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include Java, C++, C#, Python, PHP, Ruby, Perl, Delphi, Objective-C, Swift, Common Lisp, and

Smalltalk. The most used modern languages are Java, C#, and PHP, when simple applications on LINUX server are developed.

4 major principles that make a language object-oriented are encapsulation, data abstraction, polymorphism and inheritance. Any .NET based language is not preferred over another as they all are based on the Common Language Specification (CLS) and all are compiled into the same assembly language. It is about personal preference that determines what language is chosen. Of course there are other OOP languages such as Ruby, a pure OOP language, and hybrid languages such as Python, C++ and Java. [Four major principles of Object-Oriented Programming , 2012]

4.3.1 ENCAPSULATION

Encapsulation hides data implementation by restricting access to accessors and mutators. At first we need to define accessors and mutators.

- **Accessor –** is a method that is used to ask an object about itself, in OOP usually in the form of properties. Properties normally have a get method, which is an accessor method. Accessor methods are not only restricted to properties and can be any public method that gives information about the state of the object.
- **Mutator –** is a public method used to modify the state of an object, while hiding the implementation of exactly how the data are modified. Mutators are properties as the accessors, except this time it is the set method that lets the caller modify the member data behind the scenes. The use of mutators and accessors provides many advantages. By hiding the implementation of our class, we can make changes to the class without the worry that we are going to break other code that is using and calling the class for information. If we wanted, we could change type of one parameter of the class from a String to an array of single characters, but the callers would never have to know because we would still get them a single string, but behind the scenes we are dealing with a character array instead of a string object. It is transparent to the rest of the program. This type of data protection and implementation protection is called Encapsulation. Accessors and mutators as the pieces that surround the data form the class.

4.3.2 ABSTRACTION

Data abstraction is the simplest principle to understand. Data abstraction and encapsulation are closely tied together as a simple definition of data abstraction is the development of classes, objects, types in terms of their interfaces and functionality, instead of their implementation details. Abstraction denotes a model, a view, or some other focused

representation for an actual item. It is the development of a software object to represent an object in the real world. Encapsulation hides the details of that implementation.

Abstraction is used to manage complexity. Software developers use abstraction to decompose complex systems into smaller components. As development progresses, programmers know the functionality they can expect from as yet undeveloped subsystems. But programmers are not burdened by considering the ways in which the implementation of later subsystems will affect the design of earlier development.

The best definition of abstraction is: "An abstraction denotes the essential characteristics of an object that distinguish it from all other kinds of object and thus provide crisply defined conceptual boundaries, relative to the perspective of the viewer." [BOOCH, 1991]

We can apply this code for a diagnostic object. What are some attributes that a diagnostic can have? Those attributes must be represented here in our software model of a diagnostic. Attributes such as ID, date of creation, description; we need to be able to see those. We need the ability for the diagnostic to do things, such as to be run. We need to be able to ask the diagnostic if it can be read.

There we have started to create a software model of a diagnostic object; we have created an abstract type of what a diagnostic object is to us outside of the software world. The abstract diagnostic is defined by the operations that can be performed on it, and the information we can get from it and give to it. What does the abstracted diagnostic object look like to the software world that doesn't have access to its inner workings? It looks like this:

You can't really see what the code is that makes the diagnostic be run. This is encapsulation that we discussed.

So, in short, data abstraction is nothing more than the implementation of an object that contains the same essential properties and actions we can find in the original object we are representing.

4.3.3 INHERITANCE

Objects can relate to each other with either a "has a", "uses a" or an "is a" relationship. "Is a" is the inheritance way of object relationship. The example of this is a library so we use a library for example. A library lends more than just books, it also lends newspapers, magazines and e-books. All of these items can be treated the same on some level - all three types represent assets of the library that can be loaned out to people. Even though the 3 types can be viewed as the same, they are not identical. A book has an ISBN and a magazine does not. And e-book is intangible and has a file format.

Each of these library's assets should be represented by its own class definition. Without inheritance though, each class must independently implement the characteristics that are

common to all loanable assets. All assets are either checked out or available for checkout. All assets have a title, a date of acquisition and a replacement cost. Rather than duplicate functionality, inheritance allows you to inherit functionality from another class called a superclass or base class.

Now, "myBook" class I create automatically inherited these methods, and we didn't even have to touch the Book class in order for it to happen. The book and e-book classes above automatically inherited the abilities to be checked out and checked in. In our "myBook" above, now we can check the book in by calling "myBook.CheckIn()". One of the most powerful features of inheritance is the ability to extend components without any knowledge of the way in which a class was implemented. Declaration options, such as Public and Private, dictate which members of a superclass can be inherited.

4.3.4 POLYMORPHISM

Polymorphism means one name, many forms. Polymorphism manifests itself by having multiple methods all with the same name, but slightly different functionality. Many programmers in Visual Basic 6 are familiar with interface polymorphism. Polymorphism from the point of view of inheritance is the part that is new to many people as it can be difficult to fully describe the full potential of polymorphism until we get some practice with it and see exactly what happens under different scenarios. We're only going to talk about polymorphism, like the other topics, at the basic level.

There are 2 basic types of polymorphism. Overriding is also called run-time polymorphism and overloading is called compile-time polymorphism. This difference is for method overloading, the compiler determines which method will be executed and this decision is made when the code is compiled. Which method will be used for method overriding is determined at runtime based on the dynamic type of an object.

4.4 PROGRAMMING ENVIRONMENTS

As we need to choose one programming environment for final application design, we have to analyze current available languages and programming tools. Modern applications are developed as OOP – using object oriented programming – using objects and classes in OOP way observing the rules of OOP. The most important requirement was to build a system architecture based on LINUX operating system, so the simplest way was to choose Apache, PHP and My SQL.

4.4.1 .NET

.NET Framework (pronounced dot net) is a software framework developed by Microsoft and introduced a fresh operating modality and perspective on computing software and devices.

.NET is a multilanguage and multiplatform operating environment. .NET offers C#, Visual Basic .NET, and many more .NET-compliant languages. Programming in .NET does not require learning an entirely new language. The common language runtime is the common runtime of all .NET languages. In addition, Microsoft publishes the Common Language Infrastructure (CLI) document, which is a set of guidelines for creating a .NET common language runtime for any platform, such as Linux. [Codeguru, 2004]

.NET marks Microsoft's first extensive support of open standards. This means that any device that supports these standards can actively participate in a .NET conversation. This will liberate personal digital assistants (PDAs), smartphones, and embedded devices.

Microsoft .NET is web empowered. Developers can easily create feature-rich web applications. This represents the front, middle, and bottom tier of an n-tiered enterprise application. Windows Forms, a forms generation engine, and other additions in the .NET Framework made development of traditional Windows applications more intuitive, while adding additional features.

In .NET, developers use standard language syntax to create, publish, and export components. There is nothing else to learn. In the server application, you define a public class, using the syntax of the preferred .NET language. In the client, you import a reference to the component application and then create an instance of the component class. That is it. You can then use the component. Using open standards, .NET components are potentially accessible to everyone at any time.

The benefit of .NET to users is the new generation of software that .NET introduced: software that runs anywhere, at any time, on any platform, and from devices large and small.

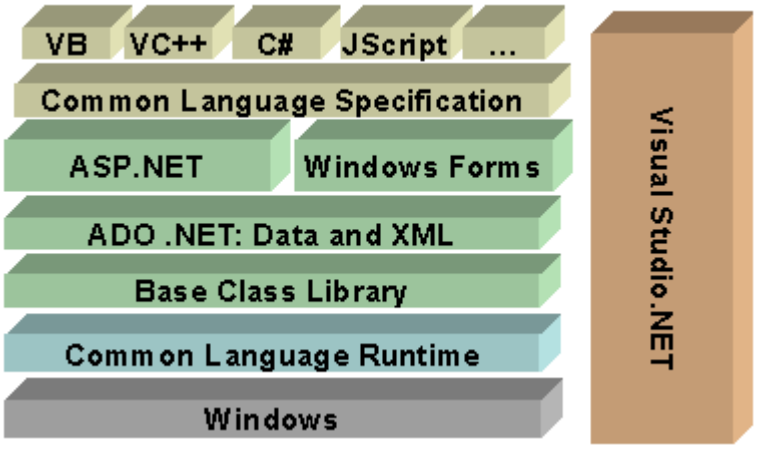

FIGURE 3 - .NET HIERARCHICAL STRUCTURE [MSDN.MICROSOFT.COM,2016]

.NET is tiered, modular, and hierarchal. Each tier of the .NET Framework is a layer of abstraction. .NET languages are the top tier and the most abstracted level. The common language runtime is the bottom tier, the least abstracted, and closest to the native environment. This is

important since the common language runtime works closely with the operating environment to manage .NET applications. The .NET Framework is partitioned into modules, each with its own distinct responsibility. Finally, since higher tiers request services only from the lower tiers, .NET is hierarchal.

.NET Framework is a managed environment. The common language runtime monitors the execution of .NET applications and provides essential services. It manages memory, handles exceptions, ensures that applications are well-behaved, and much more.

Language interoperability is one goal of .NET. Its languages share a common runtime (the common language runtime, a common class library), the Framework Class Library (FCL), a common component model, and common types. In .NET, the programming language is a lifestyle choice.

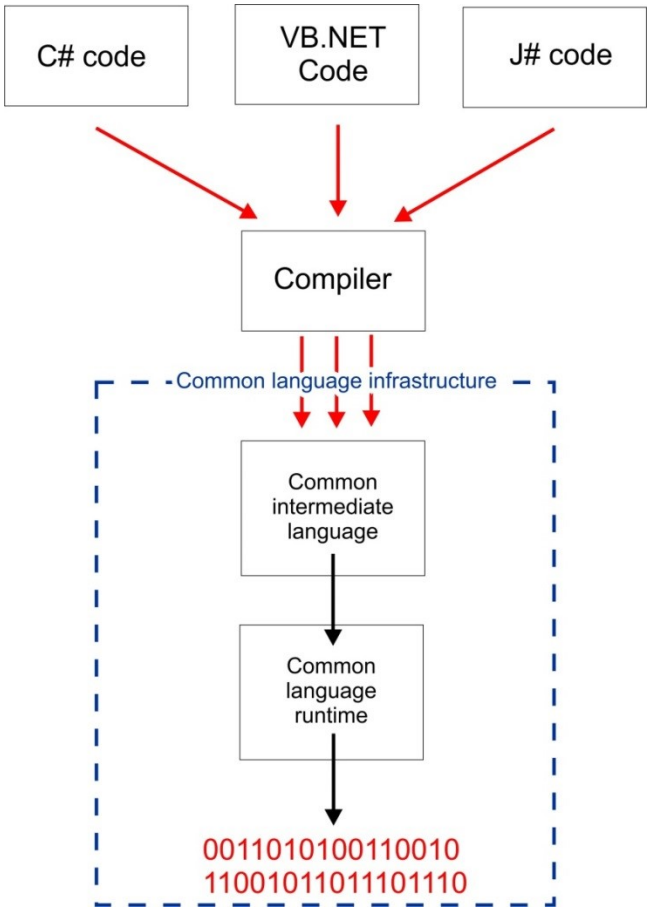

FIGURE 4 - .NET COMPILATION PROCESS

.NET supports managed and unmanaged programming languages. Applications created from managed languages, such as C# and VB.NET, execute under the management of a common runtime, called the common language runtime. Managed applications compile to Microsoft Intermediate Language (MSIL) and metadata. MSIL is a low-level language that all managed languages compile to instead of native binary. Using just-in-time compilation, at code execution,

MSIL is converted into binary optimized both to the environment and the hardware. Since all managed languages ultimately become MSIL, there is a high degree of language interoperability in .NET (see Figure 4).

Common Language Specification (CLS) is a set of specifications or guidelines defining a .NET language. Shared specifications promote language interoperability. For example, CLS defines the common types of managed languages, which is a subset of the Common Type System (CTS). This removes the issue of marshaling, a major impediment when working between two languages.

The .NET Framework Class Library (FCL) is a set of managed classes that provide access to system services. File input/output, sockets, database access, remoting, and XML are just some of the services available in the FCL. Importantly, all the .NET languages rely on the same managed classes for the same services. This is one of the reasons that, once you have learned any .NET language, you have learned 40 percent of every other managed language. The same classes, methods, parameters, and types are used for system services regardless of the language. This is one of the most important contributions of FCL.

4.4.2 PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor. [PHP, 2016]

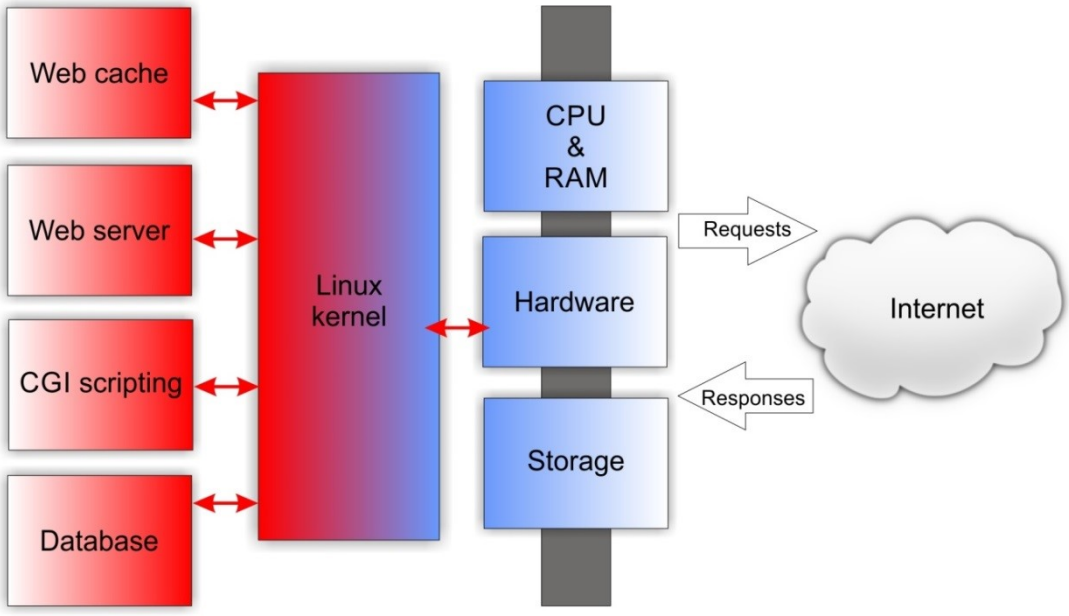

FIGURE 5- OVERVIEW OF THE LAMP SOFTWARE BUNDLE WITH WEB CACHE (SQUID).

PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a standard. Since 2014 work has gone on to create a formal PHP specification. [JACKSON, 2014]

4.4.3 JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use particularly for client-server web applications with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. [GOSLING et al., 2014]

The original and reference implementation Java compilers, virtual machines and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

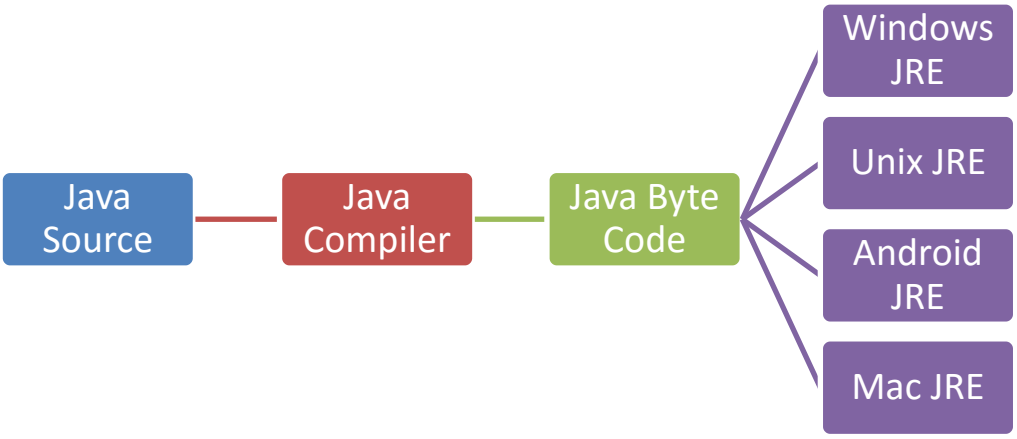

FIGURE 6 - "WRITE ONCE, RUN ANYWHERE" (WORA) PRINCIPLE

The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

4.4.4 OTHER

Delphi

Delphi is a programming language and software development kit (SDK) for desktop, mobile, web, and console applications. Delphi's compilers use their own Object Pascal dialect of Pascal and generate native code for several platforms: Windows, OS X (32-bit only), iOS and Android. Delphi, part of RAD Studio includes a code editor with Code Insight (code completion), Error Insight (real-time error-checking), and other features, refactoring a visual forms designer for both VCL (native Windows) and FMX (cross-platform, partially native per platform); an integrated debugger for all platforms including mobile, source control and support for third-party plugins. It has strong database support. Delphi is remarkably fast to compile unlike any other common languages, including C# and Swift, it is not unusual for a Delphi project of a million lines to compile in a few seconds. It is under active development, with releases every six months, with new platforms being added approximately every second release. [BUCHANAN, 2003]

Delphi was originally developed by Borland as a rapid application development tool for Windows as the successor of Turbo Pascal. Delphi added full object-orientation to the existing language and since then the language has grown and supports many other modern language features including generics and anonymous methods as well as unusual features such as inbuilt string types and native COM support. In 2007, the products were released jointly as RAD Studio. RAD Studio is a shared host for Delphi and C++Builder and can be purchased with either or both.

Python

Python is a widely used high-level programming language used for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. [KUHLMAN, 2009]

Python features a dynamic type system, automatic memory management and supports multiple programming paradigms including object-oriented, imperative, functional programming and procedural styles. It has a large and comprehensive standard library.

Python is widely used and interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Perl

Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages (Perl 5 and Perl 6). Perl is not officially an acronym, but there are various backronyms in use, the best-known is "Practical Extraction and Reporting Language". Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier. Since then, it has undergone many changes and revisions. Perl 6, which began as a redesign of Perl 5 in 2000, eventually evolved into a separate language. Both languages continue to be developed independently by different development teams and liberally borrow ideas from one another.

The Perl languages borrow features from other programming languages including C, shell script or AWK. They provide powerful text processing facilities without the arbitrary data-length limits of many contemporary Unix command line tools, facilitating easy manipulation of text files. Perl 5 gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its unsurpassed regular expression and string parsing abilities. [About Perl, 2016]

Perl 5 is used for graphics programming, system administration, network programming, finance, bioinformatics, and other applications. It has been nicknamed "the Swiss Army chainsaw of scripting languages" because of its flexibility and power and possibly also because of its "ugliness". In 1998, it was also referred to as the "duct tape that holds the Internet together", in reference to both its ubiquitous use as a glue language and its perceived inelegance.

4.5 DATABASE SYSTEMS

In this chapter we describe most frequent database systems and compare them to get the most suitable one for diagnostic expert system. All of them are relational databases using SQL (Structured Query Language) as the language for querying and maintaining the database.

4.5.1 MY SQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality. [MySQL 5.7 Reference Manual, 2016]

MySQL is a central component of the LAMP open-source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python" see Figure 5. Applications that use the MySQL database include Joomla, WordPress, phpBB and Drupal etc. MySQL is also used in many high-profile, large-scale websites, including Google, Facebook, Twitter, Flickr, and YouTube.

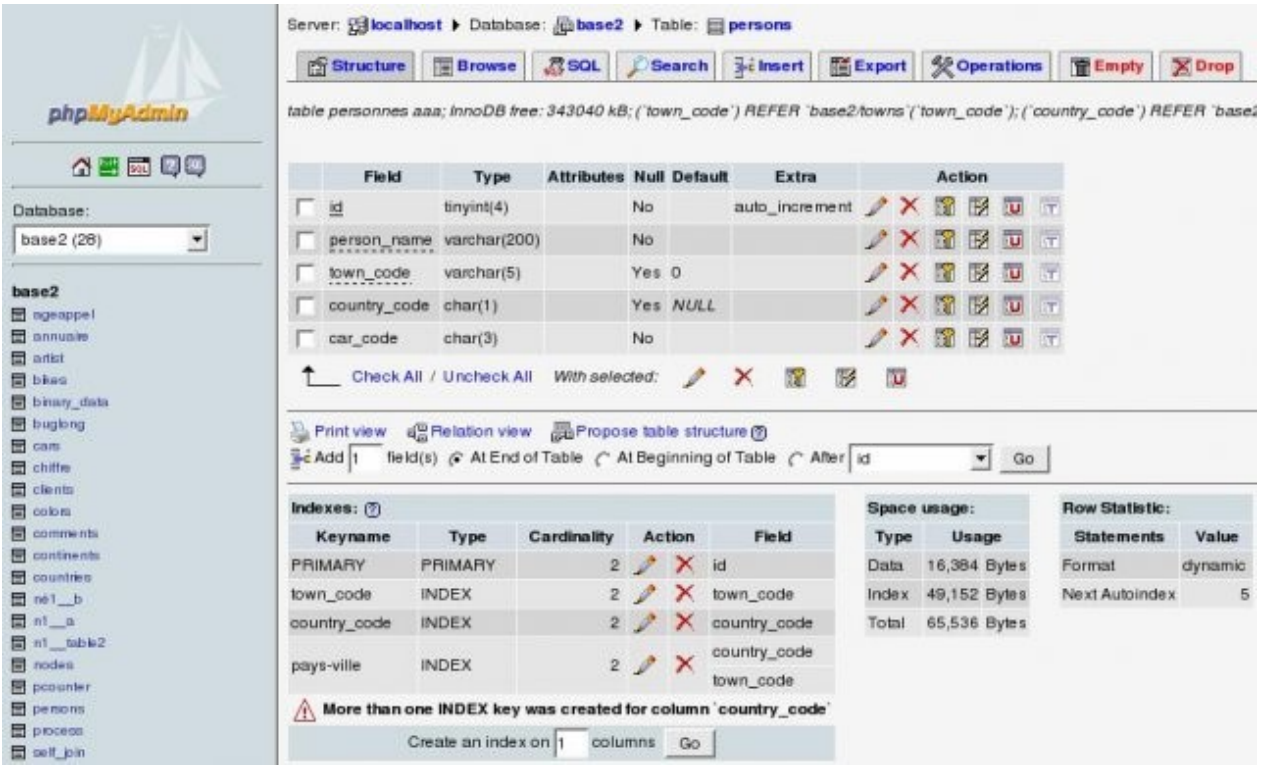

FIGURE 7 – PHPMYADMIN – A TOOL FOR MANAGING MYSQL DATABASES

MySQL is written in C and C++. Its SQL parser is written in "Yet Another Compiler-Compiler" (yacc), but it uses a home-brewed lexical analyzer. MySQL works on many system platforms, including Linux, MacOS, Microsoft Windows, Novell NetWare, OpenBSD, Symbian and SunOS etc.

The MySQL server software itself and the client libraries use dual-licensing distribution. They are offered under GPL version 2, beginning from 2000 or to use a proprietary license.

Support can be obtained from the official manual. Free support additionally is available in different IRC channels and forums. Oracle offers paid support via its MySQL Enterprise products. They differ in the scope of services and in price. Additionally, a number of third party organizations exist to provide support and services.

MySQL has received positive reviews, and reviewers noticed it performs extremely well in the average case and that the developer interfaces are there, and the documentation is very good. It has also been tested to be a fast, stable and true multi-user, multi-threaded SQL database server. [LinuxMint.com, 2016]

4.5.2 POSTGRESQL

PostgreSQL, often simply Postgres, is an object-relational database (ORDBMS) – a RDBMS with additional OOP features with an emphasis on extensibility and standards compliance. As a database server, its primary function is to store data securely and to allow for retrieval at the request of other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. On MacOS Server, PostgreSQL is the default database and it is also available for Microsoft Windows and Linux. [PostgreSQL Documentation, 2016]

PostgreSQL is Atomicity, Consistency, Isolation, Durability (ACID)-compliant and transactional. PostgreSQL has updatable views and materialized views, triggers, foreign keys; supports functions, stored procedures and other expandability.

PostgreSQL is developed by the PostgreSQL Global Development Group, a diverse group of many companies and individual contributors. It is free and open-source software, released under the terms of the PostgreSQL License, a permissive free-software license.

4.5.3 ORACLE

Oracle database (Oracle DB) is a relational database management system (RDBMS) from the Oracle Corporation. Originally developed in 1977 by Lawrence Ellison and other developers, Oracle DB is one of the most trusted and widely-used relational database engines.

The system is built around a relational database framework in which data objects may be directly accessed by users (or an application front end) through structured query language (SQL). Oracle is a fully scalable relational database architecture and is often used by global enterprises, which manage and process data across wide and local area networks. The Oracle database has its own network component to allow communications across networks. Oracle DB is also known as Oracle RDBMS and, sometimes, just Oracle. [Techopedia, 2016]

Oracle DB rivals Microsoft's SQL Server in the enterprise database market. There are other database offerings, but most of these command a tiny market share compared to Oracle DB and SQL Server. Fortunately, the structures of Oracle DB and SQL Server are quite similar, which is a benefit when learning database administration.

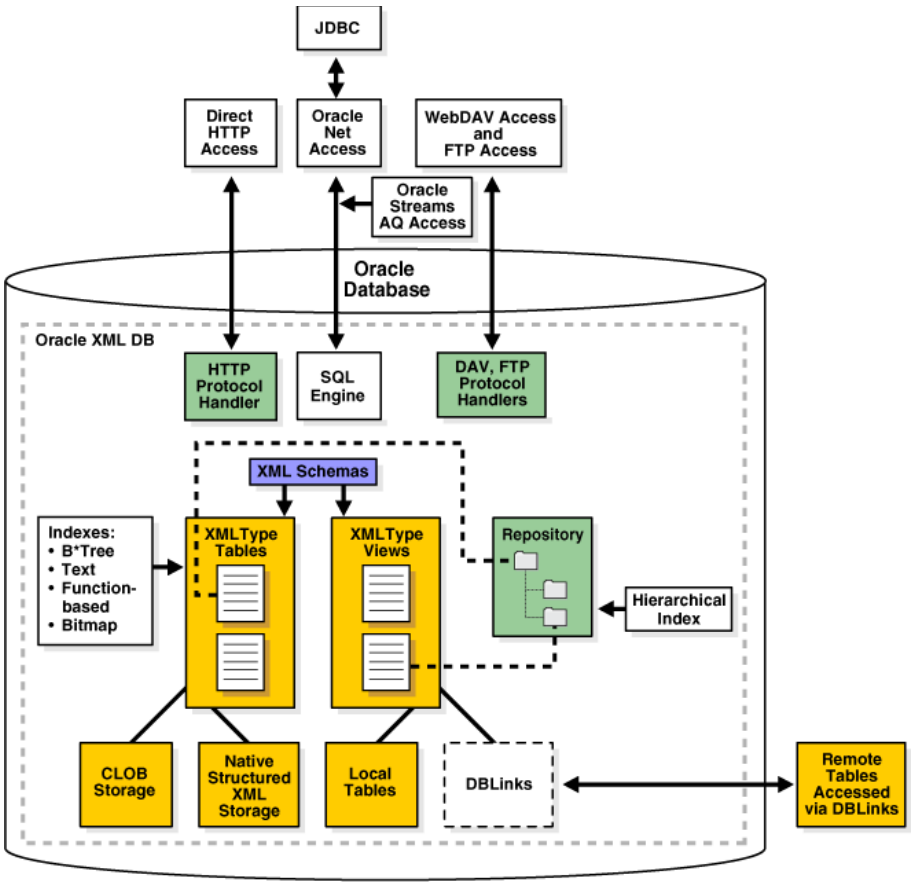

FIGURE 8 – ORACLE DATABASE SCHEMA [ORACLE HELP CENTER, 2016]

Oracle DB runs on most major platforms, including Windows, UNIX, Linux and Mac OS. Different software versions are available, based on requirements and budget. Oracle DB editions are hierarchically broken down as follows:

- Enterprise Edition: Offers all features, including superior performance and security, and is the most robust

- Standard Edition: Contains base functionality for users that do not require Enterprise Edition's robust package
- Express Edition (XE): The lightweight, free and limited Windows and Linux edition
- Oracle Lite: For mobile devices

A key feature of Oracle is that its architecture is split between the logical and the physical part. This structure means that for large-scale distributed computing also known as grid computing, the data location is irrelevant and transparent to the user, allowing for a more modular physical structure that can be added to and altered without affecting the activity of the database, its data or users. The sharing of resources in this way allows for very flexible data networks whose capacity can be adjusted up or down to suit demand, without degradation of service. It also allows for a robust system to be devised as there is no single point at which a failure can bring down the database, as the networked schema of the storage resources means that any failure would be local only.

4.5.4 MS SQL

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications - which may run either on the same computer or on another computer across a network including internet.

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

- **Enterprise**

  SQL Server Enterprise Edition includes both the core database engine and add-on services, with a range of tools for creating and managing a SQL Server cluster. It can manage databases as large as 524 petabytes and address 12 terabytes of memory and supports 640 logical processors (CPU cores). [Microsoft, 2016]

- **Standard**

  SQL Server Standard edition includes the core database engine, along with the stand-alone services. It differs from Enterprise edition in that it supports fewer active instances (number of nodes in a cluster) and does not include some high-availability functions such as hot-add memory (allowing memory to be added while the server is still running), and parallel indexes.

- **Web**
  SQL Server Web Edition is a low-TCO (Total Cost of Ownership) option for Web hosting.

- **Business Intelligence**
  Introduced in SQL Server 2012 and focusing on Self Service and Corporate Business Intelligence. It includes the Standard Edition capabilities and Business Intelligence tools.

- **Workgroup**
  SQL Server Workgroup Edition includes the core database functionality but does not include the additional services. Note that this edition has been retired in SQL Server 2012.

- **Express**
  SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 10 GB database files. The first is SQL Server Express with Tools, which includes SQL Server Management Studio Basic. SQL Server Express with Advanced Services adds full-text search capability and reporting services.

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol used to transfer data between a database server and a client. TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, named pipes, and shared memory. Consequently, access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over web services. [DELANEY, 2006]

Data storage is a database, which is a collection of tables with typed columns. SQL Server supports different data types, including primary types such as Integer, Float, Decimal, Char (including character strings), Varchar (variable length character strings), binary (for unstructured blobs of data), Text (for textual data) among others. The rounding of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down depending on arguments.

Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views). In addition to tables, a database can also contain other objects including views, stored procedures, indexes and constraints, along with a transaction log. A SQL Server database can contain a maximum of 231 objects, and can span multiple OS-level files with a maximum file size of 260 bytes.

The main mode of retrieving data from a SQL Server database is querying for it. The query is expressed using a variant of SQL called T-SQL, a dialect Microsoft SQL Server shares with Sybase SQL Server due to its legacy. The query declaratively specifies what is to be retrieved. It is processed by the query processor, which figures out the sequence of steps that will be necessary to retrieve the requested data. The sequence of actions necessary to execute a query is called a query plan. There might be multiple ways to process the same query.

SQL Server includes a cost-based query optimizer which tries to optimize on the cost, in terms of the resources it will take to execute the query. Given a query, then the query optimizer looks at the database schema, the database statistics and the system load at that time. It then decides which sequence to access the tables referred in the query, which sequence to execute the operations and what access method to be used to access the tables.

SQL Server also allows stored procedures to be defined. Stored procedures are parameterized T-SQL queries that are stored in the server itself. Stored procedures can accept values sent by the client as input parameters and send back results as output parameters. They can call defined functions and other stored procedures including the same stored procedure. They can be selectively provided access to. Unlike other queries, stored procedures have an associated name which is used at runtime to resolve into the actual queries. Also because the code need not be sent from the client every time, it reduces network traffic and somewhat improves performance. Execution plans for stored procedures are also cached as necessary.

4.5.5 DATABASE COMPARISON

In the section above I selected 4 most used SQL database systems I worked with during my working experience. I wanted to focus on commercial (proprietary) as well as free (or open source) products. In FutureSME project I started to work with MySQL and later I translated the application for using MS SQL. Differences between these two are mentioned in development part of this work (see Chapter 7.2). We can see that there are no restrictions between SQL databases when we focus on operating system as all of them are supported by Linux as well as Windows operating systems.

TABLE 3 – SQL DATABASES - OPERATING SYSTEM SUPPORT AND LICENSING

	Windows	Linux	Licence
Microsoft SQL Server	Yes	Yes	Proprietary
MySQL	Yes	Yes	GPL v2 or Proprietary
Oracle	Yes	Yes	Proprietary
PostgreSQL	Yes	Yes	PostgreSQL Licence (a liberal Open Source license)

Other parameters have no restrictions as well so finally we are evaluating database system by its price (licensing) and best-practice in particular use. In this project we were restricted by other software – content management system. As at the beginning of the project we

had agreed on Drupal that was the reason for using PHP and MySQL running on Linux. However, during the project development the requirements for implementation changed, and the decision was made to switch the content management system to Sitefinity and MS SQL running on Windows server. So I was forced to translate it into a new system architecture.

TABLE 4 – SQL DATABASES – OTHER PARAMETERS

Database	ACID	Referential integrity	Transactions	Fine-grained locking	Multiversion concurrency control	Unicode	Interface
Microsoft SQL Server	Yes	Yes	Yes	Yes*	Yes	Yes	GUI, SQL
MySQL	Yes	Yes	Yes except for DDL	Yes*	Yes	Yes	GUI, SQL
Oracle	Yes	Yes	Yes except for DDL	Yes*	Yes	Yes	API, GUI, SQL
PostgreSQL	Yes	Yes	Yes	Yes*	Yes	Yes	API, GUI, SQL

* Row-level locking

Both MS SQL and Sitefinity are commercial software products. As I wanted to have a diagnostic tool that would support both databases (MySQL and MS SQL) I created two classes for communicating with database and it is only up to installer what database is chosen and according to parameters settings, particular methods are used. There are two quite different tools for maintain the database. MySQL has well known PHPMyAdmin, which is web-based PHP tool. MS SQL has SQL Server management studio – windows desktop application, which is in my opinion much more powerful in comparing with PHPMyAdmin, has more tools, supports more plug-in modules and working with this system is significantly faster.

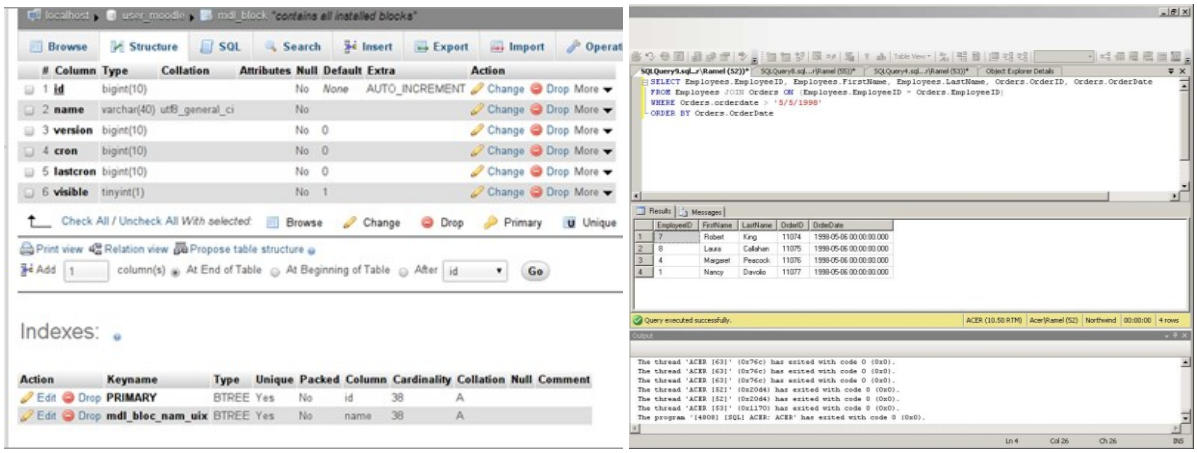

FIGURE 9 – PHPMYADMIN VS SQL MANAGEMENT STUDIO

TABLE 5 – POSTGRESQL MAIN PARAMETERS [DIGITALOCEAN, 2014]

ADVANTAGES	WHY USE
An open-source SQL standard compliant RDBMS PostgreSQL is open-source and free, yet a very powerful relational database management system. **Strong community** PostgreSQL is supported by a devoted and experienced community which can be accessed through knowledge-bases and Q&A sites 24/7 for free. **Strong third-party support** Regardless of the extremely advanced features, PostgreSQL is adorned with many great and open-source third-party tools for designing, managing and using the management system. **Extensible** It is possible to extend PostgreSQL programmatically with stored procedures, like an advanced RDBMS should be. **Objective** PostgreSQL is not just a relational database management system but an objective one - with support for nesting, and more.	**Data integrity** When reliability and data integrity are an absolute necessity without excuses, PostgreSQL is the better choice. **Complex, custom procedures** If you require your database to perform custom procedures, PostgreSQL, being extensible, is the better choice. **Integration** In the future, if there is a chance of necessity arising for migrating the entire database system to a propriety (e.g. Oracle) solution, PostgreSQL will be the most compliant and easy to handle base for the switch**.** **Complex designs** Compared to other open-source and free RDBMS implementations, for complex database designs, PostgreSQL offers the most in terms of functionality and possibilities without giving up on other valuable assets.
DISADVANTAGES	**WHY NOT USE**
Performance For simple read-heavy operations, PostgreSQL can be an over-kill and might appear less performant than the counterparts, such as MySQL. **Popularity** Given the nature of this tool, it lacks behind in terms of popularity, despite the very large amount of deployments - which might affect how easy it might be possible to get support. **Hosting** Due to above mentioned factors, it is harder to	**Speed** If all you require is fast read operations, PostgreSQL is not the tool to go for. **Simple set ups** Unless you require absolute data integrity, ACID compliance or complex designs, PostgreSQL can be an over-kill for simple set-ups. **Replication** Unless you are willing to spend the time,

ADVANTAGES	WHY USE
come by hosts or service providers that offer managed PostgreSQL instances.	energy and resources, achieving replication with MySQL might be simpler for those who lack the database and system administration experience.

TABLE 6 – MYSQL MAIN PARAMETERS [DIGITALOCEAN, 2014]

ADVANTAGES	WHY USE
Easy to work with MySQL can be installed very easily. Third-party tools, including visual ones (i.e. GUIs) make it extremely simple to get started with the database. **Feature rich** MySQL supports a lot of the SQL functionality that is expected from a RDBMS directly or indirectly. **Secure** A lot of security features, some rather advanced, are built in MySQL. **Scalable and powerful** MySQL can handle a lot of data and furthermore it can be used "at scale", if needed be. **Speedy** Giving up some standards allows MySQL to work very efficiently and cut corners, thus providing speed gains.	**Distributed operations** When you need more than what SQLite can offer, including MySQL to your deployment stack, just like any stand-alone database server, brings a lot of operational freedom together with some advanced features. **High security** MySQL's security features provide reliable protection for data-access (and use) in a simple way. **Web-sites and web-applications** A great majority of web-sites (and web-applications) can simply work on MySQL despite the constraints. This flexible and somewhat scalable tool is easy to use and easy to manage which proves very helpful in the long run. **Custom solutions** If you are working on a highly specific and extremely custom solution, MySQL can tag along easily and go by your rules thanks to its rich configuration settings and operation modes.

DISADVANTAGES	WHY NOT USE
Known limitations By design, MySQL does not intend to do everything and it comes with functional limitations that some state-of-the-art applications might require. **Reliability issues**	**SQL compliance** Since MySQL does not [try to] implement the full SQL standard, this tool is not completely SQL compliant. If you might need integration with such RDBMSs, switching from MySQL will not be easy. **Concurrency**

DISADVANTAGES	WHY NOT USE
The way certain functionality gets handled with MySQL (e.g. references, transactions, auditing etc.) renders it a little-less reliable compared to some other RDBMSs. **Stagnated development** Although MySQL is still technical an open-source product, there are complaints regarding the development process since its acquisition. However, it should be noted that there are some MySQL-based, fully-integrated databases that add value on top of the standard MySQL.	Even though MySQL and some storage engines perform really well with read operations, concurrent read-writes can be problematic. **Lack of features** Again, depending on the choice of the database-engine, MySQL can lack certain features, such as the full-text search.

TABLE 7 – ORACLE MAIN PARAMETERS [TECHWALLA, 2015]

ADVANTAGES	WHY USE
Grouping Transactions The ability to group several transactions into the same batch for processing sets Oracle apart from its competitors. Microsoft's version of SQL is limited to executing each transaction individually in a sequential order. This gives Oracle SQL far greater scalability over sequential versions of SQL, which are limited to only expanding vertically. Vertical expansion can be significantly more expensive than horizontal expansion. Expanding vertically requires the addition of server memory, hard drive capacity and ventilation equipment. Oracle can expand horizontally by clustering transactions for more efficient processing. **Improved Performance** In addition to batch processing of transactions, Oracle offers other methods of improving your database's performance. You can use multiple servers to work on the same database with the Real Application Cluster feature. This can significantly increase your processing power for only the price of an additional server. Oracle SQL also gives you more options to fine-tune the operation of your database to suit the capabilities of your server.	**Versatility** Oracle SQL gives the flexibility of choosing to run your database in any operating system. Dedicated languages are only compatible with operating systems from the same manufacturer. For example, you can only run Microsoft SQL Server on a Windows-based machine. In comparison, you can install Oracle SQL on a Unix server and benefit from the reliability of Unix while keeping the standardization of SQL. Unix is less vulnerable to many common computer viruses, which keeps your information secure. Oracle SQL is also backward-compatible, so you have the option of upgrading in the future without losing any data.

DISADVANTAGES	WHY NOT USE
Difficulty Oracle SQL is also more difficult to learn and operate than its competitors. A user-friendly version such as SQL Server is easy to install and set up with a minimum of customization. It includes five sample databases you can use as a framework for your own company's information. Microsoft also provides Visual Studio-based SQL Server management tools to help you modify the database design, change the data it contains and execute queries.	**Cost** The cost of operating Oracle SQL puts it at a disadvantage in comparison to other versions of SQL. The cost difference starts with the licensing fees, which may be up to ten times as high for Oracle. Because Oracle SQL is such a complex and highly specialized language, it can be difficult to find qualified database administrators to run your system. When you do find a suitable candidate, expect to pay two or three times more in salary than you would pay a comparable Microsoft or PostgreSQL administrator.

TABLE 8 – MS SQL MAIN PARAMETERS [TECHWALLA, 2016]

ADVANTAGES	WHY USE
Enterprise-Grade Management Software Microsoft SQL Server includes professional, enterprise level database management software. A few competitors, such as MySQL, have developed similar software in recent years, but Microsoft SQL Server is easier to use and has more features. For example triggers are fully supported in Microsoft SQL. MySQL introduced triggers recently, but they are not fully supported. The software offered by Microsoft also offers close integration with the .NET framework, which is not the case with competing products.	**Excellent Data Recovery Support** Corrupt data is always a concern when power loss or improper shutdowns occur. Microsoft SQL Server has a number of features that promote data restoration and recovery. Although individual tables can't be backed up or restored, complete database restoration options are available. Through the use of log files, caching, and backups, Microsoft's product allows you to feel confident that disaster recovery options are plentiful.

DISADVANTAGES	WHY NOT USE
Limited Compatibility Microsoft SQL Server is only designed to run on Windows-based servers. For various reasons, including licensing costs and security concerns, developers may opt to host their websites on Unix-based machines. They would be unable to use SQL Server in this case. Competing products are frequently able to run on other platforms.	**Cost** One of the major disadvantages to using MS SQL Server instead of an alternative relational database management system is that the licensing options are expensive. Although using the software for development or educational purposes is free, any sort of business use incurs a licensing fee. SQL Server 2008 Standard

DISADVANTAGES	WHY NOT USE
Unlike MS SQL Server, MySQL is supported on every major platform, including Windows, Linux, Mac OSX, and other Unix variants. In addition to being unable to run on non-Windows platforms, there also may be compatibility issues regarding interaction with applications that are running on other platforms.	Edition costs $7,171 per processor. The SQL Server Datacenter is $54,990 per processor. Competing software, such as MySQL, is often free for use, but the most expensive MySQL Enterprise package costs $4,999 per server. This is significantly cheaper.

5 THE SYSTEM DESIGN

This chapter describes design of a diagnostic system. At the beginning I focus on general diagnostic process, then on general functionality (functional scheme) and finally main structure of the system. All three parts are then used for modelling the system (creating objects and interactions between them) described in Chapter 6.

5.1 BASIC DIAGNOSTIC PROCESS

FIGURE 10 – IMPROVEMENT PROCESS SCHEME

Continuous improvement process has four steps – diagnostics, report, action planning and improvement (see Figure 10). This process can be extended by monitoring of Key Performance Indicators (KPIs). This proposed scheme is applicable for business tools with long term process – once a company passes through diagnostic, the system generates a report with current state evaluation and advices on how to improve particular processes in the company. Then an action plan using diagnostic results is compiled and other advices chosen by company representative who chooses KPIs for monitoring as well. By fulfilling the action plan an improvement is performed. When the action plan is fulfilled or after some period of time a diagnostic part is repeated and both reports are compared.

Similar scheme can be applied for short time process with different data exchange and without a report. Diagnostic, results generation and improvement is performed in real-time without report generation. This process has no pause, once it is started it runs continuously until is stopped.

5.2 Overall system functionality

A design part proceeds with overall system design. The system must be flexible in order to support various models on the one side and several processing methods on the other. At first an overall functional scheme was designed. Figure 11 shows an outline of the initial idea.

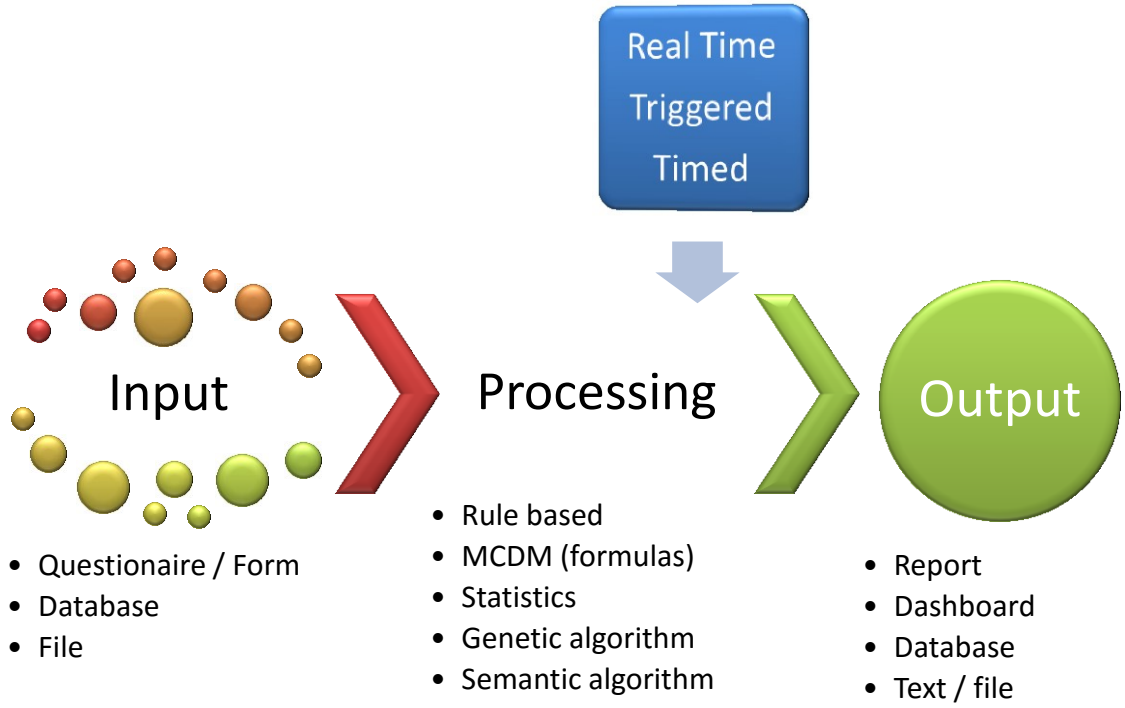

FIGURE 11 - MAIN FUNCTIONAL SCHEME OF THE DIAGNOSTIC SYSTEM

Input data are acquired via input interface or automatically directly from database or structured file i.e. XML. Processing is triggered once all input data are complete usually by a user (clicking the start button). A specific case can require a timed process starting; hence an application is equipped by timing service with custom settings. The last and probably the most interesting is the real-time processing – the system is waiting for changes in input data processing and displaying the output immediately. Processing methods are customizable – the idea is to have a modular (plug-in) format to support future extensions by modern methods or allow simple addition of new methods. Computational methods are able to process single-user input data and deliver an output. Moreover, an administrator can run the process over several (or all) users and generate an output. An output will be related to particular purpose – report (text and images, charts, graphs), dashboard – an overview of output values particularly in connection with real time processing. And finally database or files will be important once a linked operation is used.

5.3 STRUCTURE OF THE SYSTEM

The structural idea comprises a central executive unit – an interpreter (inference engine) integrating all other compartments together, using their resources and processing the results. The interpreter part is fixed (except for its settings) – all flexibility is included in a diagnostic model and an algorithm part. The diagnostic model part is a set of instructions for interpreter (engine) on how to maintain the diagnostic process – input data specification, user directions, restrictions, display format, layout etc. This part is accessible via admin interface to model designer who performs changes or designs new models. The algorithm part consists of computational techniques for data processing.

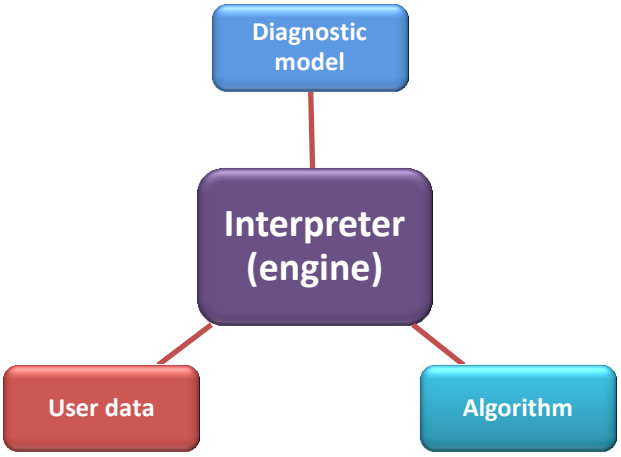

FIGURE 12 - MAIN STRUCTURAL SCHEME OF THE DIAGNOSTIC SYSTEM

A user data part is dedicated to storing user data – input as well as processed output data. If we assume the model and the algorithm are implemented, the overall diagnostic process from user's point of view will be following:

1. A user must go through registration process - once registered is allowed to log into the system
2. A user is provided with an option table with all diagnostics available. A user creates an instance of a diagnostic or selects an existing one.
3. A user is instructed for filling out all required input data.
4. After a set of input data is complete, user is asked to launch processing of input data. If more than one processing method is available, user is asked to choose one option.
5. Once the process is finished, user is provided with results – report or different relevant output.

The diagnostic system has three access roles – user, designer and administrator (see Figure 13). The user is only allowed to create instances of available diagnostics, fill out input data, launch data processing and generate an output. A designer can access a back-end interface

allowing all database actions (CRUD) on diagnostic model data. An administrator is allowed to view all data of all users and data post-processing using different methods or across several or all users.

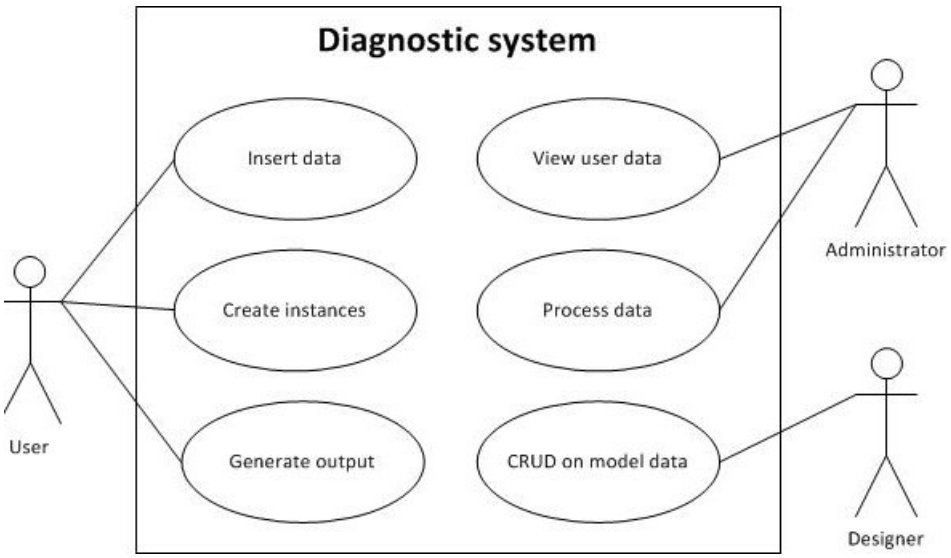

FIGURE 13 - DIAGNOSTIC SYSTEM USE CASE DIAGRAM

Figure 14 shows a detailed view of user data and model part. Colorings respect the previous scheme (Figure 12). A database structure is designed nearly according to this scheme including the interactions. A diagnostic model part includes main model identification, inputs and outputs that could be categorized. Every input variable can be in a form of a matrix with its rows (alternatives) and columns (criterions). Input variable can be restricted by a scale or a set of possible values (options). All user data – instances, inputs, outputs are stored separately from model data, but linked together.

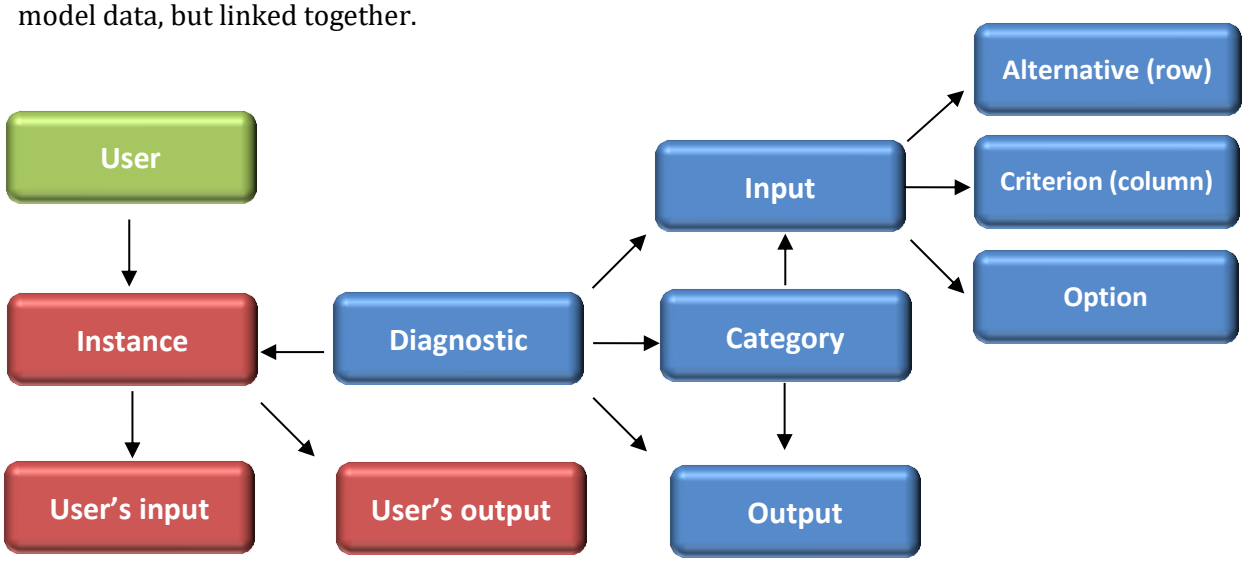

FIGURE 14 – INTERACTIONS BETWEEN USER DATA AND DIAGNOSTIC INSTRUCTIONS PART

6 MODELLING THE SYSTEM

After system design we know the main diagnostic process, we know how we want the system work, its structure and we can proceed to model its components – objects and interactions between them. I use general modeling way as we are not restricted in choosing development and database system. This model is finally used for development (see Chapter 7).

6.1 SYSTEM OBJECTS

In the database system as well as in the programming environment system objects for all designed items were created. This approach enables best way for database instructions (CRUD) and programming maintenance. System objects correspond with scheme in Figure 14.

6.1.1 INPUTS

All input data are simultaneously stored into 4 tables – *Input, Alternative, Criterion* and *Option*. *Input* table is a main object and other 3 tables are dependently filled while keeping same primary key (Input ID). *Input* is also linked with *Diagnostic* (diagnostic version) and *Category* (within diagnostic version) objects as each input must be categorized (or subcategorized) as well as linked with particular diagnostic version. *Alternatives* and *Criterions* are row and column identifiers of input data matrix. *Options* are variants of answers in case we need to restrict answers by forcing user to select one from a list.

Input object

Column	Type	Description
DiagnosticID	int	Primary key
SubareaID	varchar(20)	Primary key
InputID	int	Primary key
Text	text	Main text for input particularly question text
Instruction	text	Additional text for acquired input data in front of available answers i.e. “Choose one only”
AnswerType	varchar(20)	A = answer, P = percentage, S = selection
AnswerCountLimit	int	Maximum possible count of chosen answers

Column	Type	Description
Order	int	Order of input within diagnostic process
GroupID	varchar(20)	Identifier of input data group (input data requests can be placed in groups onto one page)
Tooltip	varchar(500)	Text used for tooltips (not mandatory)
NoDisplayCondition	varchar(500)	Rule which once is satisfied, input data request is not displayed
Params	text	Other parameters
DefaultAnswerID	int	Default answer ID in case this data request is not displayed to user

Input item (alternative)

Column	Type	Description
DiagnosticID	int	**Primary key**
SubareaID	varchar(20)	**Primary key**
InputID	int	**Primary key**
InputItemID	int	**Primary key**
Text	varchar(500)	Main text for input item
CustomValue	bool	If custom text value is supported
Tooltip	varchar(500)	Text used for tooltips (not mandatory)
DefaultValue	varchar(500)	Default item value if not filled
Order	int	Order of item within particular input

Input item options and variants

Column	Type	Description
DiagnosticID	int	**Primary key**
SubareaID	varchar(20)	**Primary key**
InputID	int	**Primary key**

OptionID / VariantID	int	**Primary key**
Text	varchar(500)	Main text for input option / variant
Order	int	Order of option / variant within particular input item

User's Input

Column	Type	Description
InstanceID	int	**Primary key**
SubareaID	varchar(20)	**Primary key**
InputID	int	**Primary key**
InputItemID	Int	**Primary key**
VariantID	int	**Primary key**
State	int	Main text for input option / variant
Value	varchar(200)	Order of option / variant within particular input item

6.1.2 RESULTS

Results objects consist of result, logic matching and user's result objects. First one is used for storing all possible results within one diagnostic. Logic matching object is used for generation variable string parts from user's data. These strings are later used in reports by filling them info specific tables or charts. Last table is used for storing results after all data processing. So we can say there are only results their logic was processes and evaluated as valid for current user (according to their user's data).

Result

Column	Type	Description
DiagnosticID	int	**Primary key**
SubareaID	varchar(20)	**Primary key**
ResultID	int	**Primary key**
Logic	text	Logic string (rule based logic)
Text	text	Text description
Summary	varchar(50)	Priority (Level 1,2,3)
Recommendation	text	Recommendation text for result
Last	bool	Once a last result is selected , result processing is terminated in particular area

Column	Type	Description
ShortSummary	text	Short description
ActionURL	varchar(200)	Suggested action URL (redirects user to advisory page)
ActionTemplateID	uniqueidentifier	Link to action template
Reason	text	Reason why this result was generated (corresponds with rule based logic)

Logic matching

Column	Type	Description
DiagnosticID	int	**Primary key**
TagID	varchar(30)	**Primary key**
LogicID	int	**Primary key**
Logic	text	Rule based logic
Result	text	Result text
ResultType	varchar(1)	S=sentence, I=input item

User's result

Column	Type	Description
InstanceID	int	**Primary key**
SubareaID	varchar(20)	**Primary key**
ResultID	int	**Primary key**
Recommendation	text	Recommendation text for result – finalized (by joining static and variable parts)

6.1.3 CATEGORIZATION

Categorization consists of three objects in hierarchy – so on the top there is *Main area* then *Area* and finally *Subarea*. Categorization is used for navigation during data acquisition process as well as reporting as we need a hierarchy used for summary reports, tables and graphs.

Main area

Column	Type	Description
DiagnosticID	int	**Primary key**
MainAreaID	int	**Primary key**

Column	Type	Description
Name	varchar(50)	
Description	text	
Order	int	

Area

Column	Type	Description
DiagnosticID	int	**Primary key**
AreaID	int	**Primary key**
Name	varchar(50)	
Description	text	
Order	int	
MainAreaID	int	Link to main area

Subarea

Column	Type	Description
DiagnosticID	int	**Primary key**
SubareaID	int	**Primary key**
Name	varchar(50)	
Description	text	
Order	int	
AreaID	int	Link to area

6.1.4 DIAGNOSTIC AND INSTANCE OBJECTS

Diagnostic object includes basic diagnostic information. Once user logs into diagnostic system, only active diagnostics are displayed for selection. System supports many diagnostic versions in one time. Once a user selects a particular diagnostic version an instance of this diagnostic is created. So one user can have many instances in one time (finished or not finished) and if the acquisition process is terminated (for any reason), current status is stored in each step. So it is possible to log into the system again and continue. Translation table is used for storing different language versions of texts. System uses hash method for encoding. English text is hashed and hash string is used as primary key as well as for searching foreign language versions. User's personal data and credentials are taken directly from CMS database tables.

Diagnostic

Column	Type	Description
DiagnosticID	**int**	**Primary key**
Name	varchar(50)	
Description	text	
CreatedDate	datetime	Date of creation
UpdatedDate	datetime	Date of last update
Active	bool	Active/ Inactive

Instance

Column	Type	Description
InstanceID	**int**	**Primary key**
DiagnosticID	int	Link do Diagnostic
CreatedDate	datetime	Date of creation
CreatedUserID	uniqueidentifier	Creator's ID
OrganisationGroupID	uniqueidentifier	Organisation's ID
UpdatedDate	datetime	Date of last update
UpdatedUserID	uniqueidentifier	Last update user's ID
ResultsGenerationDate	datetime	Date of results generation
ActionsChangeDate	datetime	Date of action plan generation
VersionID	int	Version of diagnostic
Language	char(2)	Language code i.e. EN, CS

Translation

Column	Type	Description
DiagnosticID	int	**Primary key**
HashID	char(32)	**Primary key**
Language	char(2)	**Primary key**
Text	text	Text of translation

6.2 ACQUIRING AND STORING INPUT DATA

As we already have system objects we can start modelling functionality between them. At first we have to acquire input data from user and store them into database in a meaningful way. Most used way of acquiring input data is a questionnaire. Diagnostic system designer uses predefined constructions of data which means using questions, giving alternatives, options etc.

Sometimes forcing user to choose an option or answer in a restricted area of possible alternatives is required. Otherwise there is fully custom answer expressed in a string. As we need a questionnaire with dynamic navigation (forward and backward steps), storing procedure is executed every time we move to next or previous question. For storing data input objects (input, alternative, variant, category) are used as this structure appears to be just enough versatile for most purposes of diagnostic systems (see structure in Figure 12 and objects in 6.1.1 and 6.1.3). First model (at the beginning of the project) used just an input item and its alternative which means 2D structures. By introducing variants 3D structures were created and allowed wider scale of data acquiring as well as better displaying options and developing user friendly interface. So we can simply create a form with checkboxes and force user to check particular count of them, or we can offer a few options as a radio button with one possible answer, or drop-down list with predefined answer or finally just text box for custom answer. We are also allowed to set rules for forms appearance as sometimes we need to have an additional information to previous question and it depends on user response i.e. once user responses the SME has no production, CD will not ask for number and types of production machines. For CD purposes we expect using only questionnaire for acquiring data but generally the system supports automatic retrieving and storing of input data i.e. by reading data from database (or sensors etc.) by automated procedure executed continuously in defined interval.

6.3 DATA PROCESSING

One of the main tasks of this dissertation was to transform idea of diagnostic system and its instructions formulated in human language into computer code, but there was one more condition, as we need dynamically extensible and flexible system, no hardcoded parts were allowed. First view of diagnostic ideas led to a higher grade and sophisticated system design that would work with strictly defined internal code used by interpreter engine. The engine uses internal code, dynamically transforms it into executable code (in a way of dynamic or real time compilation), then uses input data from (we can generally say) user for processing final result. Interpreter is a general computing part of diagnostic system and uses model as set of instructions on how to work with data. Model includes a system for acquiring input data including instructions for the user, restrictions of value as well as what processing algorithm will be used for computing final result.

6.3.1 *FROM TEXT INSTRUCTIONS TO EXPERT SYSTEM*

Text instructions were initially formulated in following way:

If in a question B1 is selected answer 6 OR in question B1 is selected answer 7 AND in question N2 answer 5 is unchecked THEN use result B1

This way of formulation led me to design a very general computing system. Each question will be categorized and will have its particular responses. It means each question category will have a String ID, generally a few letters, initially just one letter as we need approximately 12 categories. Question as well as response has its unique numeric (integer) ID. Interpreter loads internal code, parses the string command, transforms it into executable code, dynamically compiles that code, substitutes variables from user data, and executes the code getting the result. Internal code for previous command will be:

Condition	Result
B1.6 OR B1.7 AND !N2.5	**B1**

A similar process is used for acquiring data from the user. Each question (input item) has a no display condition which if it is satisfied (result for logic operation is true), system does not display that question and jumps to next one i.e. we have a set of questions about production planning. First question ask if a company has a production plant, if not, we need to jump (over all questions about production) into next section. This was initially formulated i.e.:

If ALL answers in question N1 unchecked MOVE onto the next sub-section

---

This means we need to have a condition that is evaluated after N1 responses are sent to server before we serve next question. The process of evaluation is the same as in previous part and we use the same interpreter method to get the result. Internal code is simple and straightforward.

Condition	Next question
!N1	**O1**

Logic string consists of question, its responses, options, variants in a form of IDs. The simplest expression is when only simple answer (if a response was chosen by a user or not) is required. It can be done by writing subcategory ID, question id, response ID and variant ID divided by dots. For example B1.6.12 means that subcategory ID is N, question ID is 1, response ID is 6 and variant ID is 12. Only subcategory and question ID is mandatory, other parameters are optional. If there is only subcategory and question ID specified, interpreter analyses if all

responses are checked by a user. A more complex expression with more than one variable is shown in Figure 15. More variables can be chained using logic operations (AND, OR) as well as negation (exclamation mark) placed just at the beginning of a variable (specification of a question). There are a few more syntax words i.e. ANY placed before a variable means that any of following variables (list of variables) can be TRUE to have final TRUE answer as it is a shortcut for situations when we need to check many (or not known count) of responses within one question.

For example

N1 OR !N1

---

means that we will get TRUE result as long as all N1 answers are checked or all are unchecked. Once we use

---

ANY N1 OR ANY !N1,

---

we will get TRUE result when at least one N1 answer is checked or at least one answer is unchecked.

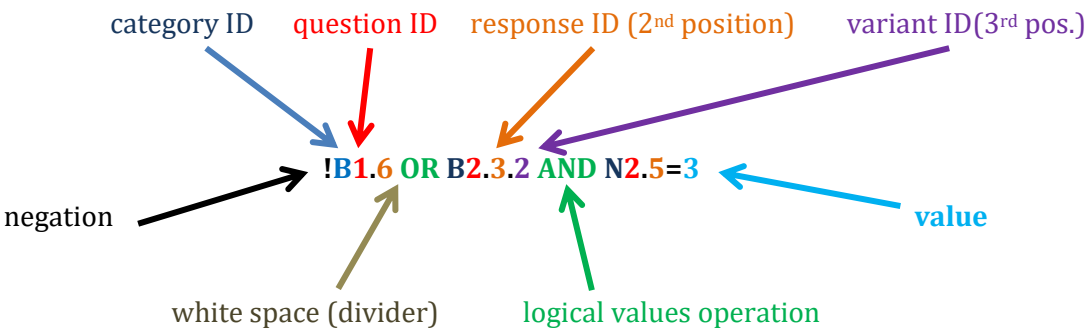

FIGURE 15 – BASIC LOGIC STRING COMPOSITION

6.3.2 *RULE BASED INTERPRETER PRINCIPLE*

Interpreter works with specified method. When we choose rule based method, interpreter uses predefined string with rule based logic. In capability Diagnostic, rule based logic is specified in questions (i.e. no display condition), results (i.e. logic for choosing results matching with user data) and logic matching when composing variable string for inserting into fixed text report parts.

Once a user finishes a diagnostic process (filling input values), the system allows results and report generation. First, all result rows are retrieved from database into an array and that array is row by row processed. Each record has its rule string (named logic in database). This rule has a format specified in Figure 15. There are 2 basic alternatives divided by white space. First alternative is a variable (named question) that can be extended to logic comparison (i.e. N2.5=3), second is logical operator (i.e. AND, OR). Interpreter always analyses variable parts as a partial logic and composes a logic formula.

For example if there is N2.5 specified, user data for question in subcategory N with question ID 2 and response ID 5 is retrieved and once there is true value in the record (response is checked by user), true value is returned and added as a partial value into final formula. Once there is extended logic comparison i.e. N2.5<3, system retrieves value selected (from list of options in select box) or filled (directly typed into input box) from user data table. If user's value corresponds with the rule (value is less than 3), returns true status.

Second alternative is logical operator. Logical operators (specified by strings) are simply translated into programming syntax of particular language. After processing all items in rule base logic string, final formula is analysed and final status returned (see all process flowchart in Figure 16).

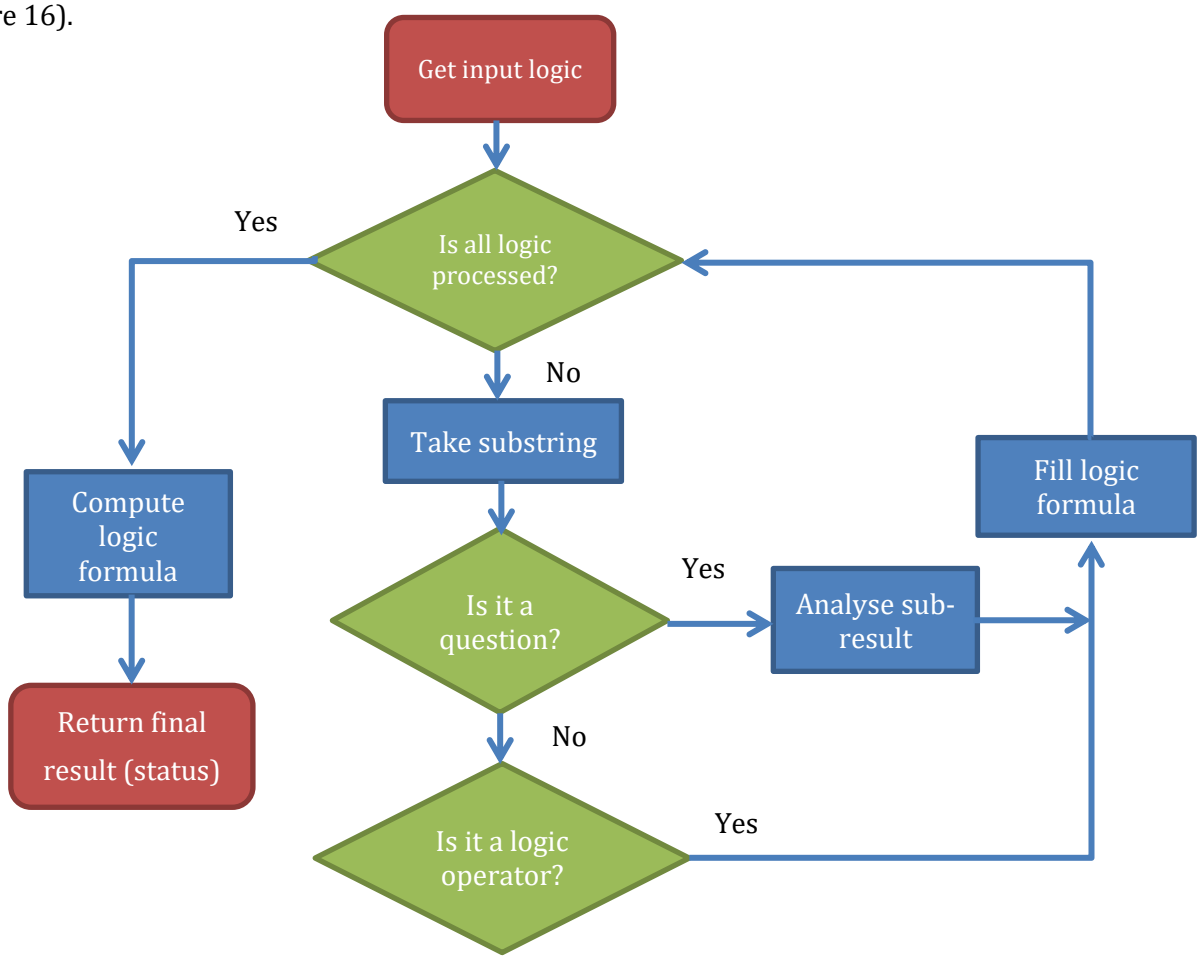

FIGURE 16 – OVERALL RULE BASED PROCESS FLOWCHART

Initial logic string i.e. "!B1.6 OR (B1.3.2 AND N2.5=3)" can be transformed into 0 + (1*1) – in case a user checks response 6 for question 1 in subcategory B as well as response 3 in variant 2 for question 2 in subcategory B and at the same time fills (or chooses) value 3 for response 5 for question 2 in subcategory N. True status will be returned as final logic result and this leads to next part of code that stores matched result into database.

Previous text described selection process – if a rule based logic is processed and corresponds with user's data, result is selected. But there are variable strings within particular rule (each logic rule row has text column). When a rule is selected, text must be completed by joining static (fixed) part and variable part. Variable part has 2 alternatives – response text or logic matching.

Generating result is based on acquired data and is described i.e.:

There are a number of avenues that can be explored to develop or expand the capabilities of the business. It may be useful to investigate how ***<<unchecked R4 responses>>*** *could benefit the organisation. For more information on these, follow the links.*

---

or

When competing on the basis of ***<<all checked answers from question B1>>****, critical processes and resources are* ***<<custom rule based text related to B1 and C1 question>>****. Assess those processes and resources that are critical to how you compete to ensure that they are managed efficiently and effectively. For guidance on how to do this, click* ***<link to advisory page>****.*

So there are fixed strings of text interspersed by variable text parts coming from computational process (bold text in <>). This request needs a special processing method for substituting variable parts in fixed text. We have to introduce internal coding system similar to rule based logic for logic formulas described above. All variable codes are placed into double angle brackets (chevrons) as single chevrons are used in HTML syntax.

We have to rewrite previous expressions into:

There are a number of avenues that can be explored to develop or expand the capabilities of the business. It may be useful to investigate how **<<!R4 responses>>** *could benefit the organisation. For more information on these, follow the links.*

---

and

---

*When competing on the basis of **<<B1>>**, critical processes and resources are **<<#B1,C1#>>**. Assess those processes and resources that are critical to how you compete to ensure that they are managed efficiently and effectively. For guidance on how to do this, click <**a href="http://www.futuresme.eu /what-is/visual-strategy" target="_blank"**>here**.*

<<!R4>> means that this tag will be replaced with all unchecked responses (comma separated strings) from R4 question. **<<B1>>** means that this tag will be replaced with all checked responses in B1 question. **<<#B1,C1#>>** means that this tag will be replaced with processed logic from logic matching table (see below). Hash signs (#) denote logic matching tag ID. There are several modes to work with these tags. At first we have to process logic rule i.e. A1.2 OR A1.3. If this rule is evaluated as true, we can proceed according to the type :

1. S type means that processing method returns a sentence (fixed string),
2. I type – requires a few more processing steps – we need to process all logic in Result column and return all input item (comma separated) texts that are evaluated as true.

TABLE 9 – EXAMPLE OF LOGIC MATCHING RECORDS

Tag	ID	Logic	Result	Type
A	1	A1.2 OR A1.3	Understand and clearly articulate why the business exists	S
A	2	A2.2 OR A2.3	Develop a vision statement that is written or verbally communicated	S
A	3	A3.2 OR A3.3	Communicate the vision to everyone in the business	S
B1, C1	1	B1.2	C1.2.1!=2 C1.11.1!=2	I
B1, C1	2	B1.3	C1.2.1!=2 C1.6.1!=2 C1.7.1!=2 C1.8.1!=2	I
B1, C1	3	B1.4	C1.4.1!=2 C1.6.1!=2 C1.9.1!=2	I

Result of processing B1, C1 logic tag can be string “Manufacturing/assembly/production process, Quality control/management ”. So the final texts prepared for use in reports can be i.e.

---

*There are a number of avenues that can be explored to develop or expand the capabilities of the business. It may be useful to investigate how **Collaboration, Recruiting skilled employees and Outsourcing** could benefit the organisation. For more information on these, follow the links.*

---

and

> *When competing on the basis of **Quality of products/services**, critical processes and resources are **Manufacturing/assembly/production process, Quality control/management**. Assess those processes and resources that are critical to how you compete to ensure that they are managed efficiently and effectively. For guidance on how to do this, click here.*

Last tag is classical HTML tag for a link and links user to advisory page (particular URL).

6.4 GENERALISATION

One of the main goals of this dissertation was to develop a general system – a system that would be useful for many (different) fields of life. This diagnostic system was designed in specific structure that allows splitting model data (knowledge base) from interpreter data. So a designer (a person who knows how to model a particular system and knows exactly his needs) is able to create his own (different) system. In the simplest way I want to create a poll and based on the input data I want to give the user some advices. For example, a car problem solving tool. At first the system asks user about specific problems (what indicator lights are on, what fuses are broken etc.) and then displays the description of the problem and particular set of advices. So the system can be kind of general allowing modelling our own instances of simple specific diagnostic system.

As we have discussed that CD is one possible way of using this system. Generally we can use this system for whatever questionnaire, polls, surveys or inquiries. This system offers not only storing data and displaying then in a (excel-like) table, but supports processing the input data and create reports with charts as well as processing the input data and create (and store) output values as action variables (in feedback control system) for controlling some process.

6.4.1 UNCERTAINTY

Uncertainty is a situation which involves imperfect and/or unknown information. It arises in subtly different ways in a number of fields, including insurance, philosophy, physics, statistics, economics, finance, psychology, sociology, engineering, metrology, and information science. It applies to predictions of future events, to physical measurements that are already made, or to the unknown. Uncertainty arises in partially observable and/or stochastic environments, as well as due to ignorance and/or indolence. [NORVIG et al., 2015]

Uncertainty in expert systems

- Expert systems provide an advantage when dealing with uncertainty as compared to decision trees.
- With decision trees, all the facts must be known to arrive at an outcome. Probability theory is devoted to dealing with theories of uncertainty.
- There are many theories of probability – each with advantages and disadvantages. [MISIASZEK, 2007]

There are several theories to deal with uncertainty i.e. Bayesian Probability, Hartley Theory, Shannon Theory, Dempster-Shafer Theory, Markov Models or Zadeh's Fuzzy Theory.

Many rule-based systems have the ability to reason about uncertain facts and events. At least three forms of uncertainty are built into a single rule. First, there is uncertainty associated with the rule antecedent. A second source of uncertainty is the level of confidence we have in the rule itself. A third focus requires a methodology for combining pieces of uncertain information. [ALESHUNAS, 2016]

Several methods have been implemented to handle uncertainty. Two general categories of solutions are probability-based methods and heuristic methods. Probability based methods are exact reasoning strategies that make use of standard probability theory. Heuristic techniques, also known as inexact methods, are necessary when the assumptions required to correctly apply a probability-based approach cannot be met. Because of the inexact nature of data, heuristic techniques are often the method of choice for rule-based systems. Fuzzy logic is an accepted heuristic technique for reasoning about uncertainty in rule-based systems. Fuzzy logic was first introduced by [ZADEH, 1965] as a problem solving method for dealing with the uncertainty inherent in words with ambiguous meanings. Fuzzy logic relies on fuzzy set membership, which allows us to associate varying degrees of membership with the elements of a set.

Bayes theorem offers a mathematically correct approach to reasoning about uncertainty. The technique is particularly appealing when accurate statistical data, which allows conditional and a priori probabilities to be easily computed, is readily available. However, certain criteria must be satisfied for valid application of the theorem. Bayes theorem was first used for computing uncertainty in the expert system PROSPECTOR.

6.4.2 NORMALISATION OF CONSTANT AND VARIABLE VALUES

Data normalization is used when seeking for relations. We can do the normalization to reach a linear, more robust relationship. Commonly when the relationship between two dataset is non-linear we transform data to reach a linear relationship. Here, normalization doesn't mean

normalizing data, it means normalizing residuals by transforming data. So normalization of data implies to normalize residuals using the methods of transformation. [KHEDIVE, 2013]

In some data mining approaches we need to normalize input data; otherwise the network will be ill-conditioned. In essence, normalization is done to have the same range of values for each of the inputs. This can guarantee stable convergence of weights and biases.

The request on data normalisation is much more on model designer as he must aware of data range consistency. Designer is allowed to set input data ranges and then assembly formulas that take account all weights and biases.

7 DEVELOPMENT

This chapter describes development according to overall design from Chapter 0 and based on the model described in Chapter 6. As my main project goal was to develop an application for diagnosing a company, I need to describe, what the Capability Diagnostic is, what features were required and what purposes should the application serve for. Then I am going to explain how I chose programming environment, database system and all related 3rd party tools for development and presenting the application. First pilot version was tested with partner companies and the application was evaluated and improved based on their feedback.

7.1 WHAT IS CAPABILITY DIAGNOSTIC?

With the building blocks in place a number of tools and methodologies were developed to support the Transformation Process and enable capability development. A transformation process is any activity or group of activities that takes one or more inputs, transforms and adds value to them, and provides outputs for customers or clients. Where the inputs are raw materials, it is relatively easy to identify the transformation involved, as when milk is transformed into cheese and butter. Where the inputs are information or people, the nature of the transformation may be less obvious. For example, a hospital transforms ill patients (the input) into healthy patients (the output). [OpenLearn, 2016]

The first of these tools is the Capability Diagnostic. The Capability Diagnostic is an online tool that provides a detailed analysis of an organisation's Managerial, Strategic, Operational and Adaptive capabilities. It objectively assesses strengths and weaknesses in key areas of manufacturing SME and can help to identify and prioritise goals. It has been developed so that it can be completed by a management team collectively or by individuals within the organisation and compared collectively. It can be run over different time periods and results can be compared to determine progress. The outputs of the diagnostic are presented in a user friendly colour coded graphical overview which helps prioritise areas that need to be addressed. The Capability Diagnostic is an important step in the Transformation Process as it identifies key areas that should be addressed by the organisation and can form an important part of developing an organisation's strategy. The Capability Diagnostic is supported by a number of reporting formats, these include:

- **Capability Diagnostic**: A comprehensive analysis of the organisation's strategic, operational, managerial, and adaptive capabilities. It also includes key priorities and recommendations for improving these capabilities.
- **Summary Diagnostic**: An overview of the business capabilities and some brief recommendations for improving these capabilities.

- **Comparison Diagnostic**: Applicable only if at least two versions of the Full Diagnostic have been completed, this report enables the user to track their progress by comparing versions.
- **People Comparison**: An analysis of any differences in the answers provided by people who took the diagnostic individually. It helps determine whether the perceptions of your business capabilities are aligned. [FUTURESME, 2008]

How can we really tell how well our business is performing? Financial and production measures show us the results but understanding what is behind the figures is vital to continuously improve. The FutureSME business diagnostic assess SME's business in terms of four key capabilities that underpin an adaptive enterprise, the key to competing in the 21st century – strategy, operations, management, and adaptability. To assess these capabilities, eight indicators are used which collectively show the adaptive footprint of the business. [BITITCI et al., 2009]

The Capability Diagnostic can be completed in several ways. Trials with SMEs has shown that most value can be gained by using the tool as a driver for discussions within a management team (and external board if applicable) in actually going through the diagnostic questions together everyone gains an insight into various aspects of the business and a common understanding of strengths and weaknesses can be achieved. It also proves a useful input into strategy discussions. It is possible to involve an external facilitator to guide you through the diagnostic and enable everyone to participate. This may be someone you already work with, or you may want to invite a consultant or account manager from your local regional development agency. Of course, you can complete the diagnostic yourself and share the findings with your colleagues and management team, using it to inform decision making. It would also be insightful to ask your colleagues to complete it independently then compare your responses. Not everyone always perceives things in the same way!

Depending on how you complete it, the diagnostic should take between 40 minutes and 2 hours (clearly discussing each question and response will extend the completion time). The questions are divided into nine sections, corresponding to eight indicators, along with a section capturing information about your company (such as location, size etc.). You can complete the questions all in one go, or save your answers and return later. A progress bar will show you how complete your diagnostic is. Once all questions have been answered, you can generate the output report which begins with an executive summary highlighting key priorities for you to address, and provides a snapshot of your adaptive footprint. The remainder of the report details the results of your diagnosis.

7.2 SELECTING DEVELOPMENT ENVIRONMENT AND DATABASE SYSTEM

As the final diagnostic system was designed for using by SMEs, the operating system, development tools and end user software must be selected with regard to final costs.

When I was involved into the FutureSME project I started with analyzing current project server with its hardware and software. There was Linux OS running on the server and Drupal as content management system. So there were two (most suitable) options for development system (Java and PHP) and four options for database system. As Drupal uses PHP and I was quite experienced in PHP I chose it in favourite connection with MySQL database as Drupal used the same setup. Among databases, there were two paid versions (Oracle and MS SQL) and two free options (PostgreSQL and MySQL). So I decided to use PHP and MySQL database selected in most popular software stack called LAMP (Linux, Apache, MySQL and PHP) composed entirely of free and open-source software, suitable for building high-availability heavy-duty dynamic web sites, and capable of serving tens of thousands of requests simultaneously. PHP and MySQL are also available for Microsoft's operating systems under IIS as they are quite common on servers in recent years. By the reason of a rich experience and simple implementation into Linux or Windows servers, PHP supporting Object Oriented Programming (OOP) in version 5.3 generating XHTML code have been chosen. Javascript on the client's side as well as Java script based library called jQuery are used. JQuery offers wide scale of tools for creation of user-friendly interface.

First versions of the diagnostic system were made in pure PHP and MySQL and all graphic objects were directly created (drawn) by PHP as it supports object-oriented graphics. Creating graphic object from scratch using basic commands for creating lines or 2D objects is sufficient for simple graphics, but in later versions quite complex graphic outputs were required, so I decided to use open source graph library that supports PHP (this was only requirement) and pChart library was the right one for my purposes.

7.3 PILOT IMPLEMENTATION

A pilot implementation of Capability Diagnostic was launched in December 2010 for testing by development team. Both model and software were debugged during this process. The interface was styled using a project portal template (see Figure 17). Diagnostic system was developed as standalone application, but offers building into any kind of web page. Since the system was integrated into FutureSME portal, user registration and user credentials were linked to portal database including all identification and security issues. Introduction page (see Figure 17) provides user with basic instructions on how to complete Capability Diagnostic process a offers user to choose a version of CD. Once there is more than one instance of CD, user can

choose between any of them. It is possible to complete the instance and then make changes with no restrictions. User can also start new instance whenever wants.

FIGURE 17- INITIAL CAPABILITY DIAGNOSTIC SCREEN

The crucial point of expert system is the front end interface as it is an entrance gate for input data. Thus, we have to communicate with a person who is filling data into various forms, so we have to create them in very user-friendly way. The main request was to have I/O interface as a web page. So PHP generates HTML code and CSS for page styling is used. Colors were selected according to main FutureSME logo – black, white and shades of blue, yellow for tooltips and traffic lights colors (red, orange and green) for reporting. The main page is divided into two main compartments – navigation part (on the left side) and main part (questionnaire) see Figure 18. Navigation part includes a list of all subcategories, count of all questions within particular subcategory and number of completed questions. So the user can monitor the progress of diagnostic process. Once a user completes one subcategory, green tick symbol appears.

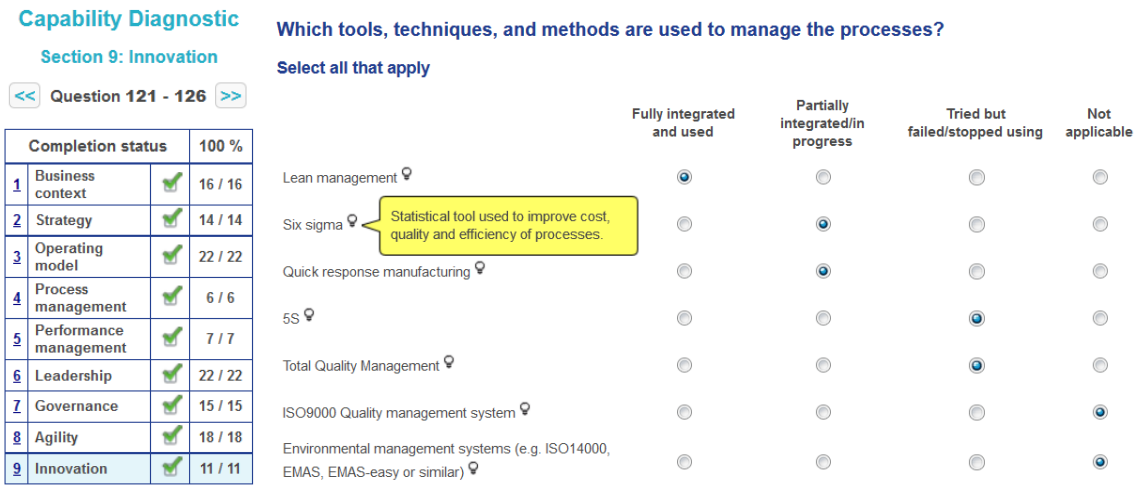

FIGURE 18 - CAPABILITY DIAGNOSTIC - A QUESTIONNAIRE

Originally there was a request for simple checkbox and select box form. There were designed questions with restricted from one to many options. If I (administrator) set that the particular question is restricted only for one answer, radio button instead of checkbox is placed. As all form is created with emphasis on clean and simple design, I sometimes needed additional information for user. This information can make the web page chaotic and full-filled once it was directly displayed as a standard pure text. A solution using tooltips was decided as most sophisticated for keeping the page clean while serving all necessary information to a user. In places where additional information is required, a clickable tooltip (bulb) sign was placed. Once a user wants to read that text, simply clicks the bulb and yellow bubble tooltip appears. It always appears in one instance so if there is one tooltip bubble and I click another bulb, the previous bubble disappears and I get additional information for current item.

Capability Diagnostic includes a questionnaire – 127 questions organized into categories (see Figure 18), rule based processing technique, reporting and action planning. All user (input and generated output) data are stored in the database they are thus available not only for reporting part but also for subsequent processing using different methods. Reports are dynamically created using WYSIWYG reporting with charting plug-in tool. Therefore styled text and indicators (Figure 24) complete their professional look and feel.

Capability Diagnostic is accessible through the FutureSME web portal that has a dashboard (see Figure 19). Dashboard is a central navigation point of the FutureSME portal offering custom tools and their results. Capability Diagnostic data are used for dashboard indication. So once a user completes a diagnostic process, main results are displayed on the dashboard (see Figure 19 top right corner).

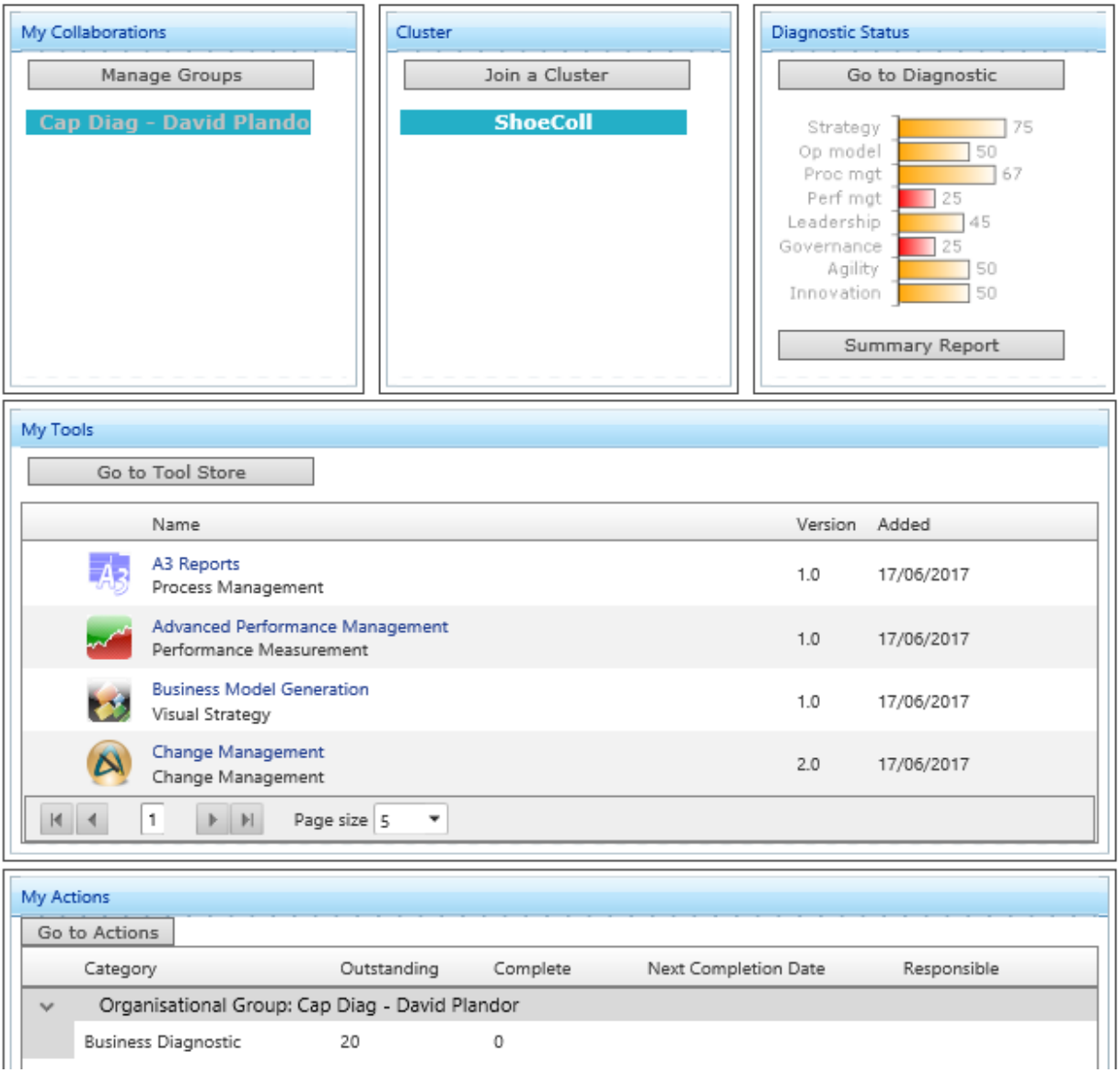

FIGURE 19 - DASHBOARD

7.4 Live System Operation and Evaluation

After a pilot operation which had lasted approximately 2 months, a system went live. Project management team announced system start-up and all end users were invited to register and go through the diagnostic process. Capability Diagnostic became a most popular tool on the portal – among over 70 representatives from partnering as well as non-partnering SMEs. There were 3 versions of diagnostic model developed for the duration of the project. The latter two versions were created without need of a programmer's intervention. The designer accessed the back-end interface and created brand new diagnostic models with new questions, new rules and reports.

There were made various changes in interpreter part as there were requirements for having input data in a form of a matrix or multiple instances of one diagnostic to compare the changes in time. These changes did not influence (only extended) the current system behavior and stored data. Despite most of SME leaders speak English; a multi-language support was implemented during the last period of the project.

The Capability Diagnostic is fully integrated into the portal and is accessed through the end user account. Capability Diagnostic was approved by over 70 company representatives from all Europe (mainly UK, Ireland, Czech Republic, Poland, Italy and Turkey) who underwent the diagnostic process (see Figure 20). Most of the managers and technicians went through the diagnostic process repeatedly to get the differential report with the overview of company's progress development in time.

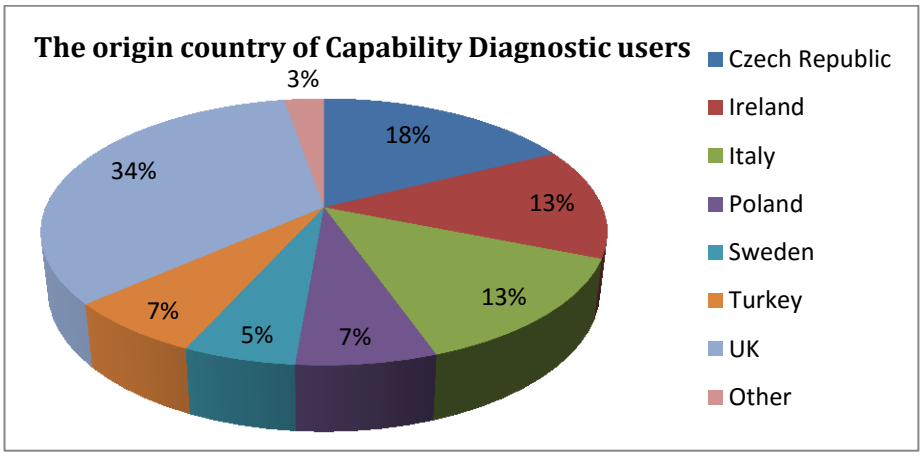

FIGURE 20 – THE ORIGIN COUNTRY OF CAPABILITY DIAGNOSTIC USERS

7.5 SYSTEM VERSION AND IMPROVEMENTS

During the project there were many changes, the crucial one was changing the server operation system from Linux to Microsoft Windows server and replacing Drupal by Sitefinity as a content management system. As Sitefinity uses MS SQL and MySQL data from Drupal were migrated into MS SQL, thus I have to do the same – migrate data, transform MySQL links into MS SQL and change SQL syntax in all controllers (methods used for CRUD operations between PHP and SQL). There was no difference in (the speed of) the response between these two database systems. As it would be very difficult to completely migrate all programming code into different programming language (i.e. Visual studio language that would have been my first choice if I had recently started development), I kept PHP. IIS was already installed because of Sitefinity, PHP was installed using Web Platform Installer (Web PI). Same application was loaded into application pool and all settings reconfigured for new system and database. After reconfiguration, front end part was incorporated into Sitefinity web page and there was no difference in speed of the application.

Main differences (and required changes in SQL code) between MySQL and MS SQL in syntax were (and the migration to MS SQL required):

- MySQL uses LIMIT while SQL Server uses TOP.
- SQL Server is much stricter on GROUP BY operations than MySQL, requiring that all non-aggregated columns from the SELECT appear in the GROUP BY clause.
- SQL Server supports a proprietary UPDATE FROM and DELETE FROM syntax that goes beyond the ANSI standard.
- Functions that exist in one system but not another. MySQL has FIND_IN_SET and GROUP_CONCAT that don't exist in SQL Server. Likewise, SQL Server has ROW_NUMBER() that doesn't exist in MySQL.
- MS SQL does not support the commenting option using #, but -- and /* */ work the same way.
- Different commands for same behaviour (i.e. NOW() x GETDATE(), UUID() x NEWID() or CHARACTER_LENGTH(string data) x LEN(string data)

All of these syntax differences had to be changed and all functionality tested.

Capability diagnostic was made in 3 versions. Main differences between versions were in different composition of input data pages, its processing as we need to have all results from different versions comparable. In second version I was asked to implement matrix data forms, so database and production code had to be upgraded (by addition of variant – criterion for answers). In last version instances for each diagnostic enabled later mutual comparison not only in time, but also between different users. This feature was mostly used within one company but the system allows comparison between every two instances in general.

TABLE 10 – CD VERSIONS PARAMETERS

Version	No. of questions	No. of instances	Finished instances	No. of answers	Results generated
V1	130 questions	80 instances	54 instances	40391 answers	1208 results
V2	127 questions	113 instances	54 instances	41838 answers	1462 results
V3	131 questions	101 instances	46 instances	37357 answers	1405 results

FutureSME partners were motivated to undergo CD process repeatedly following the scheme (see Figure 10). So they can evaluate how beneficial all the process was and that is the main reason of using CD – to help SMEs to improve processes leading to be competitive on the market.

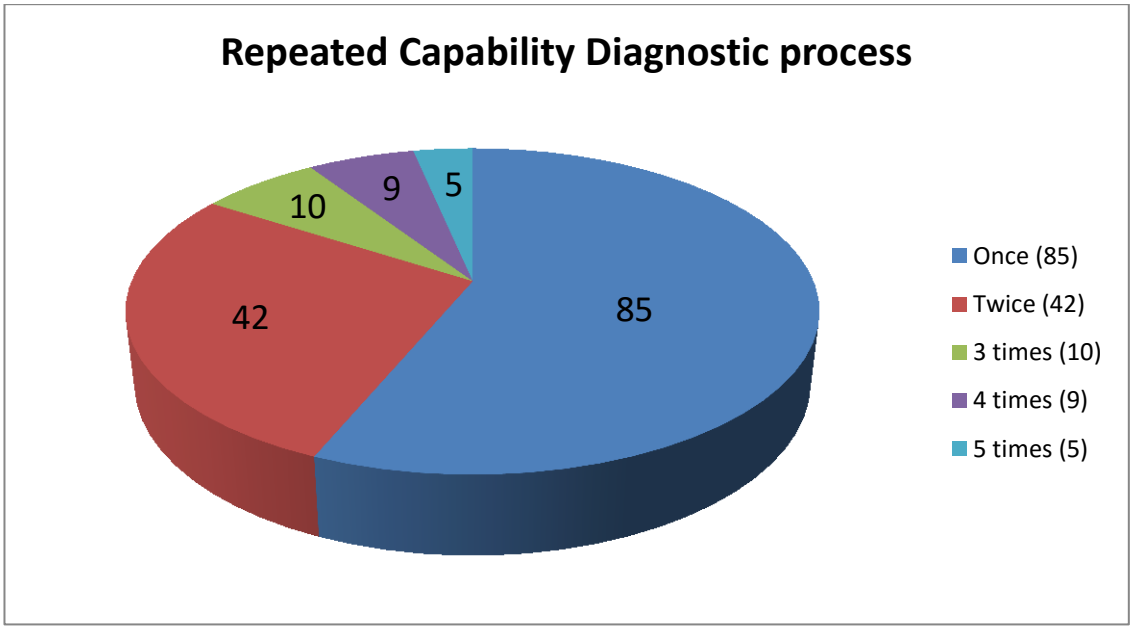

FIGURE 21 - OVERVIEW OF USERS WHO REPEATEDLY UNDERWENT CD PROCESS

8 DATA POST-PROCESSING

Once data acquiring ends diagnostic system is ready to process them and generate outputs. After first CD process is done, 3 reporting options are proposed. Once a user has more than one CD version complete, comparison report is proposed as well (see Figure 22).

Click on the options below to access your results

Report	Description	
Is your business ready for 21st century? Capability diagnostic report	This report is a comprehensive analysis of your diagnostic results in terms of the strategic, operational, managerial, and adaptive capabilities of the business. It presents your company's adaptive footprint, key priorities and recommendations for improving the business capabilities.	Show report
Summary diagnostic report	The summary report provides an overview of the business capabilities by presenting the company's adaptive footprint and a list of recommendations for improving business capabilities.	Show report
Are you becoming an adaptive enterprise? Comparison diagnostic report	The comparison report shows the change in capabilities of the business between a number of versions of the diagnostic. It allows the business to track progress towards becoming an adaptive enterprise.	Show report
How aligned are your team's perceptions of the current business capabilities? People comparison report	This report shows any differences between the answers provided by people who have taken the diagnostic individually. It allows the business to understand whether or not the perceptions of the current state of the business capabilities are aligned.	Show report

FIGURE 22 – DATA POST-PROCESSING OPTIONS AFTER CD COMPLETION

Report is main output of CD, includes advices on how to improve particular section of the company. CD also supports more powerful tool – action plan – that helps to monitor all individual partial steps (tasks) to reach the goal – improvement. On the basis of processed results KPI set for monitoring is proposed.

8.1 REPORT

A user is allowed to start processing the results and generating a report once the completion status is 100%. Now we are going to describe a system for result processing and generating the report. All possible results are stored in a database as a part of the diagnostic template. The results are filed into a category, which can (but does not have to) correspond to the question categories.

Each result includes a text part with a description of a company's gap, recommendations how to improve or eliminate a gap and type of result – i.e. sustain, improve, urge, etc. The logic part includes a simply defined logic condition. A program processes the data in a cycle - it goes through the list of all results and the logic condition of each result is examined. A result is assigned to a user when the condition is satisfied. A recommendation text is generated for each result. The dynamically generated texts are inserted into the static template i.e. a list of ticked responses or responses with a value higher than 10% etc.

Appendix 1 – Strategic capability gap analysis and recommendations

Area	Subarea	Result	Gap	Recommendation
Strategy	Guiding vision	Sustain	There is a guiding vision in the business that is understood by everyone	For more information about guiding vision, visit the Strategy information page.
	Customer value proposition	Improve	You seem to lack a focused customer value proposition	The business should have a primary value proposition on which it competes, supported by one or two secondary propositions. You need to define why customers come to the business. For guidance on how to do this, click here.
	Core competencies	Sustain	The business is clear on which processes and resources it should focus on to deliver value to its customers	To learn more about core competencies, visit the Strategy information page.
	Strategy	Improve	There is room for improvement in your strategy development and implementation process	To ensure that a focused business strategy is development and implemented, the business should For guidance on how to do this, click here.
Operating model	Financial management	Sustain	The profit formula of the business is defined and operates effectively	To learn more about financial management, visit the Finance information page.
	Operations	Sustain	The budget allocation for the business is aligned to its priorities and competitive basis	For more information on budgeting and financial management, follow the links.
	Operations	Sustain	The business outsourcing decisions are in alignment with its competitive basis	For more information about outsourcing, visit the Operating Model information page.
	Social responsibility	Sustain	The business appears to behave in a socially responsible manner	For more information on social responsibility, click here.

FIGURE 23 - EXAMPLE OF GENERATED RESULTS

Results are displayed in a report – a classic web page with static and dynamically generated objects. The report template is always created by the designer. For designing the report structure the best a TinyMCE web library is used. Perhaps we can count on a programmer's contribution when we need to add some more complicated objects such as graphs. The report utilizes the pChart library for good quality graphs. At the end of the report a list of questions with responses is printed. The final report is exportable into PDF.

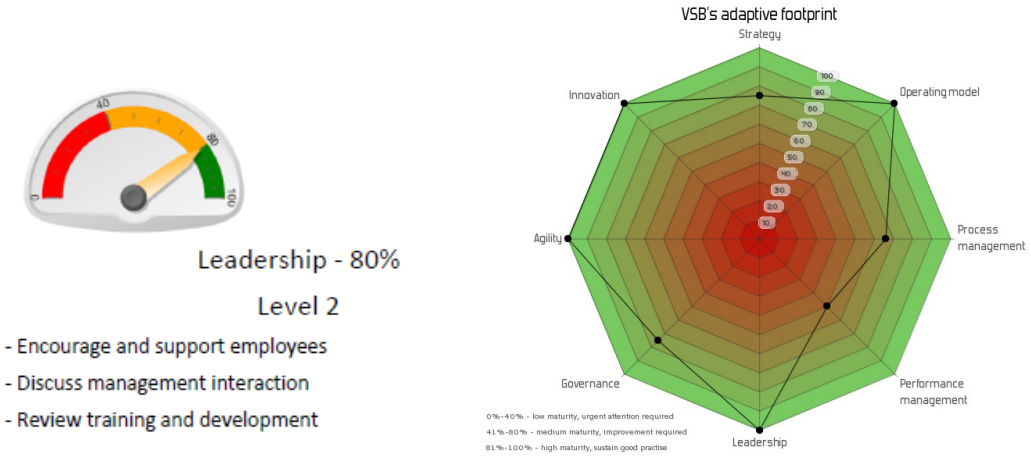

FIGURE 24 - REPORT INDICATORS

8.2 COMPARING RESULTS

As the capability diagnostic supports various instances of one diagnostic version (edition), comparing result for two different instances is supported. This can be helpful in case we have diagnostic data for one company acquired in different time. Mainly this is common to pass through the diagnostic process at the beginning (once the capability diagnostic is

introduced to SME) and after a few months when results from first "iteration" are applied to action planning and some steps from action plan are completed. Then we can display the progress and evaluate how capability diagnostic process can be helpful for the SME. However, CD can generally compare instances between two different versions, but CD designer has to beware of data consistency between particular two versions. Comparing results issue is only concerned with reporting part of CD as other parts (of executing code and database) work disregarding data consistency.

8.3 ACTION PLAN

The filling of the action list is available once diagnostic process is finished. It is a procedure, which uses all already stored user's results and inserts them into an action database table. The action list includes actions that are necessary for company planning in the future. The action list can be filled from other sources on the portal or actions can be created manually by a user in different parts of the portal as well. We will focus on data coming from the capability diagnostic. By clicking the button a procedure for filling the action list is invoked. A program goes through all results and stores them into an action table in the database. An overview of all actions is accessible in different part of the portal. This is done by implementing web page into CMS page in the same way as main diagnostic page. A user can assign additional data to an action i.e. responsible person, start and due date etc. A user can fill/recreate the action list periodically, always by the insertion of currently valid results.

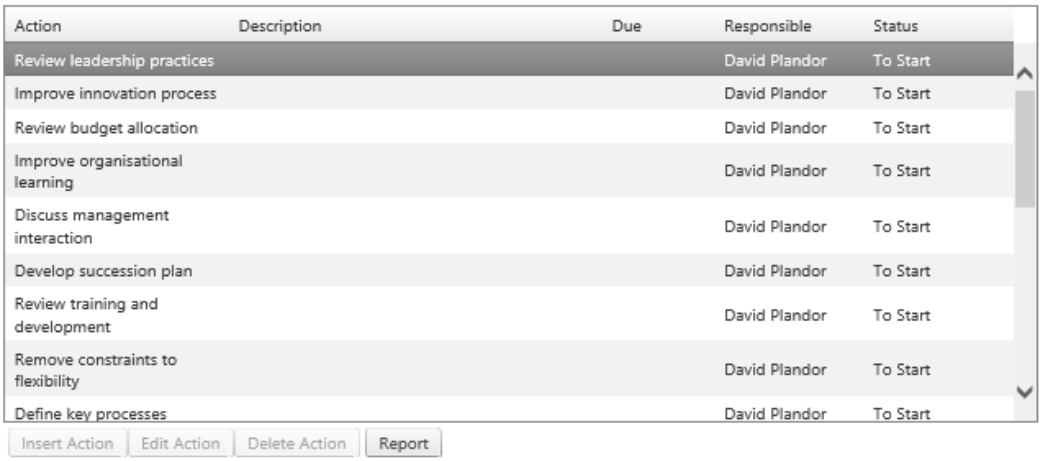

FIGURE 25 – ACTION PLAN – LIST OF ACTIONS

8.4 KPIS

KPIs are widely used measurement indicators of company performance based on the principle of the continuous monitoring of predefined parameters that affect a company's business the most. Determining them is not an easy process and it takes time and effort to fully implement them. Especially the time factor is a serious issue for the SME. Also, 2-4 employees in

the implementation team, which might be possible in large companies, may be too costly to have in an SME.

Standard KPIs were selected according to the best practices mapped during our research with partner European companies. They are related to non-financial measures, for example, quality control and preventing failures rather than dealing with failures after they appear - Right 1st Time, as well as financial measures, for example the overall sales in an exact time period - Sales per Month.

These KPIs are measured frequently, to be able to provide company management with actual results in a suitable form and offer a possible solution on how to handle possible deviations in business and operational performance. Table 11 shows an example of KPIs, not every reader will be familiar with these indicators, as the indicators themselves vary across each industry, sector, production and/or services, especially for the SMEs, which this tool is being developed for.

TABLE 11 - EXAMPLE OF KPIS

Right 1st Time	Sales per Month
On Time in Full	Profit per Product
Sales Generated	Cash Flow
Staff/Trainings	Production Costs
Process Lead Time	Client Management
Market Share (%)	Marketing Expenses
Investments in Innovation and Research	Ratio of full-time employees
Employee Satisfaction Index	Number of Registered Patents
Expenses for Research and Innovation	Added Value per One Employee
Total Expenses	Ratio of Marginal Revenue

8.5 TRANSLATIONS

Capability Diagnostic had to support language versions. Most sophisticated system for translations appears to use hash encoding for each text expression. So the idea is to encode every string (in English) into hash string used for primary key in database as well. As we have all text string in database we can offer list of all strings for translation. Translation method will take string in English, encode it into hash and search in database by hash and language code (i.e. CS for Czech). Once there is a record, returns foreign language string back. Once there is no translation, returns the same string (in English). This way of translating is advantageous in case of changing text or adding new text by designer during active diagnostic versions when updating language version is forgotten. This means that there is always some text in the front end part (user interface) instead of displaying blank spaces or throwing warnings. In the admin site there is always language version check during the start-up/login process.

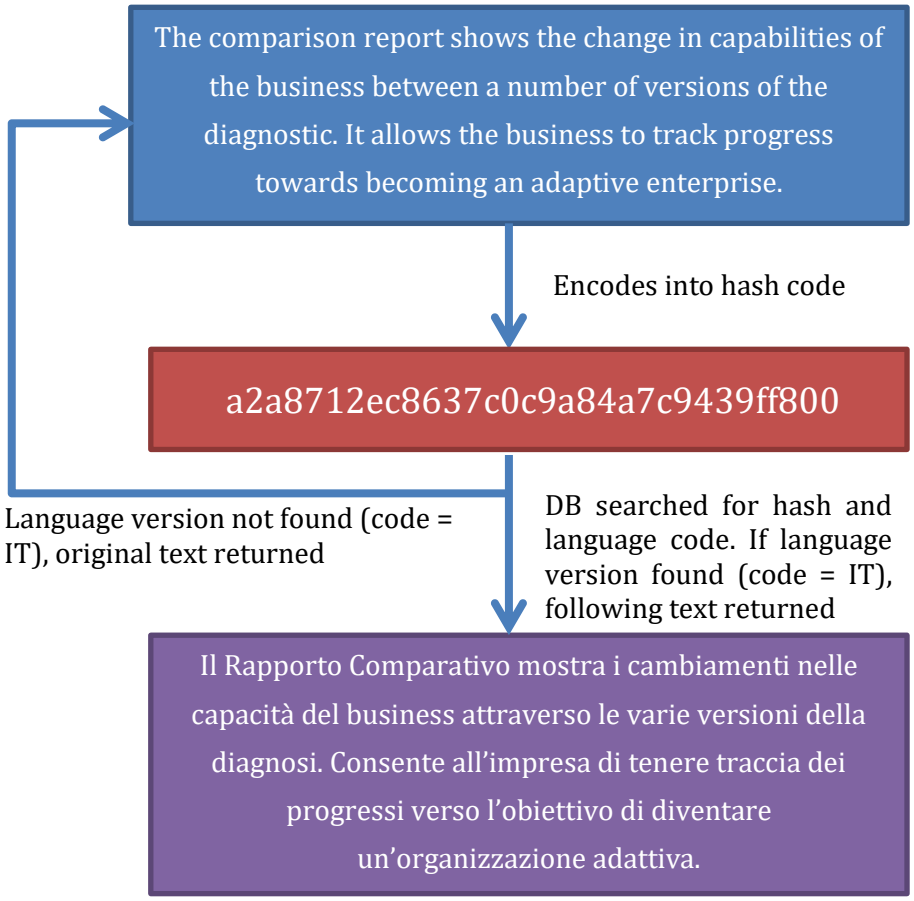

FIGURE 26 – LANGUAGE STRINGS TRANSLATION PRINCIPLE

8.6 ADMINISTRATION

Administration is a separate part of the CD for editing currently existing data but also for the creation of totally new diagnostic templates – the defining of questions, responses, results including logic conditions etc. New versions of a diagnostic can be created by copying existing responses or the results from previous versions. Users do not have to answer same question in new versions. All text fields can be defined using WYSIWYG editor for text highlighting or adding a link redirecting a user to advisory/interesting sites within the portal on somewhere else on the web.

FIGURE 27 – ADMINISTRATION

9 CONTRIBUTION

This research work was focused on research of current state in decision making systems working in environments dealing with uncertainty type of data acquired from production processes, research of methodology used for designing a system capable to process standard as well as non-measurable data considering uncertainty and tacit knowledge, design of a general diagnostic expert system for processing input data and output data generation with a separated functional environment and a diagnostic model part, with consideration of the rules for a model template to be able to be implemented, modelling an instance of the diagnostic system with designed data acquisition – a rule based data processing triggered by a user, and a report generation design, development of a server-client solution with web presentation and its pilot implementation, live operation – testing with end-users in practice, improving the system on the basis of testing and reviews.

In fact, the research work itself and very close collaboration with companies with immediate step-by-step testing and implementation started the transformation process of today's production and service SMEs and business companies, which is now leading them towards the Industry 4.0. age.

This standard includes:

- **Interoperability** – ability to communicate across all company (people, all components of intelligent production company via IoT (Internet of Things) and IoS (Internet of Services),
- **Virtualisation** – ability to interconnection of all physical systems with virtual model and simulation tools,
- **Decentralisation** – decision making and control processes are performed in parallel and separated in autonomic subsystems,
- **Ability to work in real time** – real time processing is a key precondition for any communication, decision making and control in real system,
- **Orientation on services** – preference of computational philosophy of offers and using standard services – SOA architecture (Service Oriented Architectures),
- **Modularity and re-configurability** – Industry 4.0 systems should be maximally modular and autonomously reconfigurable based on automatic recognition and prediction.

It was obvious from the very beginning of the work, that the differences in digitalization, distribution of competences, followed by monitoring and measurement, evaluation and validation of measured results with data processing will be of a constrain, and the proposed methodology will have to work with those differences. The transformation process started with the analysis of the abilities for preparedness and digital maturity of small and medium

companies and their staff, managers and each individual employee. The research had to consider the infrastructure and architecture of hardware equipment and software means and brought new methodologies to the current state and proposal for upgrading them, followed by knowledge provided to help with the transformation steps.

Capability Diagnostic system was developed as a tool for transformation process that leads an SME from current state analysis through advisory on how to improve company processes to continuous monitoring of KPIs that display key indicator values and help the SME to have comprehensive overview of company's life and allow keeping a tight rein on company's processes. Final CD system is developed as simple dynamic expert system that allows using various diagnostic models, various computational methods as well as customisable reporting module. These features of the system helped very fast improvements and testing of the system and allowed almost immediate changes in the diagnostic model. This also means that a designer who knows how to work with the system is allowed to create his own module (instructions, questions, possible answers, options, report etc.).

CD system was implemented in FutureSME project and used by many SMEs across EU. Users were instructed on how to work with the system and mainly guided by facilitator during first diagnostic instance. Next round they already passed themselves. All project members (SMEs) were asked about tools and asked to review them during and at the end of the project. CD was continuously improved according to their reviews and CD was finally considered as the most contributing application for their SMEs in the FutureSME project. CD application is still (2017) accessible via FutureSME portal (www.futuresme.eu) and anybody is allowed to pass through the diagnostic process (after registration) or use any other tools from the portal for their company's process improvement. And also general diagnostic system as a shell is available for creating new diagnostic applications or systems for process improvement support.

10 PŘÍNOS DISERTAČNÍ PRÁCE

Tato disertační práce je zaměřena na průzkum aktuálního stavu rozhodovacích systémů pracujících v prostředí s neurčitými daty získanými z výrobních procesů, průzkum metodologie použité pro návrh systému schopného zpracovat kromě standardních dat také data neměřitelná či neurčitá, návrh obecného diagnostického systému se znalostní bází nebo systému pro zpracování vstupních dat a generování výstupu s modulárním oddělením výkonové části a modelových dat s ohledem na možnosti implementace pravidel pro datový model, modelování instance diagnostického systému se získáváním navržených dat – pravidlové zpracování dat řízené uživatelem, generování reportu, vývoj webového server-klient řešení a jeho pilotní implementaci, nasazení do ostrého provozu, testování s konečnými uživateli, vylepšování systému na základě testování a zpětné vazby.

Ve skutečnosti výzkumná práce, úzká spolupráce s podniky, implementace a testování započalo transformační proces nynějších výrobních, obchodních či služby poskytujících podniků, který dnes vede směrem ke standardu Průmyslu 4.0.

Mezi parametry tohoto standardu patří:

- **Interoperabilita** – schopnost lidí a všech komponent inteligentního výrobního podniku spolu komunikovat prostřednictvím IoT (internetu věcí) a IoS (internetu služeb).
- **Virtualizace** – schopnost propojování fyzických systémů s virtuálními modely a simulačními nástroji.
- **Decentralizace** – rozhodování a řízení probíhá paralelně a autonomně v jednotlivých subsystémech.
- **Schopnost pracovat v reálném čase** – dodržení požadavku reálného času je klíčovou podmínkou pro libovolnou komunikaci, rozhodování a řízení v systémech reálného světa.
- **Orientace na služby** – preference výpočetní filosofie nabízení a využívání standardních služeb – architektury typu SOA (Service Oriented Architectures).
- **Modularita a rekonfugurabilita** – systémy Průmyslu 4.0 by měly být maximálně modulární a schopny autonomní rekonfigurace na základě automatického rozpoznání a predikce situace.

Od samého začátku práce bylo zřejmé, že rozdíly v digitalizaci, distribuci kompetencí následovanou monitorováním a měřením, hodnocením a validací naměřených výsledků se zpracováním dat bude omezením a navržená metodologie bude muset pracovat s těmito rozdíly. Transformační proces započal analýzou schopností, připravenosti a digitální vyzrálosti malých a středních podniků a každého jejich zaměstnance. Výzkum musel zvážit infrastrukturu a architekturu hardwarového vybavení a softwarových prostředků a přinesl jak nové

metodologie, tak navrhl rozvoj těch aktuálních následován znalostí poskytnutou pro jednotlivé kroky transformačního procesu.

Capability Diagnostic byl vyvinut jako nástroj pro transformační proces vedoucí malý či střední podnik z aktuálního stavu pomocí poradenství (jak procesy v podniku zlepšit) ke kontinuálnímu sledování kritických parametrů (KPI). Monitorováním klíčových hodnot získává vedení podniku komplexní přehled o dění a umožňuje jim držet otěže podniku pevně v rukou. Finální CD systém je vyvinut jako jednoduchý dynamický expertní systém, který umožňuje použití různých diagnostických modelů, výpočetních metod i uživatelský definovatelných výstupních reportů. Tyto vlastnosti zajišťují možnost velmi rychlého provádění změn v diagnostickém modelu, ladění a testování. Zároveň designer, který umí se systémem pracovat, je schopen vytvářet zcela nové vlastní modely (instrukce, otázky, možné odpovědi, možnosti, reporty atd.)

CD systém byl implementován v projektu FutureSME a použit řadou malých a středních podniků napříč EU. Uživatelé byli instruování, jak s aplikací pracovat a většinou také během první diagnostiky vedeni poradcem. Další kola diagnostiky již prováděli samostatně. Všechny zúčastněné podniky byly během a v závěru projektu dotazování na vyvinuté nástroje a vyzývání k jejich recenzování. CD byla opakovaně vylepšována na základě zpětné vazby a v závěru projektu byla hodnocena jako jedna z nejpřínosnějších aplikací. Capability Diagnostic je stále (2017) přístupný prostřednictvím portálu FutureSME (www.futuresme.eu) a kdokoli je oprávněn (po registraci) projít diagnostickým procesem nebo využít další nástroje na portálu vyvinuté pro zlepšování firemních procesů. Taktéž obecný diagnostický systém je využitelný pro vytváření nových diagnostických modelů nebo systémů pro zlepšování technologických procesů.

11 SUMMARY

This dissertation is aimed at designing a diagnostic expert system as a tool for technology improvement support. Initial research was focused on current diagnostic systems, their target fields of use (machine decreased performance in industry, bug localization in information systems or diagnostic systems in healthcare etc.) and their computing (decision making) methods i.e. probability theory, rule based systems, multi-criterial decision making or neural networks. Methodology used for design is based on expert system principles including internal processing computational techniques, where genetic algorithms and semantic web are also mentioned. Although there are several diagnostic systems, all of them are used in a specific case and/or with one processing method. My first idea of designing a flexible modular diagnostic expert system was when I became a member of a FutureSME project team where I was assigned to develop a system for Capability Diagnostic – a tool for diagnosing small and medium companies.

All FutureSME project goals were focused on SMEs, lifeblood of European business, as they are critical to the European economy and will always exist however SMEs also live in an environment of continuing change. This change is accelerating and standing still is not a real option.

Some of the drivers of change are:

- Competitive disadvantages in comparison to emerging economies such as higher wages, greater regulation, greater social responsibilities, working conditions
- The nature of work itself is changing as a result of technological advances
- Workers expectation for a more fulfilling work/life balance
- Demographics – age profile of population
- Legislation - environmental and work directives
- Recession requiring new approached to conducting business
- New technologies and social media platforms are driving an unprecedented reorganisation of how value is generated
- Changes to customer needs

In order for European SMEs to prosper in the future, these challenges need to be taken into account so they can survive and prosper. Productivity and Management tools and methodologies have been developed down through the ages help organisations adapt to change and to bring efficiency and productivity to industry. By in large they are geared at large organisations who have the resources to train their people and implement these techniques. Large organisations also by their nature have dedicated resources to manage the critical disciplines of a business. By contrast, SMEs owner/managers are resource constrained and time

constrained, responsible for multiple disciplines, trying to balance several things simultaneously and usually do not have the luxury to devote to developing their organisation's capabilities.

The goal of the FutureSME project was to develop tools and methodologies that help the SME to develop their capabilities to survive and prosper in an ever changing environment by providing the tools and methodologies that are appropriate and easy to implement within their resource constraints.

The project developed a methodology for SMEs to develop their organisational capabilities to help them sustain their profitability and growth in an uncertain and increasingly competitive environment. The methodology is known as the futureSME Transformation Process. The futureSME Transformation Process is fundamentally about getting the different levels of an organisation to Think, Act and Reflect together and build its capabilities to enable it to grow and sustain itself into the future. It is not a once-off process. It is a continuous process and should be repeated on a continuous basis. Every time the process is repeated it gets embedded further and deeper into the organisation until it becomes part of the culture of that organisation.
In summary the steps involved in the process are:

1. Complete the Full Capability Diagnostic to objectively assess your business capabilities and identify areas that require improvement.
2. Develop your business Strategy to articulate the Vision of the organisation and develop Strategy to achieve this Vision. Think together
3. Plan and Implement at the Operational level to achieve the strategy. Act together.
4. Review and Communicate what you have learned as an organisation, communication should be 2 way. Reflect together.

As a member of FutureSME team I designed and developed Capability Diagnostic - the most significant tool for transformation process. This expert system substitutes a human expert (facilitator) who analyses SMEs, processes its data and proposes improvements, creates action plan and supervises its fulfilment. Expert system appeared to be the best solution according to these reasons (and project goals) as well as minimal expenses in comparison with a human expert, while being able to work without human factor influence (subjectivity, fallibility, restricted working time, tiredness etc.) Based on these requirements I analysed process performed by human expert and transformed it into expert system – designed its structure and functionality. The process characteristic led me to a rule base system, thus I encoded all the rules into internal pseudocode and designed my original algorithm for input data processing and output data generation. As output data need presentation, a report and action plan were designed and developed.

Desired system had to be designed to support diagnostic model designers in testing and tuning the model based on FutureSME transformation process as both of these were also developed

during the project and there had been just an overall structure and functionality idea at the beginning of the project. The system was designed according to expert system principles with focus on modular structure and maximal extensibility without future interference with finished code structures. All computing algorithms, data processing, transformation into pseudocode and final development into web application are results of my own original design and development. Computing algorithm was designed regarding maximal flexibility which offers diagnostic model creating and tuning only with pseudocode knowledge but without exact programming language skills. The final system is a result of continuous research, design and development leading to robust diagnostic system approved by over 70 SME EU representatives. System was updated during the project (3 versions) and 154 diagnostic iterations were performed while some users repeatedly diagnosed their companies to get their progress in time. System is still available at www.futuresme.eu.

Capability Diagnostic helps SMEs to improve and optimise their technology processes. However, it's a first instance of the system used to solve real problem and there is a wide scale of prospective exploitation. Furthermore, there is a valuable attribute – a possibility to use the system for model design, development, testing as well as approving custom computational methods, their comparison and evaluation in particular cases. This environment is efficient solution for technology optimisation.

The key features are:

- **General solution** – can be used across all industry branches as well as social, healthcare, pedagogical or academic sectors.
- **Model vs knowledge-based processing** – the system supports processing data comparing to model (rules) as well as post-processing using algorithms over collected data across all users as a knowledge base.
- **Dynamic (modular) structure** – both models and processing techniques are separated and customisable
- **Flexibility and accessibility –** system allows custom diagnostic model creation only with pseudocode knowledge.
- **Efficiency/financial aspects** – a web solution, thus an end user does not need any special equipment except for computer with internet (or intranet) access and web browser. As far as server side is concerned, besides the LAN server there is no problem to operate the system using simple webhosting services with server-side scripting and database support.
- **Multilingual support –** system allows to have one diagnostic model and logic in several language translations

The dissertation focuses on feasibility of general solution – dynamic work with models and algorithms with emphasis on modern methods. The final application is simple for beginners

but powerful for advanced users and designers and can be applied in various situations in both model and knowledge based modes. Some of these ideas have been already published such as generating KPIs using genetic algorithm or application of financial models for SMEs.

Future improvements of the system should be focused on real time processing – automatic data retrieving instead of user interface driven way. After these improvements the SME would not need any diagnostic-improvement iterations and the system would work in continuous process analysing current values and displaying crucial values (i.e. KPIs or warnings regarding current company's state). System is already prepared for this upgrade thanks to modular architecture. This upgrade would add value sparing user's time spent by step-by-step entering values. This solution would require individual bridge (connection) to company's ERP system or database which will be specific for every single SME. Further future development could lead to creation of a designer, which would procure rules creation without any knowledge of pseudocode. I hope this Capability Diagnostic system will help many SMEs in better competition on the market and also the general system (shell) will be used in Industry 4.0 standard mainly in cases when simple diagnostic with real time processing and output is required.

12 ZÁVĚR

Tato disertační práce je zaměřena na návrh diagnostického expertního systému jako nástroje pro podporu zlepšování procesů. Nejprve byl popsán aktuální stav v oblasti diagnostických systémů především z pohledu účelu, ke kterému se používají jako je diagnostika poklesu výkonu strojů v průmyslu, vyhledávání chyb v informačních systémech či stanovení diagnóz v lékařství, a taktéž z pohledu interních výpočetních (rozhodovacích) metod jako je teorie pravděpodobnosti, pravidlový systém, multikriteriální hodnocení či neuronové sítě. Metodologie použitá pro návrh se zabývá principy stavby expertních systémů včetně podstaty interních výpočetních algoritmů, kde jsou zmíněny také genetické algoritmy či sémantický web. Ačkoli existuje celá řada diagnostických systémů, mnoho z nich je použito ve specifických situacích použitím jedné výpočetní metody. Poprvé mě napadlo navrhnout flexibilní modulární diagnostický expertní systém v době, kdy jsem se stal členem výzkumného a vývojového týmu Vysoké školy báňské – Technické university Ostrava v projektu FutureSME. V tomto projektu jsem měl za úkol navrhnout a vyvinout Capability Diagnostic – nástroj pro diagnostiku malých a středních podniků.

Všechny cíle projektu FutureSME byly zaměřeny na malé a střední podniky jako hybnou sílu evropského obchodu, neboť tyto jsou klíčové pro evropskou ekonomiku, budou stále existovat, zatímco se musí neustále vypořádávat s kontinuálními změnami. Tyto změny jsou stále rychlejší a nelze jen čekat.

Mezi hlavní důvody patří:

- Konkurenční nevýhody v porovnání s rozvojovými ekonomikami jako jsou vyšší daně, vyšší regulace, vyšší sociální zodpovědnost, pracovní podmínky
- Vlastní kultura práce se mění jako výsledek technologických pokroků
- Očekávání pracovníků ve smyslu naplňující práce a životní rovnováhy
- Demografie – věkový profil populace
- Legislativa – direktivy životního a pracovního prostředí
- Recese vyžadující nové přístupy k podnikatelské činnosti
- Nové technologie a sociální média diktující reorganizaci a způsob generování hodnot
- Změny dle požadavků zákazníka

Aby bylo možné zajistit přežití a prosperitu malých a středních podniků v budoucnu, je nutno vzít tyto výzvy v úvahu. V průběhu let již byly vyvinuty metodologie a nástroje pro zvýšení produktivity a zlepšení organizace podniků. Většina je však zaměřena na velké korporace, které mají dostatek zdrojů pro proškolení zaměstnanců či zavedení těchto metod. Velké podniky také mají vymezené prostředky pro tyto kritické oblasti. Naopak manažeři/majitelé malých a středních podniků mají omezené prostředky jak finanční tak časové, jsou zodpovědní hned za

několik oblastí snažíce se efektivně rozložit síly. Nemohou si tedy dovolit ten luxus zaměřit se na rozvoj organizačních schopností podniku.

Cílem projektu FutureSME bylo vyvinout nástroje a metodologie, které by pomohly malým a středním podnikům rozvíjet jejich schopnosti, přežít a prosperovat v měnícím se obchodním prostředí tím, že jim nabídne nástroje a metodologie, přiměřené a jednoduché pro implementaci zohledňující jejich finanční možnosti.

Projekt vyvinul metodologii pro malé a střední podniky pro rozvoj organizačních schopností, která jim pomáhá udržet ziskovost a růst v neurčitém a stále konkurenčnějším prostředí. Metodologie je známa jako transformační proces FutureSME. Základem transformačního procesu je zvýšení úrovně organizace podniku pomocí společného myšlení, konání a reflexe, budování schopností k zajištění udržitelnosti a rozvoje v budoucnu. Tento proces, který je kontinuální a měl by se provádět v daných intervalech, popisují tyto kroky:

1. Provedení kompletní diagnostiky podniku vložením objektivních údajů, aby byly identifikovány oblasti, který vyžadují zlepšení.
2. Vypracování společné strategie vyjadřující vize organizace.
3. Plánování a implementace operační úrovně k dosažení strategie.
4. Revize a komunikace provedených změn, společné zhodnocení.

V rámci projektu FutureSME jsem jako nejvýznamnější nástroj transformačního procesu navrhl a vyvinul expertní systém nazvaný Capability Diagnostic. Ten nahrazuje lidského experta (poradce), který provádí analýzu procesů podniku, následně data vyhodnocuje, navrhuje nápravná opatření, stanovuje akční plán a dohlíží na jeho plnění. Expertní systém se z výše uvedených důvodů (a cílů projektu) jevil jako vhodné řešení především z důvodu minimálních finančních nároků v porovnání s lidským expertem, zatímco je schopen pracovat bez vlivů lidských faktorů (subjektivita, omylnost, omezená pracovní doba, únava apod.). Na základě tohoto požadavku jsem analyzoval proces prováděný expertem, ten následně zpracoval do podoby expertního systému, navrhl jeho strukturu a funkcionalitu. Vzhledem k tomu, že charakteru procesu nejlépe vyhovovala podoba pravidlového expertního systému, formuloval jsem jednotlivá pravidla do podoby interního pseudokódu a navrhl svůj vlastní originální algoritmus, který na základě pravidel zpracovává vložená vstupní data do podoby výsledných dat. Pro výstupní data jsem následně navrhl způsob prezentace a to především v podobě výstupního reportu, dále pak jako tzv. akční plán.

Cílový systém byl vyvinut tak, aby umožnil průběžné testování a ladění diagnostického modelu založeného na zmíněném FutureSME transformačním procesu, neboť ten byl taktéž předmětem vývoje a na začátku mé práce byla známa pouze jeho nepříliš detailní struktura a funkčnost. Systém jsem navrhl dle zásad tvorby expertních systémů, avšak s ohledem na modulární strukturu a maximální možnost budoucího rozšíření bez potřeby zásahu do stávajících struktur.

Veškeré výpočetní algoritmy, zpracování dat, převod požadavků ve formě instrukcí či pseudokódu a jejich následná realizace v podobě vyvinuté aplikace jsou výsledkem mého vlastního návrhu a vývoje. Výpočetní algoritmus jsem navrhl s ohledem na maximální flexibilitu, která uživateli, který je schopen na základě znalosti pseudokódu, ale bez znalosti programování (syntaxe realizačního programovacího jazyka), umožní tvorbu a ladění diagnostického modelu. Finální verze diagnostického nástroje je výsledkem mého kontinuálního výzkumu, návrhu a vývoje, který vedl k realizaci robustního diagnostického systému. Ten ověřilo přes 70 zástupců malých a středních podniků napříč EU. Systém postupně prošel aktualizacemi (3 verzemi) a celkem bylo dokončeno 154 diagnostických iterací, kdy někteří uživatelé prováděli diagnostiku svého podniku opakovaně, aby zjistili, jak se firemní procesy vyvíjejí v čase. Systém je stále přístupný a funkční na webu projektu www. futuresme.eu.

Capability Diagnostic pomáhá malým a středním podnikům zlepšovat a optimalizovat technologické procesy. Tato konkrétní aplikace je první variantou navrženého diagnostického expertního systému pro řešení reálné problematiky, systém však nabízí širokou škálu potenciálního využití. Navíc je zde hodnotný parametr – možnost použít systém pro návrh, vývoj a testování diagnostického modelu stejně jako ověřování nových (uživatelem vytvořených) matematických výpočetních metod (modulů), jejich porovnávání a ověřování na konkrétních případech. Jedná se o efektivní řešení optimalizace technologických procesů.

Klíčové vlastnosti jsou:

- **Obecné řešení** – může být použit napříč průmyslovými oblastmi a mimo to také např. v sociálním, zdravotním či akademickém sektoru.
- **Model x znalostní báze** – systém umožňuje zpracování dat použitím modelu (např. pravidlového systému) nebo použitím algoritmů nad databázovými daty jako znalostní bází.
- **Dynamická (modulární) struktura** – modelová data i výpočetní algoritmy jsou logicky odděleny a je možno jimi systém rozšiřovat.
- **Flexibilita a přístupnost –** systém umožňuje vytváření diagnostických modelů pouze na základě znalosti pseudokódu.
- **Efektivnost/finanční nároky** – webové řešení vyžaduje od klienta pouze webový prohlížeč a internetové (intranetové) připojení. Vezmeme-li v úvahu provozovatele serveru, je potřeba pouze základních webhostingových služeb (LAMP).
- **Multijazyčnost** – systém umožňuje mít jeden diagnostický model v několika jazykových verzích.

Disertační práce se zaměřuje na proveditelnost obecného řešení – dynamickou práci s modely a algoritmy s ohledem na využití moderních metod. Finální aplikace je jednoduchá pro začátečníky, ale výkonná pro pokročilé uživatele a designery, a může být použita v různých

situacích. Některé nápady již byly publikovány jako např. generování KPI pomocí genetických algoritmů nebo aplikace finančních modelů pro malé a střední podniky.

Budoucí rozvoj systému by měl být zaměřen na zpracování v reálném čase – automatické stahování a zpracování dat namísto uživatelsky řízeného. Po těchto úpravách by nebylo nutné např. provádět opakované iterace diagnostiky-zlepšení a systém by mohl pracovat v kontinuálním režimu analyzováním aktuálních a zobrazováním kritických hodnot (KPI) nebo varování s ohledem na aktuální stav podniku. Systém je již připraven pro tuto úpravu díky modulární architektuře. Tento upgrade by zvýšil hodnotu aplikace především úsporou času uživatele stráveným opakovaným vkládáním vstupních hodnot do systému. Toto řešení by vyžadovalo pro každý podnik vytvoření individuálního můstku mezi diagnostickým systémem a firemním ERP systémem či jeho databází. Další vývoj je možno také směřovat v oblasti návrhu diagnostických modelů, kdy by bylo vhodné vyvinout tzv. návrhář, který by umožnil vytvářet pravidla diagnostického modelu bez znalosti interního pseudokódu. Věřím, že Capability Diagnostic pomůže zvýšit konkurenceschopnost malých a střeních podniků a také že vyvinutý diagnostický expertní systém bude v budoucnu použit ve standardu Průmyslu 4.0 především v případech, kdy je vyžadována jednoduchá diagnostika se zpracováním v reálném čase.

BIBLIOGRAPHY

ABDULLAH, U., M.J. SAWAR and A. AHMED, 2009. Design of a Rule Based System Using Structured Query Language. In: *Eighth IEEE International Conference on Autonomic and Secure Computing*. Chengdu: IEEE, pp. 223 - 228.

About Perl, 2016 [online]. Available at: perl.org

ALESHUNAS, J., 2016. *Managing Uncertainty in Rule-Based Systems*. Available at: http:// mercury.webster.edu/aleshunas/CSIS%205420/CSIS%205420/ Instructor%20Materials/ Chapter%2013.pdf%2FInstructor%2520Materials%2FChapter%252013.pdf&usg

ANDERSEN, P., 2005. In: *Business Diagnostic Tools* [online].2005. Available at: http:// www.printnet.com.au/business-diagnostics-landing.html

Artificial Intelligence Blog, 2010. *Nonmonotonic Reasoning* [online]. Available at: http:// artificialintelligence-notes.blogspot.cz/2010/07/nonmonotonic-reasoning.html

BELOHLAVEK, D., 2012. *Expert Systems based on Ontogenetic Maps*. Blue Eagle Group.

BIH, J., 2006. In: *Paradigm Shift: An Introduction to Fuzzy Logic* [online].2006. Available at: http:/ /www.cse.unr.edu/~bebis/CS365/Papers/FuzzyLogic.pdf

BITITCI, U. and A. ATES, 2009. The appropriateness of current intervention policy patterns and delivery mechanisms to address the …. In: *In Configuring manufacturing value chains - Responding to an uncertain world - 14th Cambridge Symposium on International Manufacturing*. Cambridge, pp. 11. Available at: www.ifm.eng.cam.ac.uk/cim/ symposium2009/…/20_umit_bititci.pdf

BITITCI, U. S. and C. MAGUIRE, 2009. An Investigation of Productivity And Competitiveness Challenges Facing European SMEs. In: *Competitive Paper. Strathclyde Institute for Operations Management, University of Strathclyde*. Glasgow, UK, pp. 23.

BOOCH, G., 1991. *Object-Oriented Design With Applications*. Menlo Park, Califormia: Bejjamin/Cummings.

BRACHMAN, R., 1985. Introduction. In: *Readings in Knowledge Representation*.

BUCHANAN, W., 2003. Mastering Delphi Programming. Palgrave Macmillan, pp. 10-. ISBN 978-1-137-17356-0.

Business Diagnostics, 2004 [online]. Available at: http://www.businessdiagnostics.com.au/

CHANG, R. I., 2005. Disease Diagnosis Using Query-Based Neural Networks. *Advances in Neural Networks – ISNN 2005*. Lecture Notes in Computer Science, vol. MMMCDXCVIII, pp. 767-73.

CHANTLER, M. J. et al., 1995. A Methodology for the development of model-based diagnostic systems. In: *IEE Colloquium on Real-Time Knowledge Based Systems*. London,UK: IEEE, pp. 1-3.

Codeguru, 2004. *The.NET Architecture* [online] [cit. 2016]. Available at: http:// www.codeguru.com/csharp/sample_chapter/article.php/c8245/The-NET-Architecture.htm#page-3

COHEN, M., I. FERNANDO and F. HENSKENS, 2011. A Domain Specific Expert System Model for Diagnostic Consultation in Psychiatry. In: *12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. Sydney, Australia: University of Technology, pp. 3-6.

DELANEY, K., 2006. Inside Microsoft SQL Server 2005: The Storage Engine. Microsoft Press. ISBN 0-7356-2105-5.

DigitalOcean, 2014. *A Comparison Of Relational Database Management Systems* [online]. Available at: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems

EHRGOTT, M. and X. GANDIBLEUX, 2003. Multiobjective Combinatorial Optimization - Theory, Methodology, and Applications. In: *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Dordecht: Kluwer Academic Publishers, pp. 369-444.

ERICSSON, H. and C. M., 2012. *Design and development guidelines for manufacturing at Volvo Cars*. Charlmers Univesity Of Technology.

FELLBAUM, C., 2010. *WordNet*. Dordrecht: Springer Netherlands.

Four major principles of Object-Oriented Programming , 2012 [online]. Available at: http:// codebetter.com/raymondlewallen/2005/07/19/4-major-principles-of-object-oriented-programming/

FUTURESME, 2008. *Research proposal FP7 - Future Industrial Model for SMEs – FutureSME*. Glasgow: Strathclyde University, University of Strathclyde.

GADD, K. W., 1995. Business self-assessment: A strategic tool for building process robustness and achieving integrated …, **1** (3), a266-85.

GEOFFRION, A. M., J. S. DYER and A. FEINBERG, 1972. An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of …, **19** (4), a2357-368.

GOSLING, J. et al., 2014. In: *The Java Language Specification* [online].2014.

HAKES, C., 1999. *The Business Excellence Handbook*. London: BQC Performance Management.

HAYES-ROTH, F., 1984. Knowledge-Based Expert Systems. IEEE Computer, Issue 17, pp. 263-73.

HEYWOOD, M. I., 2012. In: *NPC Behaviours Using Rule Based Methods* [online]. Canada:2012. Available at: http://web.cs.dal.ca/~mheywood/CSCI3154/Notes/3154-RuleBased.pdf

ILIIF, E. C., 1997. In: *Computerized medical diagnostic system including re-enter function and sensitivity factors* [online]. La Jolla, USA:1997. Available at: http://www.google.cz/patents/US5594638?hl=cs&dq=diagnostic+system+in+industry

JACKSON, J., 2014. IT world. In: *PHP gets a formal specification, at last* [online]. 31. July. 2014. Available at: http://www.itworld.com/article/2697195/enterprise-software/php-gets-a-formal-specification--at-last.html

JACKSON, P., 1998. *Introduction To Expert Systems*. 3rd. Addison Wesley.

KAMBHAMPATI, S., 1996. Formalizing Dependency Directed Backtracking and Explanation Based Learning in Refinement Search. Arizoba State University [cit. 2015]. Available at: http://www.aaai.org/Papers/AAAI/1996/AAAI96-113.pdf

KAUSHIK, and P. CHAVAN, 2014. Agile FMEA. DYNAMIC POSITIONING CONFERENCE. Available at: https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwj3jo-msf_TAhWnd5oKHbQbBMEQFgg2MAI&url=http%3A%2F%2Fdynamic-positioning.com%2Fproceedings%2Fdp2014%2FBackup_reliability_kaushik.pdf&usg=AFQjCNFmTUsuhiGPhIM8fMSjpBeBNMrXPQ&sig2=Ec4_Y1

KHEDIVE, E., 2013. *When and why do we need data normalization?*. ResearchGate.net. Available at: https://www.researchgate.net/post/When_and_why_do_we_need_data_normalization

KIDD, A. L., 2012. *Knowledge Acquisition for Expert Systems*. Springer Science & Business Media.

KIM, B. and G. S. MAY, 1997. Real-Time Diagnosis of Semiconductor Manufacturing Equipment Using a Hybrid Neural Network Expert …, **20** (1).

KUHLMAN, D., 2009. A Python Book: Beginning Python, Advanced Python, and Python Exercises.

LEVESQUE, H. and R. BRACHMAN, 1985. A Fundamental Tradeoff in Knowledge Representation and Reasoning. In: *Reading in Knowledge Representation*, pp. 41-70.

LINDSAY, R. K. et al., 1980. *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project*. McGraw-Hill Book Company.

LinuxMint.com, 2016. *MySQL Server Review* [online]. Available at: http:// community.linuxmint.com/software/view/mysql-server

MARCUS, S., 2013. *Automating Knowledge Acquisition for Expert Systems*. Springer Science & Business Media.

MCANDERSON, R., 2010. In: *On the Mark Marketing* [online].2010. Available at: http:// www.onthemark.com.au/business_diagnostic.htm

MCGINLEY, P., 2012. Decision analysis software survey, vol. XXXIX. Available at: http:// www.orms-today.org/surveys/das/das.html

Microsoft, 2016. *SQL Server 2008: Editions* [online]. Available at: https://www.microsoft.com/ en-us/sql-server/default.aspx

MILLER, G. A., 1995. WordNet: A Lexical Database for English, **38** (19), 3.

MISIASZEK, M., 2007. *Reasoning Under Uncertainty* [cit. 2015]. Available at: www4.cookman.edu/faculty/banisakher/chapter04_AI.ppt

MITCHELL, M., 1999. *An Introduction to Genetic Algorithms*. Fifth printing. Cambridge: A Bradford Book The MIT Press.

MULLINERA, E., K. SMALLBONE and V. MALIENE, 2013. An assessment of sustainable housing affordability using a multiple criteria decision making method, **41** (2), a2270-279.

MySQL 5.7 Reference Manual, 2016. *MySQL History* [online]. Available at: https:// dev.mysql.com/doc/refman/5.7/en/history.html

NORVIG, P. and S. THRUN, 2015. Intro to Artificial Intelligence. *Udacity* [online] [cit. 2015]. Available at: https://www.udacity.com/wiki/cs271/unit1-notes

OpenLearn, 2016. *Understanding operations management* [online]. Available at: http:// www.open.edu/openlearn/money-management/management/leadership-and-management/understanding-operations-management/content-section-3.4

Pcmag.com, 2013. *Conventional programming* [online].

PHP, 2016. *History of PHP* [online] [cit. 2016]. Available at: http://php.net/manual/en/history.php.php

PostgreSQL Documentation, 2016. *What is PostgreSQL?* [online]. Available at: https://www.postgresql.org/docs/current/static/intro-whatis.html

RapidBI, 2007. *Rapid Business Improvement* [online]. Available at: http://rapidbi.com/created/business_diagnostic_tools/

RITCHIE, L. and B.G. DALE, 2000. Self-assessment using the business excellence model: A study of practice and process, vol. LXVI, pp. 241-54.

SIOW, C. H. R., J. B. YANG and B.G. DALE, 2001. A new modelling framework for organisational self-assessment: Development and application.

Small Business Diagnostic Tool, 1992 [online]. Available at: http://www.accountingsoftwaresecrets.com/knowledge-base/add-ons/expert-analysis/diagnostic-tools/small-business-diagnostic-tool

SOVARONG, L. and C. J. SPANOS, 1997. A general equipment diagnostic system and its application on photolithographic sequences, **10** (3), a2329-343.

Techopedia, 2016. *Oracle Database* [online]. Available at: https://www.techopedia.com/definition/8711/oracle-database

Techwalla, 2015. *Advantages & Disadvantages of Oracle SQL* [online]. Available at: https://www.techwalla.com/articles/advantages-disadvantages-of-oracle-sql

Techwalla, 2016. *Advantages & Disadvantages of Microsoft SQL* [online]. Available at: https://www.techwalla.com/articles/advantages-disadvantages-of-microsoft-sql

The Building Small Business Human Resources (HR) Quiz, 2008 [online]. Available at: http://hrskills.smallbusinesscheckup.com/check-up.html

TIHHOMIROV, V. et al., 2013. Assessment of diagnostic test for automated bug localization. In: *14th IEEE Latin American Test Workshop (LATW)*. Cordoba, Argentina: IEEE, pp. 1-6.

TRIANTAPHYLLOU, E., 2000. *Multi-Criteria Decision Making: A Comparative Study*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 320 pp..

Tutorials' point, 2015. *Artificial Intelligence - Expert Systems* [online]. Available at: http://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm

WANG, Y. M., J.B. YANG and D.L. XU, 2006. Environmental Impact Assessment Using the Evidential Reasoning Approach.

WEISTROFFER, H. R., C. H. SMITH and S. C. NARULA, 2005. Multiple Criteria Decision Support Software. In: *Multiple Criteria Decision Analysis : State of the Art Surveys, Chapter 24*. New York: Springer.

WOODS, W. A., 1978. Semantics and quantification in natural language question answering, **17** (3).

YACHIKU, H., R. INOUE and T. KAWAI, 2009. Diagnostic Support Technology by Fusion of Model and Semantic Network. In: *Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009), 28-30 September …*. Athens: Springer, pp. 886-91.

ZADEH, L. , 1965. *Fuzzy sets*.

ZHANG, X., 2012. Model-based Design of Diagnostic System. In: *3rd International Conference on System Science, Engineering Design and Manufacturing Informatization*. Chengdu: IEEE, pp. 43-46.

ZIMMERMANN, H. J., 2012. *Fuzzy Sets, Decision Making, and Expert Systems*. Springer Science & Business Media.

ZLATARVA, N., 1992. Artificial Intelligence Review. In: *Truth Maintenance Systems and their Application for Verifying Expert System Knowledge Bases* [online].1992.

AUTOR'S PUBLICATIONS

PLANDOR, D., M. BABIUCH and R. FARANA, 2011. Using .NET technology for process control and monitoring, *In 12th International Carpathian Control Conference (ICCC),* Velké Karlovice, Czech Republic, 2011

PLANDOR, D. and L. LANDRYOVÁ, 2011. Business Diagnostic Tool, *In 6th International Conference on Software and Data Technologies*, Seville, Spain, submitted to BITA - Special Session on Business IT Alignment, Seville, Spain, 2011

PLANDOR, D. and L. LANDRYOVÁ, 2011. KPI determination based on company's self-assessment, 2011, *In International Conference on Advances in Production Management Systems (APMS)*, Stavanger, Norway, 2011

PLANDOR, D., M. BABIUCH and R. FARANA, 2011. Implementation of Process Control and Monitoring Software with Dynamic Compilation in .NET Technology, Acta Montanistica Slovaca, Košice, Slovak Republic, 2011, Vol. 16, Nr. 1, pp. 8-16. ISSN 1335-1788. IF v roce 2010 = 0.134 (JCR 2010). Available at: http://actamont.tuke.sk/ams2011.html

PLANDOR, D. and L. LANDRYOVÁ, 2012. Generating KPIs using genetic algorithms*, In 13th International Carpathian Control Conference (ICCC),* High Tatras, Podbanské, Slovakia, 2012

PLANDOR, D. and L. LANDRYOVÁ, 2012. Bankruptcy and Financial Standing Models Application for SMEs*, In The Seventh International Conference on Software Engineering Advances (ICSEA)*, 2012, 18-23 November 2012, Lisbon, Portugal, pp.161-164, 2012

IMPLEMENTED ENGINEER'S WORK

PLANDOR, D. and M. BABIUCH, 2010. Registered software in RIV 2010 - RIV/61989100:27230/10:10224995 – WinCoMet – software for process monitoring and control (2010)

PLANDOR, D. and J. TŮMA, 2010 Registered functional specimen in RIV/61989100:27230/10:86076037 – System for car operational parameters monitoring by mobile device (2010)

PLANDOR, D. and L. LANDRYOVÁ, 2011. Verified technology in RIV/61989100:27230/11:86080177 - Business diagnostic system for SMEs (2011)

PLANDOR, D. and L. LANDRYOVÁ, 2011. Registered software in RIV/61989100:27230/11:86079974 – Administration module for business diagnostic system (2011)

PLANDOR, D., L. LANDRYOVÁ and M. BABIUCH, 2011. Finance module for SMEs – Registered software in RIV. 2012. VŠB – Technical University of Ostrava, Nr. 032/13-12-2012_SW, CODN: 61989100, Available at http://www.352.vsb.cz/veda.htm.

PLANDOR, D., R. TICHÝ and L. LANDRYOVÁ, 2012 Verified technology in RIV – Nr. 053/13-12-2012_OT - Micropayment system for SMEs