ebbits

Enabling the business-based
Internet of Things and Services

(FP7 257852)

D3.4 Business framework for online OEEE applications for production and energy optimization

Published by the ebbits Consortium

Dissemination Level: PU = Public

Project co-funded by the European Commission within the 7th Framework Programme
Objective ICT-2009.1.3: Internet of Things and Enterprise environments

Document control page

Document file:	D3.4 Business framework for online OEEE applications for production and energy optimisation
Document version:	1.1
Document owner:	COMAU
Work package:	WP3 – Enterprise Frameworks for Life-cycle Management
Task:	T3.3 – Production and energy optimization
Deliverable type:	R = Report
Document status:	☒ approved by the document owner for internal review
	☒ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Pietro A. Cultrona (COMAU)	2011-05-09	TOC
0.2	Jozef Glova (TUK)	2011-09-22	Ch. 4.1 added
0.3	Antje Boehm-Peters (SAP)	2011-10-04	Ch. 4.2 added
0.4	Pietro A. Cultrona COMAU)	2012-01-17	Ch. 6.2 modified
0.5	Antje Boehm-Peters (SAP)	2012-02-02	Ch. 6.2 modified
0.6a	Jozef Glova (TUK)	2012-02-15	Ch. 6.3, 6.5 added, Ch. 3.1, 4.1 modified
0.6b	Gonzalo Alcaraz, Mauricio Caceres (ISMB)	2012-02-17	Ch. 6.4
0.7	Pietro A. Cultrona (COMAU)	2012-02-24	Contributions merging and integration, formatting
0.9	Pietro A. Cultrona (COMAU), Jozef Glova (TUK)	2012-02-27	Minor changes and executive summary
1.1	Pietro A. Cultrona (COMAU), Jozef Glova (TUK)	2012-03-14	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
Jonathan Simon (FIT)	07/03/2012	Approved pending revisions (enhancements Ch. 3, suggest removing redundant descriptions)
Peter Rosengren	29/02/2012	

Index:

1.   Executive summary

This Deliverable reports the work done in Task T3.3 "Production and Energy Optimization", that includes development of the usage scenario, the business framework and metric for an ebbits IoPTS application for production optimization with energy awareness using Overall Equipment and Energy Efficiency (OEEE) metrics.

Chapter 2 of this deliverable provides an overview of the steps involved in the definition of an automotive manufacturing system. This iterative production process involves phases such as Concept, Simulation and Optimization, Design/Engineering, Commissioning, and finally Operation. Starting from this process, in Chapter 3, the concept of Overall Equipment Efficiency (OEE) is described in detail, analysing Availability, Performance and Quality that are the three figures that compose this Key Performance Indicator (KPI). Then the concept of OEEE is introduced, envisaging the importance of the energy monitoring and the integration with the OEE KPI.

Chapter 4 provides a detailed description of generic optimization methodologies and models, especially focused on discrete event simulation. Arguments such as linear programming, integer programming, quadratic programming, branch and bound, genetic, algorithms and Lagrange's multipliers are thus taken into account. Basing on these methodologies, the subsequent section describes the commercial solutions and non-commercial solutions that are actually available with their main features and licensing features.

Following, Chapter 5 provides some investigations related to Automotive Energy Management state of the art and gives some figures of the car production environment.

Finally Chapter 6 briefly introduces the Automotive Manufacturing Use Case Scenario that will be father developed in the following phases of the ebbits project. In particular two scenarios have been taken into account considering the specific needs of an automotive production system provider like Comau.

2.   Introduction

2.1   Purpose, context and scope of this deliverable

The purpose of this deliverable is to provide a business framework for the online OEEE applications for production and energy optimization.

The development of an automotive production process is composed by several steps:

- The Concept, Design and Engineering phase, which consists in the initial development of the production lines.

- The Simulation and Optimization, done in order to improve the production line performances.

- The commissioning on the customers' side, which includes all the phases of the installation (mechanical, electrical, fluidics, etc.)

- Operation, that starts when the production line starts producing the elements and gives a feedback to the engineering in order to improve the line or the upcoming production lines.

The following diagram briefly summarizes this iterative process.
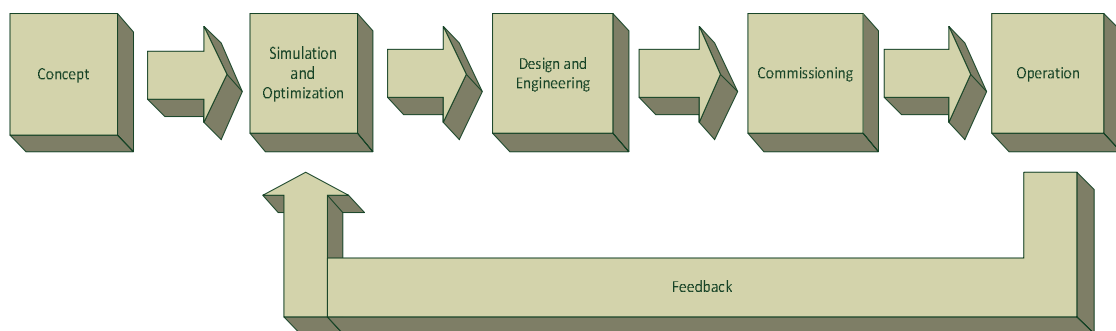
Figure 1: Production plant engineering and optimization

In the beginning, an initial concept is developed. Subsequently the process is modelled and simulation and optimization takes place. The initial inputs for this operation are typically composed by the main process parameters such as: Throughput, Cycle Time, Line Management Strategies (Kanban, Sequence, etc.), Mean Time Between Failures (MTBF), Mean Time To Recovery (MTTR) and some additional inputs like the plant space, OEE and specific customer requests.

Until today, this process optimization loop has been focused only on specific throughput and cost targets. An important step forward can be done introducing in this cycle the energy data and providing in this sense an output globally optimized including the energy side.

3.   OEE and OEEE Key Performance Indicators

According to European Commission (EC, 2007) manufacturing is still the driving force of the European economy, contributing over 6.500 billion euro in GDP and providing more than 30 million jobs. It covers more than 25 different industrial sectors, largely dominated by SMEs, and generates annually 1.500 billion euro of value added.

However, the average efficiency in manufacturing plants, measured on the so called Overall Equipment Efficiency (OEE) index, is only 60% (and a world-class rate is 85%). In practice of manufacturing plants, the generally accepted world-class goals for each factor are as follows 90% of Availability, 95% of Performance and 99,9% of Quality, whereby as already mentioned a composite OEE number is higher than 85% (Vorne, 2008).

The loss from inefficiencies in production could thus be as high as 1.000 billion euro annually. The OEE index is used in most manufacturing companies as a key metric in TPM (Total Productive Maintenance) and LEAN Manufacturing to provide a consistent way of measuring the effectiveness of the production.

Worldwide, industry consumes almost one-half of all the commercial energy used (IEA, 2012) and is responsible for roughly similar shares of greenhouse gases. An increasing number of manufacturing companies are facing a supply chain, which is demanding quantification of the carbon footprint and demands for future reductions. In order to stay competitive in the 21st Century; companies need to be include sustainability in their production optimisation schemes.

Hence, the aim of this part of deliverable is also to define a total optimisation metrics by adding energy to the OEE index, thus creating a new key index, the *OEEE (Overall Equipment and Energy Efficiency)*

The main scope of this chapter is the introduction of the OEE and OEEE Key Performance Indicators.

3.1   OEE Description

According (Vorne, 2011) OEE stands for Overall Equipment Effectiveness. Essentially, it is a single figure that signifies the utilisation of a machine. This can be at a job level, shift level or overall level.

Vorne Industries (Vorne, 2008) describes the OEE or Overall Equipment Effectiveness as a "best practices" way to monitor and improve the effectiveness of manufacturing processes, i.e. machines, manufacturing cells, assembly lines, etc. OEE is often used as a key performance indicator (KPI) in Total Productive Maintenance (TPM) and Lean Manufacturing programs and gives us a consistent way to measure the effectiveness of TPM and other initiatives by providing an overall framework for measuring production efficiency. OEE reduces complex production problems into simple, intuitive presentation of information.

Three main factors make up the OEE calculation. They are: Availability, Performance and Quality. These are expressed as a percentage and then the three factors are multiplied together to give you a single OEE figure – again expressed as a percentage.

The point of the final calculation is that it gives you a single figure to measure, and then compares, your OEE. Therefore you may, on a single machine perhaps, compare the OEE between jobs. This will allow you to see which jobs run well and which ones don't. You can then take corrective action.

You may compare shifts – and gain an insight to whether one shift performs better than another – again you can investigate the underlying reasons and take action to improve the OEE.

You may compare machines within, or across several plants. You may even compare different manufacturing plants where similar product are manufactured and understand underlying reasons why one may have a better OEE than another, and then take corrective action.

Ultimately you can, if the data is available, compare your OEE to that of your competitors or industry best and put plans in place to reach the best in class. Like best practice initiatives you may look at

industries that have similar characteristics to your own and then try to emulate practices that improve your OEE to the levels they sustain.

Total operating time		
A Loading time		not scheduled
B Running time	Idle time failure	
C Theoretical output		
D Actual output	speed loss minor stops	
E Actual output		
F Good output	scrap rework	

B/A = Availability   D/C = Performance   F/E = Quality
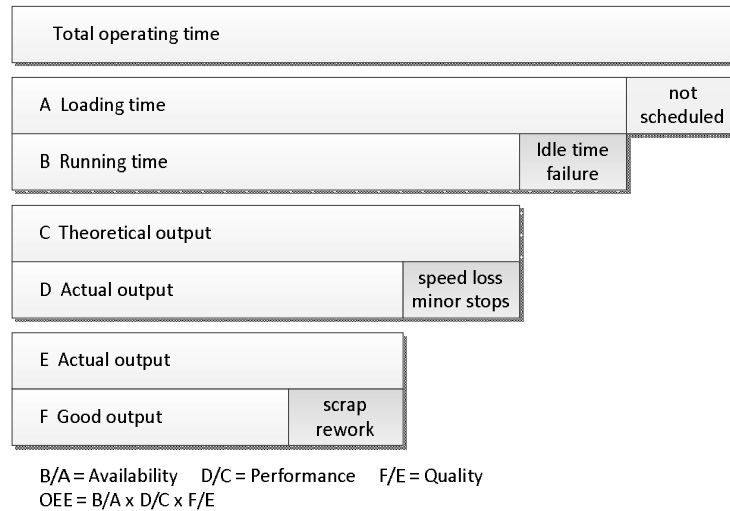OEE = B/A x D/C x F/E

Figure 2 Calculating OEE Index. Source: own.

Overall Equipment Effectiveness' analysis starts with definition of the amount of time a facility is open and available for equipment operation (Total Operating Time).

If we subtract a category of time called Planned Shut Down from Plant Operating Time, which includes all events that should be excluded from efficiency analysis because there was no intention of running production (e.g. breaks, lunch, scheduled maintenance, or periods where there is nothing to produce). The remaining available time is so called Planned Production Time.

Availability takes into account Down Time Loss, which includes any event that stop planned production for an appreciable length of time (usually several minutes – long enough to log as a trackable event). Examples include equipment failures, material shortages, and changeover time. Changeover time is included in OEE analysis, since it is a form of down time. While it may not be possible to eliminate changeover time, in most cases it can be reduced. The remaining available time is called Operating Time or actual production time (Vorne, 2008).

$$\frac{Actual\ production\ time}{Planned\ production\ time} = Availability$$

Performance takes into account Speed Loss, which includes any factors that cause the process to operate at less than the maximum possible speed, when running. Examples include machine wear, substandard materials, misfeeds, and operator inefficiency. It is the actual achieved output (expressed in elements produced) against the ideal output for the machine (expected production). Often the machine ideal or optimum output may be the figure published by the machine manufacturer. However, we all know that the ideal output may be affected by the situation of the machine, heat, cold, product running through etc. We would expect the ideal output to remain

constant and therefore variances may easily be identified. The remaining available time is called Net Operating Time (Vorne, 2008).

$$\frac{Actual\ output}{Theoretical\ output} = Performance$$

Quality is the final factor on the overall OEE calculation. This is simply a measure of good product or output divided by the actual output (total product for the job, shift, day, week, etc.). An element for discussion here is whether you include expected set up waste within the figures.

$$\frac{Good\ output}{Actual\ output} = Quality$$

OEE Index

We now simply multiply the figures together.

$$Availability \times Performance \times Quality = OEE$$

Overall Equipment Effectiveness (OEE) metrics are coming in handy during this recession. OEE is a set of measurements that tracks the availability, performance and quality output of plant equipment (see also Figure 3).

More and more, plants are turning to OEE to get increased efficiencies from their existing equipment. The rationale for measuring OEE is varied. Some seek to extend the life of existing equipment; others want to reduce costs during the recession. The overriding goal is to increase production without investing in more equipment or personnel. Recently, plants have turned to real-time OEE so they can make efficiency adjustments while the equipment is running. Some are adding a new dimension—energy consumption metrics—to the classic OEE measurements of equipment availability, performance and quality (AW, 2009).
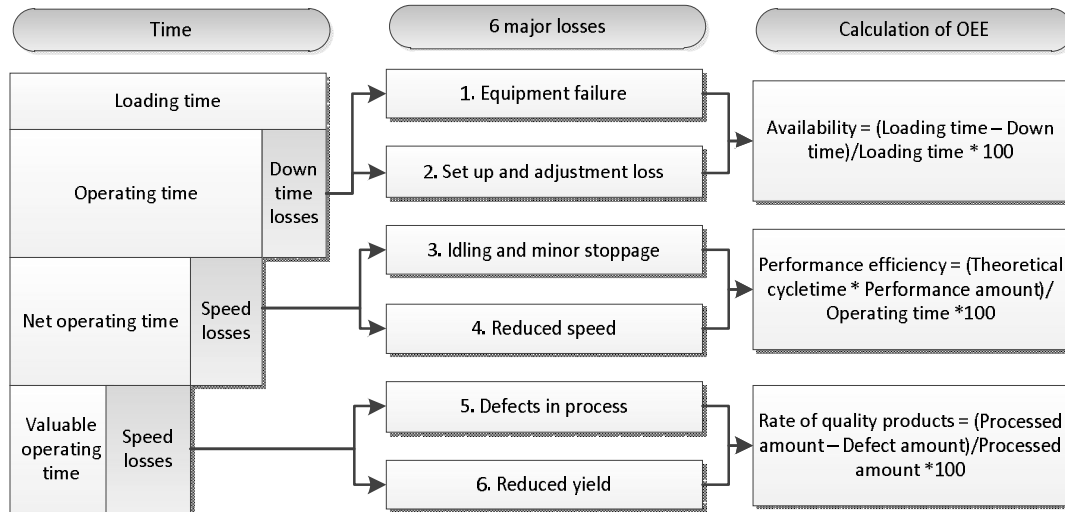
Figure 3: The OEE formulation. Source: own using descriptions in (Vorne, 2008)

3.2 OEEE Description

Manufacturing operations are among the most energy-intensive in the world. For instance in Germany industry is responsible for around 40% of the total energy consumption. Moreover industry energy use is responsible for almost 30% of all greenhouse gas emissions, which contribute to global climate change.

Energy management approaches and measurement and verification protocols are becoming more and more a significant issue in manufacturing environment. Nowadays energy data and information are not easily available and often they are disaggregated, thus their analysis and the consequent optimization strategies are very difficult to be implemented. In general when the energy managers try to implement energy monitoring applications, they find themselves drowning in the volume of data generated. Thus it becomes important to define and apply Key Performance Indicators (KPIs) to summarize volumes of data into a few critical "nuggets" of actionable information.

By combining the OEE KPI calculation with the energy management technologies it is possible to define a new KPI: the OEEE Overall Equipment and Energy Efficiency index.

This key performance indicator will be applied to get an effective tool for machinery management to measure or document improvements in energy or WAGES (Water, Air, Gas, Electric and Steam or in general energy consumption) usage.

In conclusion the above mentioned OEE will be extended with the energy consumption parameter to encompass energy management.

OEEE = OEE + E = Availability x Performance x Quality x Energy

where

OEE - overall equipment efficiency

4.   Optimization Models and Software

This chapter starts describing simulation and optimization strategies, as well as Discrete Event Simulation (DES). Following, an overview of model optimization software is presented, taking into account free and commercial applications.

4.1   Optimization Methodologies and Models

This part is organized as follows: it starts with brief overview on simulation, especially discrete event simulation. The general optimization methodologies and models like linear, integer and quadratic programming will be briefly discussed in rest of this part.

Many systems in areas such as manufacturing, supply chain management or financial management are too complex to be modelled analytically. DES has long been a useful tool for evaluating the performance of such systems. Discrete event simulation represents modelling, simulating and analysing systems utilizing computational and mathematical techniques, while creating a model construct of a conceptual framework that describes a system. The system is rather simulated by performing experiments using the computer implementation of the model and analysed to draw conclusions from output that assist in the decision making process. DES has long become one of the mainstream computer-aided decision-making tools due to availability of powerful computers. Figure 4 on the left, illustrates the ways of studying a system. Most often, the system is studied via experiment with the actual model, or experiment with a model of the actual system. On the right of Figure 4, the model taxonomy used in the simulation process utilizing either deterministic or stochastic models is illustrated. In classical thinking there are three types of simulation: discrete event, continuous, and Monte Carlo simulation. Meanwhile, in current thinking and work, these lines are becoming less distinct.

In particular discrete event simulation utilizes a mathematical or logical model of a physical system that portrays state changes at precise points in simulated time. Both the nature of the state change and the timestamp of this modification need a precise description. Customers waiting for service, the management of parts inventory or military combat are typical domains of discrete event simulation. Continuous simulation uses equational models, often of physical systems, which do not portray precise time and state relationships that result in discontinuities. The objective of studies using such models does not require the explicit representation of state and time relationships. Examples of such systems are found in ecological modelling, ballistic re-entry, or large-scale economic models. Monte Carlo simulation, the name given by John van Neumann and Stanislaw M. Ulam to reflect its gambling similarity, utilizes models of uncertainty where representation of time is unnecessary. The term originally attributed to "a situation in which a difficult non-probabilistic problem is solved through the invention of a stochastic process that satisfies the relations of the deterministic problem". A more recent characterization is that Monte Carlo is "the method of repetitive trials. Typical of Monte Carlo simulation is the approximation of a definite integral by circumscribing the region with a known geometric shape, then generating random points to estimate the area of the region through the proportion of points falling within the region boundaries (Albrecht, 2012).

According to Albrecht (2012), DES languages can be identified by six characteristics as follows:
- generation of random number to represent uncertainty,
- process transformers, to permit other than uniform random varieties to be used,
- list processing capability, so that object can be created, manipulated, and deleted,
- statistical analysis routines, to provide the descriptive summary of model behaviour,
- report generation, to provide the presentation of potentially large amount of data in an effective way for decision making, and
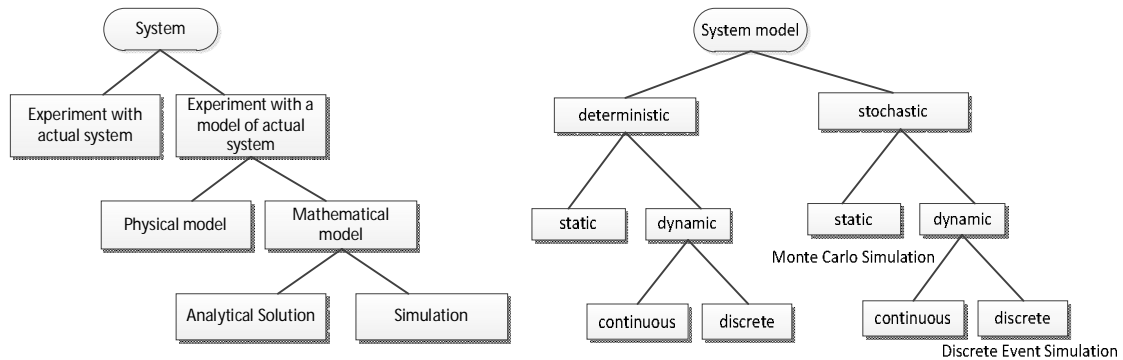- A timing executive or time flow mechanism.

Figure 4: Ways to study a system (left) and system model taxonomy (right). Source: own

In the computational science domain optimization means selection of the best alternative out of a set of all possible alternatives that are available (Rosinová, et al., 2007). In order to find an optimal solution to a real problem, we first need to find a suitable abstract representation of the problem, i.e. use mathematical language in order to define a mathematical model, which will suitably reflect the problem domain and enable us to look for optimal solution.

In this section a brief description of the most often used mathematical optimization models will be provided. The analysis will not be restricted to a particular type of real applications; rather it provides a broader overview of available mathematical optimization models, i.e. generic optimization models.

One very generic model, suitable to model many real world problems, is Constraint Satisfaction Problem (CSP), for detailed description see also (Tsang, 1993) or (Rossi, et al., 2006). Constraint satisfaction problem 0 is defined as a triple $\langle X, D, C \rangle$, where:

X is a set of variables $\{x_1, x_2, ..., x_n\}$,

D is a set of domains $\{D_1, D_2, ..., D_n\}$ – each variable x_i ($i = 1 ... n$), has defined a set of possible values, i.e. its domain ($x_i \in D_i$)

C is a set of constraints $\{C_1, C_2, ..., C_t\}$ (defined on subsets of variables from X),

The goal usually is to find (all or just one) such assignments of values from D to variables X that all constraints C are satisfied. But this basic definition of CSP does not speak anything about optimization. CSP is able to model many real world problems (e.g. in operations research, scheduling, supply chain management to name a few) and their feasible solutions. If we want to be able to differentiate, which solution is better than the other ones, we need to have defined besides $\langle X, D, C \rangle$ also an objective function (optimization criteria). In such a case the goal will be to find such a feasible solution, which minimizes (or maximizes) the given objective function (optimization criteria). Such type of problems is usually referred to as Constraint Optimization Problem (COP).

Traditionally, the authors assume that variables can take values from finite domains 0, but here we will not restrict ourselves to finite domains only. We think that the definition stated above is general enough to cover a large set of mathematical optimization models. For example:

1) If we restrict the domains D_i to real numbers only, the constraints to linear equality and linear inequality constraints, and the optimization criteria to linear objective function, we get traditional linear programming model (for more details see section 4.1.1 below). For linear programming problems polynomial algorithm exists already.

2) If we keep all the restrictions from the case 1) above, but restrict domains D_i to integers, we get integer programming (for more details see section 4.1.2 below). If the domains of variables are just two valued sets $\{0, 1\}$, we speak about binary integer programming. Both types of problems (integer programming as well as binary integer programming) are NP-hard.

3) If we keep all the restrictions from the case 1) above, but the objective function will be a quadratic one, we get quadratic programming (for more details see section 4.1.3 below).

Problems of this type are in some cases solvable in polynomial time – when the matrix used for calculation of quadratic objective function coefficients is positive definite. When this matrix is indefinite, the problem is NP-hard.

Coming back to the original CSP or COP resp., where finite domains D_i ($i = 1 \ldots n$) of variables x_i ($i = 1 \ldots n$) are assumed 0, 0, 0. In such a case each constraint C_i, defined between some subset of variables from X, let say between k variables ($x_{i1}, \ldots, x_{ik}$) is a relation defined as a subset of the Cartesian product $D_{i1} x \ldots x D_{ik}$. There are basically two main approaches how to solve such kind of problems.

a) *Inference (constraint satisfaction)* tries to reduce the number of alternative values for particular variables so that it looks on small subsets of variables, which are constrained by any of the constraint from C. This makes it possible to eliminate from the domain of a particular variable those values which certainly cannot be present in the final solution (because they would violate some of the constraints). Algorithms performing such elimination or checking are called consistency checking algorithms. They are very fast and effective for small variable sets, composed of one (node consistency) or two variables (arc consistency). For larger subsets the computational complexity of consistency algorithms grows significantly and therefore search strategies are usually applied on CSP or COP reduced by inference strategies.

b) *Search* (or combinatorial search) is actually a method to (as efficiently as possible) look for feasible solutions, or for the optimal one resp. There are plenty of search algorithms, which range from very simple ones (like exhaustive search, backtracking, etc.) up to very sophisticated ones (dynamic backtracking, full looking ahead etc.). All search methods mentioned up to now are complete search methods, i.e. they systematically search the whole space of possible solutions. One prominent search approach for discrete optimization is branch and bound (for details see section 6.1.4). But many of the discrete optimization problems are proved to be NP-hard, so the complete search may simply run for very (very) long time even on very powerful computers. Therefore heuristic search approaches are often applied as well. Although they cannot guarantee that an optimal solution will be found, they often yield very good suboptimal solutions in reasonable time. From this group of methods we can mention a few, e.g. simulated annealing, tabu search or genetic.

Very nice and efficient is the combination of both basic approaches, i.e. constraint satisfaction with search. An excellent prototyping framework for modelling and solving constraint optimization problems, which provides suitable algorithms from both above mentioned groups, is constraint logic programming, e.g. the open source ECL^{i}PS^e 0 (Niederlinski, 2011). This framework uses logic programming as basic framework (declarative programming approach based on first order logic), enhanced with various constraint satisfaction, search and optimization strategies (both, for discrete as well as continuous domains of variables in CSP or COP) that are already available in form of built-in predicates.

Linear programming

Linear programming can be used to express a wide variety of different kinds of problems. We can use algorithms for linear programming to solve the max-flow problem, solve the min-cost max-flow problem, find minimax-optimal strategies in games, and many other things (Blum, 2011).

Tasks of linear programming are a special case of tasks of mathematical programming when objective function and constraints are linear functions. The term linear programming arose during World War Two in USA. The word „linear" results from characters of the objective function and the constraints and the word „programming" results from applications in areas of planning or action scheduling. Such tasks belong for example to this kind of tasks:

– tasks for determination of optimal production program (profit can be as the objective function, the constraints are applied on sources);

– tasks for optimal mixture proportion from particular stocks – sources (the objective function can be a price of sources, the constraints can be content of particular substances in the mixture and quantity of sources) – in more general this problem can be also used for scheduling of some medical processes;

- transportation tasks – a task is to transfer sources from suppliers to consumers where the objective function can be costs and quantities of sources and required quantities of supplies can be the constraints;

- assignment tasks – a task is to assign particular operations (performances, services, operations, etc.) to particular resources (workers, machines, equipments, etc.), the objective function can be costs (or time), the constraints can be capacity values of resources for particular operations, requirements for execution of all required operations, etc..

Special methods and algorithms for their solution have been developed for tasks of linear programming.

Linear programming is used mainly for production control and scheduling. We can say that a typical task of the linear programming is the following: a company is able to produce n different products from m different sources. Assumption of particular sources for unit of each product is known, but assumption of the sources is limited. The task is to design such a production program, that rough production value is maximal.

According to Blum (2011) the linear programming problem may be defined as the problem of *maximizing or minimizing a linear function subject to linear constraints*. The constraints may be equalities or inequalities. The objective function value that is being optimized is in the form:

$$J(\overline{x}) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

The constraints are given by following equations:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq a_{10}$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq a_{20}$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq a_{n0}$$
$$x_j \geq 0, \quad j = 1, \ldots, n$$

Where x_j are the decision variables whose optimal values are to be found, and c_i, a_{ij}, and a_{i0} are constants derived from the specifics of the problem.

Integer programming

An integer programming problem may be defined again as the problem of *maximizing or minimizing a linear function subject to linear constraints*. The constraints may be equalities or inequalities. The objective function value that is being optimized is again in the form:

$$J(\overline{x}) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

The constraints are given by following equations:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq a_{10}$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq a_{20}$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq a_{n0}$$
$$x_j \geq 0, \quad j = 1, \ldots, n$$
$$x_j \in M_j \subseteq Z$$

Where x_j are the decision variables whose optimal values are to be found, c_i, a_{ij}, and a_{i0} are constants derived from the specifics of the problem, and Z is set of integers. .The difference compared to the linear programming is that solutions x_j must be integers.

The tasks of the integer programming can be classified as follows:

– pure integer programming tasks:

$$M_j = \{1, 2, \ldots, n\}$$

all variables are integers;

– mixed integer programming tasks;

$$M_j \subset \{1, 2, \ldots, n\}$$

some variables need not to be integers;

– Zero – one integer programming tasks

$$M_j = \{0, 1\}$$

all variables can get only values 0 or 1.

A few examples of tasks that can be solved by the integer programming:

- tasks of optimization of production schedule where amount of products counts in pieces;

- cutting problem – the decision variables represents amounts of pieces of initial material that are cut by particular cutting schedules;

- location problem – the decision variables x_{ij} get values 0 or 1 according to that i-th object from one group is or is not allocated to the j-th object from another group;

- Knapsack problem – any integer programming, which has only one constraint, is referred to as a knapsack problem. N items to be packed in a knapsack. The knapsack can hold weight up to W [kg] of items. Each item has weight w_i [kg] and benefit a_i. Aim is to pack the knapsack such that the total benefit is maximized. The decision variables x_j get values 0 or 1 according to that j-th subject is or is not packed into the knapsack.

Quadratic programming

A linearly constrained optimization problem with a quadratic objective function is called quadratic programming. Because of its many applications, quadratic programming is often viewed as a discipline in and of itself. More importantly, though, it forms the basis of several general nonlinear programming algorithms (Pardalos, et al., 1991).

Quadratic programming is a method used to optimize a multivariable quadratic function that may or may not be linearly constrained (wiseGEEK, 2011). Many real world problems, such as optimizing a company's portfolio or reducing a manufacturer's costs represent tasks which objective function is a sum of linear and quadratic form, non-trivial constraints are linear non-equations, where it is required that all variables are non-negative. Mathematic formulation of the task is in the form:

$$J(\overline{x}) = \overline{c}^T \overline{x} + \overline{x}^T \overline{C} \overline{x} \to \min$$
$$\overline{A} \overline{x} \leq b, \quad \overline{b} \geq 0$$
$$\overline{x} \geq 0$$

Mathematical optimization techniques are used in quadratic programming to minimize an objective function. The objective function is comprised of a number of decision variables that may or may not be bounded.

As mentioned above it is often interesting to use multivariate quadratic functions to describe real world problems. For instance, using modern portfolio theory, a financial analyst will try to optimize a company's portfolio by choosing the proportion of assets that minimize the risk associated with a given expected return. A quadratic equation made up of asset weights and the correlation between assets describes the portfolio variance which can be minimized using quadratic programming. Another example might be a manufacturer that uses a quadratic equation to describe the relationship between different quality components and a product's cost. The manufacturer can

minimize costs while maintaining certain standards by adding linear constraints to the quadratic program.

One of the most important conditions in solving a quadratic program is the convexity of the objective equation. Non-convex quadratic functions may have local or global minimums, but they are more difficult to determine. There are many approaches to solving a convex quadratic function using quadratic programming. The most common approach is an expansion of the simplex algorithm. Some other methods include the interior point or barrier method, active set method, and the conjugate gradient method. These methods are integrated into certain programs such as Mathematica® and Matlab® and they are available in libraries for many programming languages.

Branch and bound

Branch and bound (BB) is a general algorithm for finding optimal solutions of various optimization problems, especially in discrete and combinatorial optimization (Paralič, 2010). BB is the method of solution of tasks of integer programming. It consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded by using high and low estimated bounds of the quantity being optimized.

For definiteness, we assume that the goal is to find the *minimum* value of a function $f\left(\overline{x}\right)$, where $\overline{x}$ ranges over some set of *admissible* or *candidate solutions*. Note that one can find the *maximum* value of $f\left(\overline{x}\right)$ by finding the minimum of $g\left(\overline{x}\right) = -f\left(\overline{x}\right)$.

A branch-and-bound procedure requires two tools (principles) which determine its name.

1. *Branching* – splits a set of admissible (or candidate) solutions (SAS) to line-up of generally disjoint subsets. In this way the task is decomposed to smaller and more easy solvable tasks, these tasks are tasks of identical type and therefore it is possible to use recursive function call or procedure call.

2. *Bounding (bound estimation)* – it estimates values of objective function on a set of admissible solutions or on some its subset respectively (high bound and low bound). The key idea of the BB algorithm is: if the low bound for some tree node (set of candidates) *A* is greater than the high bound for some other node *B*, then *A* may be safely discarded from the search. This step is called pruning, and is usually implemented by maintaining a global variable *m* (shared among all nodes of the tree) that records the minimum high bound seen among all subsets examined so far. Any node whose low bound is greater than *m* can be discarded.

The procedure of the branch and bound method for a case of maximization of an objective function is the following:

1. <u>Initialization</u> – we consider that whole SAS is unique candidate for pruning. We establish low bound of the objective function $f_L = -\infty$ (*this the best found solution until now, it holds globally*) and then we estimate high bound of the objective function f_H (*it holds only for given set of solutions, in this case whole SAS*) in dependence on nature of the task (the greatest theoretically possible value of the objective function for a solution from SAS).

2. <u>Branching</u> – according to chosen branching law (for example the greatest value f_H) we choose one set from candidates of branching and we split it into one or more subsets of admissible solutions. If the set of candidates is empty then the algorithm terminates and the low bound f_L represents an optimal solution, $f_L = f\left(\overline{x}\right) \Rightarrow \overline{x}$ is an optimal solution. . If $f_L = -\infty$ then solution does not exist (SAS is empty).

3. <u>Determination of high bound</u> f_H – for each new subset we determine (establish) the high bound of the objective function on this subset (*the best theoretically possible value of the objective function for solutions from given subset*).

4. <u>Pruning</u> – we eliminate subsets where $f_H < f_L$ or they are empty.

5. <u>Determination of low bound</u> f_L – we determine the best acceptable solution $\overline{x}$ for each new subset. If $f(\overline{x}) \geq f_L$ then we modify actual low bound $f_L = f(\overline{x})$, $\overline{x}$ is the best found solution until now and we make again pruning where we detect if it is possible to eliminate more subsets following the condition from step 4.

6. <u>Continue</u> by step 2.

By BB algorithm we can solve for example such types of tasks – a local airline provides aerial communications from an island A to two target towns – B and C. Company has only one aeroplane, aim is to decide how many flights to realize to each of towns so that profit would be maximal.

Genetic algorithms

Genetic algorithms are a group of search techniques used in computer science developed in the 1970s. These search techniques are based on the process of biological evolution and are used to provide useful solutions to optimization and search problems. Unlike artificial neural networks (ANNs), designed to function like neurons in the brain, these algorithms utilize the concepts of natural selection to determine the best solution for a problem. As they are based on biological evolution, they use techniques that are emulated from the concepts of inheritance, mutation and selection (Orfano, 2011).

As Orfano (2011) explains in simplified terms, genetic algorithms generate a population of solutions for a given problem. Next, the 'fittest' solutions are selected and these solutions 'reproduced' (with a chance of 'mutations' occurring) resulting in a new population of possible solutions, with a different composition than the previous population. This process ends when a solution is found that adheres to certain criteria.

Genetic algorithms are created mathematically using vectors, which are quantities that have direction and magnitude. Parameters for each trading rule are represented with a one-dimensional vector that can be thought of as a chromosome in genetic terms. Meanwhile, the values used in each parameter can be thought of as genes, which are then modified using natural selection.

According to Investopedia (Invest, 2011) there are three types of genetic operations that can then be performed:

- Crossovers represent the reproduction and biological crossover seen in biology, whereby a child takes on certain characteristics of its parents.

- Mutations represent biological mutation and are used to maintain genetic diversity from one generation of a population to the next by introducing random small changes.

- Selections are the stage at which individual genomes are chosen from a population for later breeding (recombination or crossover).

In general there are three operators used in a five-step process:

1. Initialize a random population..

2. Select the chromosomes.

3. Apply mutation or crossover operators to the selected parents and generate an offspring.

4. Recombine the offspring and the current population to form a new population with the selection operator.

5. Repeat steps two to four.

Lagrange's multipliers

Lagrange's function is a method for solution of optimization tasks with constraints of type of equation (Jensen, 2004). Mathematic formulation of the task is in the form:

$$f\left(\overline{x}\right) \Rightarrow \min$$

$$\overline{g}\left(\overline{x}\right) = 0$$

The task is to find an extreme of function $f\left(\overline{x}\right)$ with constraints $\overline{g}\left(\overline{x}\right)$. Necessary condition for finding an extreme is in the form:

$$\nabla f\left(\overline{x}\right) = -\overline{\lambda}^{T}\nabla\overline{g}\left(\overline{x}\right)$$

Where ∇ is gradient of the functions $f\left(\overline{x}\right)$ and $\overline{g}\left(\overline{x}\right)$ and $\overline{\lambda}$ is vector of unknown real numbers – Lagrange's multipliers. Lagrange multipliers are a very useful technique in multivariable calculus. One of the most common problems in calculus is that of finding maxima or minima (in general, "extrema") of a function, but it is often difficult to find a closed form for the function being extremized. According to (Jensen, 2004) such difficulties often arise when one wishes to maximize or minimize a function subject to fixed outside conditions or constraints. The method of Lagrange multipliers is a powerful tool for solving this class of problems without the need to explicitly solve the conditions and use them to eliminate extra variables. In general, Lagrange multipliers are useful when some of the variables in the simplest description of a problem are made redundant by the constraints.

By Lagrange multipliers we can solve for example such type of tasks – determine dimensions of a tank so that for given volume it would have minimal surface.

4.2   Software for generic Optimization Models

For each of the above mentioned optimization models, there are a lot of implementations for special purposes as commercial and non-commercial solutions available, cf. (GOS, 2011) (IPS, 2011) (LPS, 2011) (MLOS, 2011) (QPS, 2011). Each of them implements one or more mathematical models in a totally configurable system or process and according to specific inputs, it provides the outputs which are influenced by it. This chapter will focus on the most popular ones.

4.2.1   Commercial Solutions

FICO™ Xpress Optimization Suite

Xpress Optimization Suite (FICO, 2011) by FICO is a suite of products that includes a mathematical modelling/programming language Xpress Mosel and a full range of powerful optimization engines, called Xpress-Optimizer.

Xpress-Mosel is designed for rapid modeling of optimization problems. It is easy to learn and user extensible. It features a redesigned visual development environment and includes IVE, a visual editor (and program editor), compiler, execution environment, debugger and profiler.
Xpress-Optimizer is a set of sophisticated, multi-threaded optimization algorithms for solving linear programming (LP), mixed integer programming (MIP), mixed integer quadratic programming (MIQP), quadratic programming (QP), quadratically constrained programming (QCQP), mixed integer quadratically constrained programming (MIQCQP) and convex general non-linear programming (NLP) problems.

The Xpress-Optimizer has both a low-level matrix-oriented API as well as an object-oriented API called BCL which provides functions for incrementally building the model in an intuitive fashion, solving and analyzing the results

Xpress is available on all common computer platforms and provides a range of user/software interfaces including a visual development environment, callable library APIs in C, C++, VB, Java, .NET and standalone command-line interfaces.

FICO offers Xpress Optimization Suite licenses to companies that need the full product suite or just an optimization engine.

MATLAB® by MathWorks®

MATLAB (MathWorks, 2011) by MathWorks is a high-level language and interactive environment to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and FORTRAN. Several toolboxes are available on a wide range of topics, such as
- Parallel Computing
- Math, Statistics, and Optimization
- Control System Design and Analysis
- Signal Processing and Communications
- Image Processing and Computer Vision
- Test and Measurement
- Computational Finance
- Computational Biology
- Code Generation
- Application Deployment
- Database Connectivity and Reporting

For optimization problems two toolboxes are of interest.

Global Optimization Toolbox

Global Optimization Toolbox provides methods that search for global solutions to problems that contain multiple maxima or minima. It includes global search, multi-start, pattern search, genetic algorithm, and simulated annealing solvers. These solvers can be used to solve optimization problems where the objective or constraint function is continuous, discontinuous, stochastic, does not possess derivatives, or includes simulations or black-box functions with undefined values for some parameter settings.

Optimization Toolbox

Optimization Toolbox™ provides widely used algorithms for standard and large-scale optimization. These algorithms solve constrained and unconstrained continuous and discrete problems. The toolbox includes functions for linear programming, quadratic programming, binary integer programming, nonlinear optimization, nonlinear least squares, systems of nonlinear equations, and multi-objective optimization.

Figure 5: Linear programming used in the design of a plant for generating steam and electrical power.

In addition to the toolboxes provided by MathWorks, MATLAB software provides interfaces to external routines written in other programming languages, data that needs to be shared with external routines, clients or servers communicating via Microsoft® Component Object Model (COM), and peripheral devices that communicate directly with MATLAB software.

Several add-ons are also available for solving special problems, e.g. LOQO has a MATLAB interface for quadratic programming (L/M, 2011).

Several types of licenses are available for commercial, academic, or student use, both as individual license and group or network licenses. Even though costs arise when using products by MathWorks, pricing details were not available.

Evolver by Palisades Software

Evolver (Evolver, 2011) by Palisades Software is a genetic algorithm optimization add-in for Microsoft Excel. Evolver uses innovative genetic algorithm (GA) technology to quickly solve optimization problems in finance, distribution, scheduling, resource allocation, manufacturing, budgeting, engineering, and more.
Virtually any type of problem that can be modelled in Excel can be solved by Evolver, including complex nonlinear problems.

Evolver is available as Professional and Industrial editions. The Professional edition allows up to 250 adjustable variables per model, while the Industrial edition allows unlimited variables.

Evolver is available by itself or as part of the DecisionTools Suite, Palisade's complete risk and decision analysis toolkit. The DecisionTools Suite includes @RISK for risk analysis using Monte Carlo simulation, NeuralTools for prediction using neural networks, StatTools for statistical analysis, PrecisionTree for decision trees, TopRank for what-if analysis, and more. Evolver is fully compatible with all DecisionTools programs and can be combined with them for greater insight and analysis.
Evolver is available through a variety of licensing options, including stand-alone, concurrent network, corporate, and academic licenses.

Analytica by Lumina Decision Systems

Analytica® (LDS, 2011) is a visual software package developed for creating, analysing and communicating quantitative decision models. It can be used in all industries, such as Manufacturing and Transportation, Food and Agriculture, Health and Pharma, or Energy. Several modules are available.

Analytica editions are available for Model Builders as well as for End Users.

Analytica Optimizer for model builders is the highest level of Analytica. It offers linear programming (LP), quadratic, mixed integer, and general non-linear programming. Analytica Decision Engine (ADE) is needed to run the models and provide an API for external software.
Analytica also provides the possibility to plug and play other solver engines, e.g. MOSEK, XPRESS, OptQuest et al.

ADE license includes two licenses, one for a public server and one for a development server, plus a copy of Analytica Enterprise as the development tool. Optimizer can be also be added to ADE.
The Analytica License Agreement (LDS1, 2011) or ADE license agreement (LDS2, 2011) are both available in PDF format.

Witness by Lanner

WITNESS (LG, 2012) is a business simulation software for modelling business activities throughputs and simulating the implications of different business decisions. WITNESS includes in-depth support for Six Sigma projects.

It is available in two versions - the Manufacturing Performance Edition and the Service and Process Performance Edition. Optional modules, such as WITNESS Optimizer, add virtual reality, optimization, VISIO and CAD linkage.
A developers-edition of WITNESS offering a COM object model and ActiveX displays is also available.

The WITNESS Optimizer offers a wide choice of options for experimentation. These include:
- Random solutions: to enable an appreciation of the shape of the solution space.
- All combinations: this option should indeed be chosen if time allows as it covers all possibilities.
- Min/Mid/Max: tests the extremes and mid points of all parameter settings. Covers all options for non-range parameters.
- Hill Climb: a simple algorithm. Fast but prone to get stuck in local optima.
- Adaptive Thermo statistical Simulated Annealing: the main algorithm, a variant of simulated annealing with extra adaptive nature. Includes some elements of tabu search. Developed by Lanner in conjunction with optimization experts specifically to tackle simulation experimentation.
- Six Sigma Algorithms: an adaptation of the main algorithm to ensure that the amount of change from a defined 'As-is' situation is limited. Used for determining the best improvements that can be made without a complete redesign of a process.
- WITNESS Optimizer presents its reports as the algorithm is running. The main tables and charts can be interacted with during the run to view, sort, copy, paste and analyze. Optimizer not only searches for the optimum solution but also usually completes the analysis quickly. Using intelligent algorithms, which measure and react to the success of each analysis path, the best solution is often found by trying less than 1% of all possible combinations.

Tecnomatix Plant Simulation by Siemens

Plant Simulation (Siemens, 2012) is a discrete-event simulation tool that helps you to create digital models of logistic systems (such as production), so that you can explore a system's characteristics and optimize its performance. These digital models allow you to run experiments and what-if scenarios without disturbing existing production systems or – when used in the planning process – long before the real production systems are installed. Extensive analysis tools, such as bottleneck

analysis, statistics and charts let you evaluate different manufacturing scenarios. The results provide you with the information needed to make fast, reliable, smarter decisions in the early stages of production planning.

Arena Simulation Software by Rockwell

The Arena Simulation Suite (Rockwell, 2012) offers a convenient bundle of products for the organization with a wide range of modelling problems. It includes every modelling template and tool offered by Arena. It is available in different editions, e.g. Basic Edition, Basic Edition Plus, Standard Edition, Professional Edition, or Enterprise Suite.

Templates of the Enterprise Suite include: Basic Process, Advanced Process, Advanced Transfer, Flow Process, Blocks, Elements, Packaging, Contact Script, Contact Data. Tools include: OptQuest for Arena and Visual Designer.

Examples of applications include:
* Evaluating the ROI of new technology and capital expenditures such as equipment
* Detailed analysis of complex manufacturing processes that include material-handling intensive operations (fork trucks, AGV systems, and labour)
* Analysis of Six Sigma implementations, Kanban/Pull or Push systems
* Determining line set up, changeover procedures, and maintenance programs
* Improving local and global supply chain processes
* Detailed analysis of warehousing, logistics, or transportation, military and mining applications
* Creating a custom Arena-based template for modelling logistics network of a global travel agency

Automod Simulation Software 3D by Visual8

AutoMod Simulation (Visual8, 2012) is a product of Visual8, USA. A pioneer in 3-D discrete event simulation, AutoMod offers a variety of specialized modules that are used across diverse industries such as manufacturing, aerospace, logistics, semiconductor and transportation. Systems of any size or level of detail can easily be simulated, from manual operations to fully automated facilities, with true-to-scale 3-D virtual reality animation to analyze complex systems.

4.2.2 Non-commercial Solutions

SciLab

SciLab (Scilab, 2011) is a public domain clone of MATLAB®. It is the free software for numerical computation providing a powerful computing environment for engineering and scientific applications. Gathering both industrial needs and scientific advances, SciLab covers a wide spectrum of areas: aerospace, automotive, energy, defense, finance, chemistry, biology, or medicine.

SciLab includes hundreds of mathematical functions. It has a high level programming language allowing access to advanced data structures, 2-D and 3-D graphical functions.

In terms of optimization, SciLab supports nonlinear optimization, quadratic optimization, nonlinear least-square optimization, semi definite programming, genetic algorithms, simulated annealing, and linear matrix inequalities.

SciLab is governed by the CeCILL license (GPL compatible) abiding by the rules of distribution of free software. The software can be used, modified and/or redistributed under the terms of the CeCILL license as circulated by CEA, CNRS and INRIA at the following URL: http://www.cecill.info

Differential Evolution (DE)

One popular example for genetic algorithm is the differential evolution (DE) algorithm (DifferentionalEvolution, 2011). It is a practical approach to global numerical optimization which is easy to understand, simple to implement, reliable, and fast. It is available in different programming languages, such as Java, C, C++, Python as well as MATLAB code, or SciLab code.

It is widely used in commercial software packages, such as
- Built in optimizer in MATHEMATICA's function Nminimize (since version 4.2)
- MATLAB's GA toolbox contains a variant of DE
- Digital Filter Design
- Diffraction grating design
- Electricity market simulation
- Auto2Fit
- LMS Virtual Lab Optimization
- Optimization of optical systems
- Finite Element Design

5.   Energy in Automotive Manufacturing

The chapter introduces briefly energy Management Strategies in the Automotive Environment. In the following, specific data related to the energy figures of a European Automotive OEM are presented.

5.1   Energy management SOTA in Automotive[1]

The main energy types used in vehicle assembly plants are electricity, steam, gas and compressed air. About two-thirds of the budget for assembly plants is spent on electricity, while fuels are used to generate hot water and steam (mainly for paint booths), as well as heat in curing ovens.

Total energy costs are equivalent to approximately 1% of the production output by the vehicle assembly plants, making it a relatively small cost factor in the total production process. The energy costs for the assembly of a car have declined from about $70/car in 1990 to about $60/car in 1995. This cost reduction may be due to reduced energy costs during that period, increased capacity utilization at assembly plants or improved energy efficient processing (Galitsky and Worrell, 2008). According to data of major car producers in EU and USA (Galitsky and Worrell, 2008) the consumption of thermal and electrical energy per car is about 2 MWh/car and 750 to 1040 kWh/car correspondingly.

Energy management programs

Participation in voluntary programs like the EPA ENERGY STAR program or gaining ISO 14001 certification can help car producers track energy and implement energy efficiency measures. General Motors notes that using energy management programs in combination with the ISO program has had the largest effect on conserving energy at their plants. In a Canadian plant, an energy management system, using sub-metering, achieved over a 5% reduction in annual electric energy used over a three year period.

The Volvo Car Company at Born (the Netherlands) implemented an energy monitoring and registration system with the additional goal of influencing staff behaviour by regular feedback of monitoring results. The ventilation and heating controls were readjusted and recalibrated, but the changes in staff behaviour produced the best results. Some of these included closing windows and doors, reporting high room temperatures rather than opening a window, adjusting maintenance work in the lacquering line to shorten maintenance periods, switching off unused machinery, switching off lights and coolers when leaving an office, removing superfluous lights and preventing blockage of radiator and ventilation grids (Galitsky and Worrell, 2008).

Ford has also employed energy management systems at some of its plants. Ford's Cologne-Niehl plant in Germany implemented a computer-based energy management system. The computer system runs the electricity supply for the plant; it automatically adjusts heating and lighting and controls the peak demand of machinery.

In the U.S., Ford has begun assessment projects that focus on shutdown procedures, compressed air systems, lighting, building exhaust, and painting RTOs (regenerative thermal oxidizers). Ford has established a shutdown goal of 50% of demand during the week and 25% of demand on the weekends (Galitsky and Worrell, 2008).

This can significantly reduce energy on down times. For example, Ford's Edison Assembly Plant in New Jersey set up procedures to shut down equipment during non-production periods and issued work instructions to its employees, which provide 14% energy reduction in one year. Some European plants have achieved shutdown energy use equal to 20% of full load energy use. General Motors installed energy management systems in eleven facilities and has achieved more than $3.6 million in annual savings.

---

[1] http://kap-project.eu/fileadmin/KAP/user_upload/Deliverables/D1.1.pdf

The Rover Group in the UK implemented a relatively simple employee awareness program at their Longbridge site. The key to success was raising awareness at all employee levels. This was achieved by forming an energy group, a publicity campaign, a competition and performance reporting against targets. The savings achieved were in excess of $1,900,000 in the **first 6 months, with costs of just** $13,000.

5.2   Automotive Energy figures

Energy management is becoming more and more crucial in all manufacturing processes. Taking into account as an example the Automotive sector and considering particular one Original Equipment Manufacturer (OEM), it is possible to see that the energy represents about 10% of the total costs of all processes.
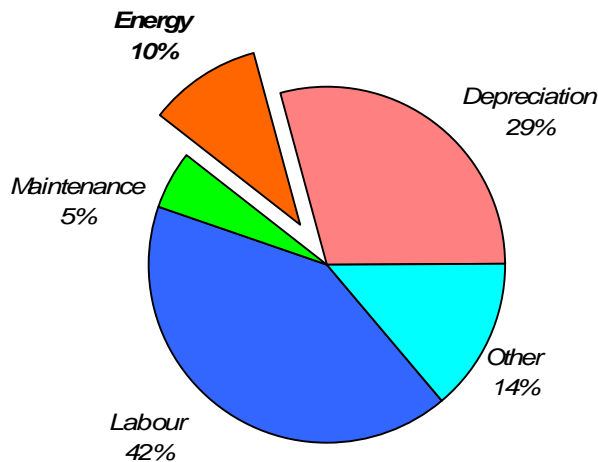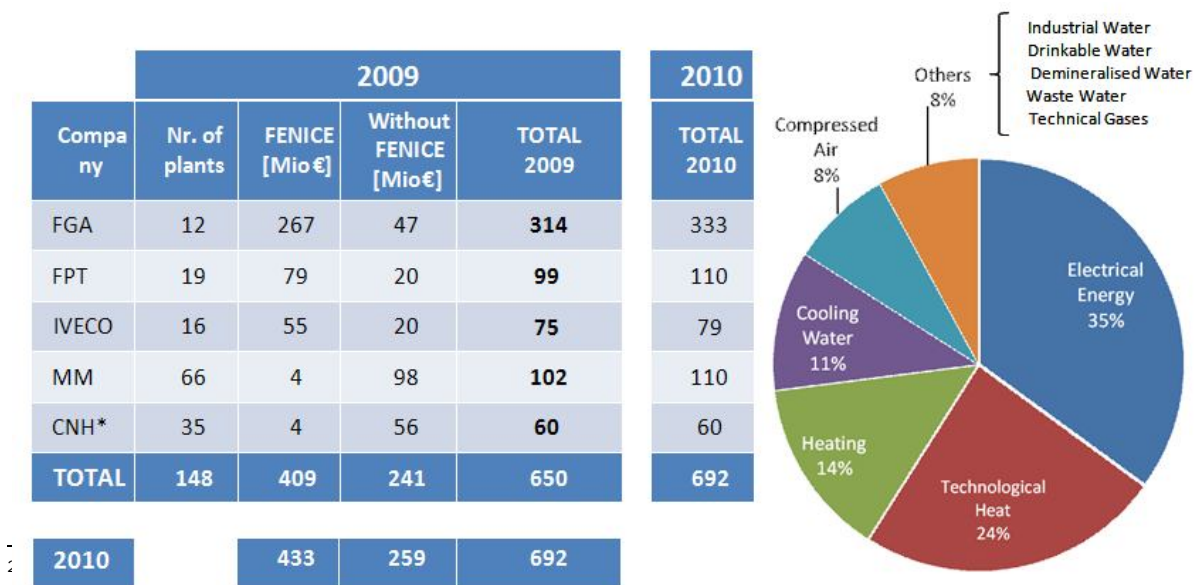

Figure 6: Automotive OEM Processing Costs

Analyzing specifically the total energy costs related to all processes of an Automotive OEM, it is possible to figure out that the consumptions could be subdivided into different areas such as generic use of electrical energy, heating, cooling and compressed air. As reported in Figure 4, electrical energy could reach up to 35% of the whole energy consumption.

2

| Company | Nr. of plants | 2009 | | | 2010 |
		FENICE [Mio€]	Without FENICE [Mio€]	TOTAL 2009	TOTAL 2010
FGA	12	267	47	**314**	333
FPT	19	79	20	**99**	110
IVECO	16	55	20	**75**	79
MM	66	4	98	**102**	110
CNH*	35	4	56	**60**	60
TOTAL	**148**	**409**	**241**	**650**	**692**
2010		**433**	**259**		**692**

* Estimated

Figure 7: FG 2009 annual expenditure in Energy (Source FG Annual REPORT)

6.   Automotive Manufacturing Use Case

A major challenge in realising the business potential of the ebbits IoPTS is found in the integration of multiple businesses operating in collaborative and ad-hoc environments with dynamic constellations of services and partnerships. One of the major scopes of project task T3.3 "Production and Energy Optimization" that includes this Deliverable is to develop a usage scenario in order to define the relationship between the IoPTS enabled Automotive Manufacturing System and methodologies for creating value based and process models in order to understand how the business interaction between stakeholders, such as the Plant Manager, the Maintenance Staff, etc., and the ebbits Platform takes place. This section will focus on the overall framework for production and energy optimization.

In this environment, an ebbits infrastructure can provide an online access to data between the following actors:

- Plant Manager able to gather an overall picture of the production trends.

- Maintenance Crew and Line Supervisor, which can interact in a more detailed level with the system.

- Ebbits Developer and System Integrator with a full access to the system and to each level of data

- Operator and Customer able to access in a very limited way to the system that provides only information on a low level.

The picture below, also analysed in D3.3, summarizes the interaction on the different levels (based on the ISA-95 Model) among the production plant and the ebbits platform.
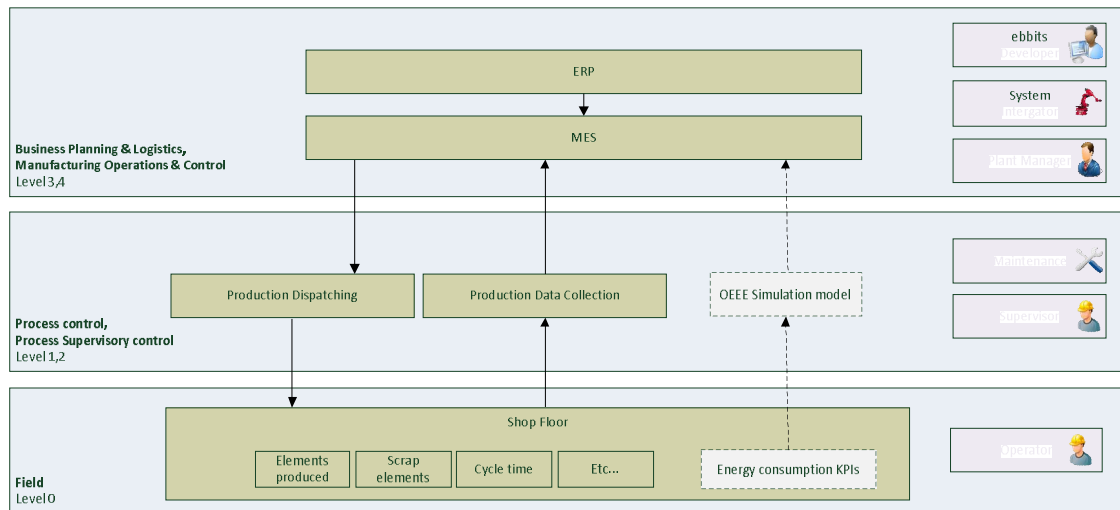

Figure 8: Stakeholders interaction in manufacturing Plant

The scope of ebbits thus will be to facilitate the integration of new devices and to make the aggregation of information gathered by the complex chain that composes the production process easier.

6.1   Body in White process

In the Design and Engineering phases it is possible to verify the main concepts of the BIW manufacturing process. In order to design welding fixtures, it is essential to understand the

processes and the sequence of operations that will be performed on the element to be worked. The first step consists in the definition of the mathematics and the process flow, in order to define the correct relationship among the operations.

The process study is a solution that helps to define and evaluate the process in order to deploy the best plan for designing the fixtures and to cut all the mechatronic devices around the product.
The expected result is a process study containing a full description of how a product is assembled and manufactured, which permits to increase your profitability by improving critical success factors in your fixture design. This process allows the designer to leverage and optimize existing resources and to better integrate the designing phases from beginning to end. It does not only help the engineer to create the fixtures faster, but it also helps to begin the design earlier in the overall cycle or even to start later, in order to accommodate unforeseen changes. The Assembly Process study allows evaluating alternatives, coordinating resources, optimizing throughput, plan for multiple variants, implement changes and estimate costs and cycle times, from the very early stages of concept planning through process.

Studying, Analyzing and Managing Manufacturing Processes enables to collaboratively plan and analyze manufacturing processes. Process study provides a collaborative environment for planning designing processes. A broad range of applications enable you to define and verify body assembly sequences, create clamping plan layout, define the required time for each operation, verify plan performance, perform plan balancing and analyze spot welding and guns.

The BIW Manufacturing Process includes:

- Body assembly operations
- Joining technologies
- Process flow

The major process of an automotive body (automotive body assembly) is electric resistance welding. Resistance welding is a group of fusion welding processes that use a combination of heat and pressure to accomplish coalescence and where electric resistance welding is not applicable, arc welding and mig-brazing, laser welding are performed. The body of automotive is made up of several hundreds of stamped components which are joined together by spot welding process.
Overall quality of the car body (BIW) and quality of the sub-assemblies, apart from quality of each stamped part, depends remarkably on quality of the welded joint.
The study of part joining, welding operations such as spot-welding and the definition of welding points and their allocation to welding operations is important.

The study of the welding process can be classified as follow:

- The number of welding spot
- The position and volume number of welding spot
- The type of welding gun
- The material and sheet metal thickness for setting of welding condition

The cycle time (so called cycle time diagram) is estimating the required working time of each station. Cycle time study is based on the information such as:

- Part loading
- Clamping
- Gun forward
- Welding
- Gun retreat
- Unclamping
- Part unloading

6.2   Energy management in Body in White plant

Automotive Body in White (BIW) line is a complex task involving the integration of hundreds of mechatronic devices such as robots, transport systems, warehouses, welding devices, etc. Simulation and Optimization are commonly applied in the design and implementation of such production systems. There are many areas within BIW plant that show complicated behaviour due to the varying nature of manufacturing processes.

Manufacturers around the world have put forth significant efforts in sustainability, i.e. the optimal usage of energy and the minimization of waste. Water, air, gas, electric and steam (WAGES or in general energy) have become one of the most elusive and hard-to-manage costs in manufacturing, with high levels of cost variability and supply volatility. By better understanding and managing WAGES resource consumption across the company, manufacturers can improve the production line effectively and so defend themselves against the other competitors in today's difficult and very uncertain business environment.

For instance, a plant manager can use this insight to pinpoint variable WAGES costs on the plant-floor, and consider ways to improve profitability. Energy Specific Custom Periodic Reports, based on a specific time frame, could support on this sense the Management to define and implement Energy Optimization Strategies.

On the other side, the Customer would be able to visualize and report the energy usage to identify cost saving opportunities. Therefore an intelligence monitoring system is required to log consumption data and thus provides the necessary information to the maintenance and to the engineering department. On the market different kind of meters are already available to monitor energy consumption. Also some energy metrics software, allow energy data collection through Object Linking and Embedding (OLE) for Process Control.

Comau as one of the most important companies providing production systems to the Automotive sectors has a significant tracking history in developing Body in White production plants.

Recently Energy has become an in important issue to be taken into account in the production of new plants and during the retooling of the old ones. Many companies in fact ask for energy monitoring systems to be applied on the already existing machines in order to collect consumption data and to optimize the production process following the "green and sustainability" trends highlighted by the market.

Actually in Body in white production plant several steps forward have been made. At Comau, in the last generation of Body in White shops, compared to the previous ones, several solutions have been deployed to reduce the whole energy consumptions and CO_2 emissions.

For example:

- The welding technology has been moved from normal frequency to middle frequency with an energy saving of 27%.

- The old robot generation (NH kinematic structure / C4G Robot Controller), has been replaced by a new one (NJ mechanical structure combined with C5G Robot control Unit) with a reduction of the energy consumption estimated at 20%.

- The optimization of the jobs performed by each robot caused a reduction for the robot installation inside the lines with a 5% energy waste reduction.

- The replacement of the pneumatic welding gun balancing with a specific compensation software produced a 35% compressed air saving.

In general, the installation of a new technology can produce an energy saving estimated at 15% if compared to an old one able to deal with the same workload.

The aim of the project in this direction is to develop an intelligent monitoring system able to log data consumption and to provide the necessary information to the maintenance and to the engineering department. With this data collection, a specific offline analysis and optimization can be performed.

6.3   Architectural Description

In order to evaluate the capabilities of the ebbits platform on the Automotive Manufacturing Use Case an architectural study has been development. This concept foresees a monitoring, diagnostics and control application. In particular two scenarios will be deeply analysed. They are respectively:

- Day by day operation, in which a deviation from the standard production with the consequent application of the corrective actions will be taken into consideration

- Introduction of a new BIW model, i.e. in an existing production plant several operations are requested in order to reconfigure the existing production lines minimizing the interference with the normal production.

The picture below summarizes the Automotive Manufacturing Use Case Framework :
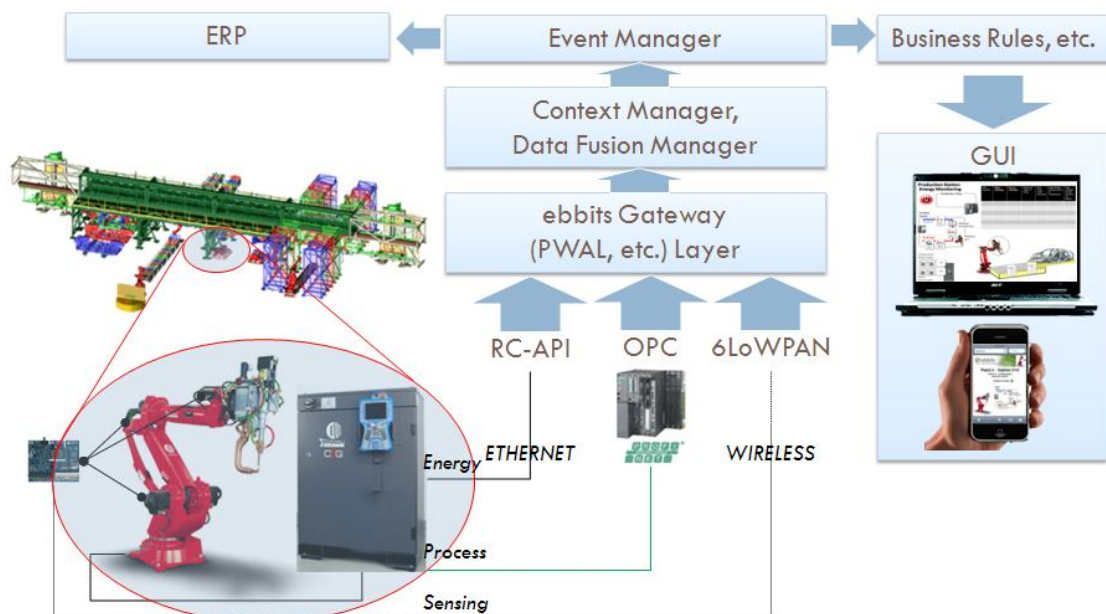

Figure 9: ebbits Automotive Manufacturing Use Case Framework

Starting from the concept that the shop floor is composed by several production lines already described in deliverable D3.3 "Business Logic Models", a single robot coordinated through its controller cabinet and a supervisor PLC is taken into account for the upcoming evaluation process.

The platform is composed by several components:

- On the low level, the robot and the PLC are representative for the industrial devices

- An ebbits Gateway, the Physical World Adaptation Layer (PWAL), has the main role of interfacing the field with the application level

- Context Manager and Data Fusion Manager are responsible as a middle layer between the monitoring applications and the PWAL

- The Event Manager's main scope is to handle all the events raised by the lower levels to the Enterprise Resource Planning and the Business Rules Engine

- All the platform data and information are visualized through specific visualization applications (GUI).

6.4   Data collection strategy for OEE/OEEE calculation

In order to compute the OEE and OEEE indexes, the required information (already explained in section 0) should be gathered from the field. For such reason, within this section we propose a first data collection strategy that could be deployed at "production station" level and then replicated to stations which compose a production line and then extended to the whole plant.

Basically, almost all of the data needed to compute these indexes are located in two subsystems of the manufacturing scenario: PLCs and ERP systems. The latter contains the information related to "planned" activities (e.g., planned production time), while in the former, PLCs are in charge of gathering all the complementary "measured" data (e.g., energy consumption, uptime, etc.).

In order to better picture the whole process, Figure 10 shows an information flow architecture, which describes how all the data coming from different physical world devices and enterprise systems, gathered through the respective interface components: the Physical World Adaptation Layer (PWAL) and Enterprise Hub (EH), is used by the OEE/OEEE calculator logic (a set of context rules for instance) and then offered to the business application layer for a different kind of optimization process, which could be implemented as a set of business rules or a more complex processing involving a specific optimization tool (for instance one of software solutions previously presented in subsection 4.2).
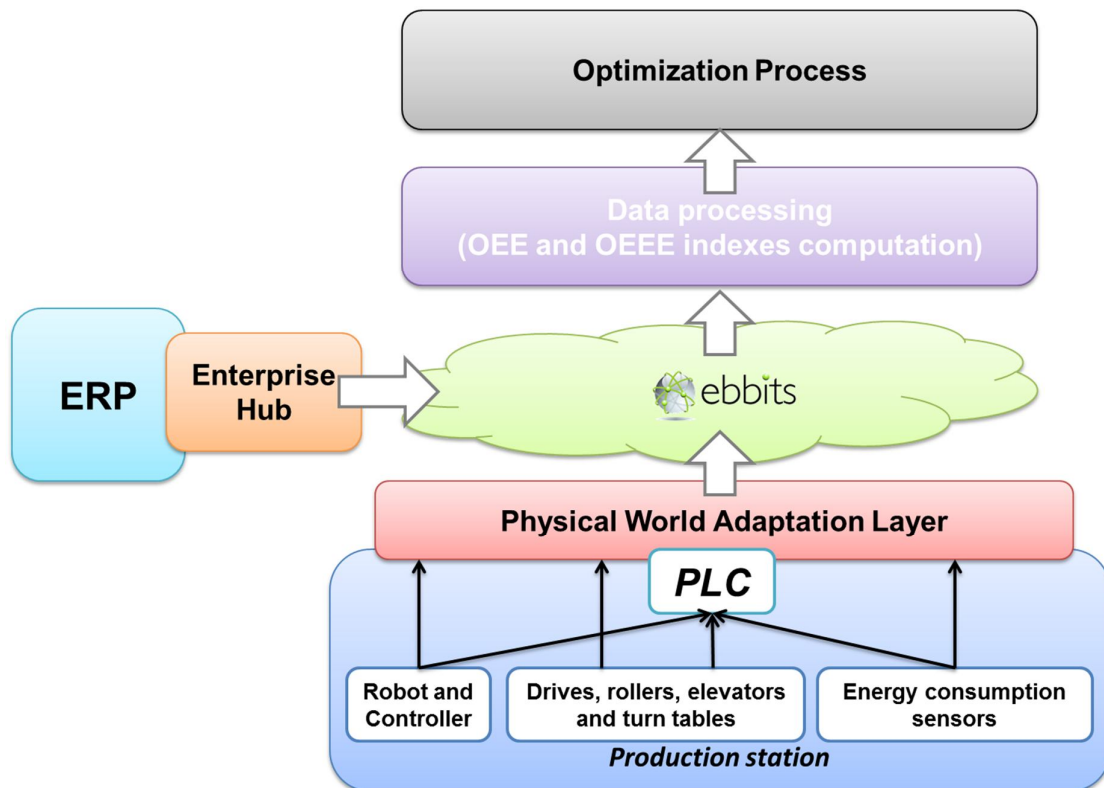

Figure 10: Data collection strategy for OEE/OEEE calculation

The computation of the OEE and OEEE indexes require different types of information to be collected, thus it is important to identify which data is needed for each one of the indexes, and according to their nature and requirements, devise the best strategy (where, when and how) to extract such

information. In the following, each OEE and OEEE index is analysed, and its data collection strategy is discussed:

6.4.1 OEE indexes

Availability [percentage]: Calculated as the ratio of Actual Production time to Planned Production Time.

Actual production time (hours): PLCs could provide indirectly this information by generating time stamped events related to the start/stop of the station operation.

Planned production time (hours): ERP systems should provide this information according to the planned activities during the analysed period of time.

Performance [percentage]: Calculated as the ratio of Actual Run Rate to Ideal Run Rate, or alternatively, as the ratio of Ideal Cycle Time to Actual Cycle Time.

Actual cycle time (minutes, hours): PLCs could provide indirectly this information by generating time stamped events related to each completed cycle.

Ideal cycle time (minutes, hours): ERP systems should provide this information according to the specified duration of a cycle.

Quality [percentage]: Calculated as the ratio of "Good" Products to Total Products.

Good Products (number of produced items that met quality standards): PLCs could provide indirectly the number of good products by generating an event each time the quality actor that decide when a product meets or not the quality standard (either by a manned inspection or an automated quality control process).

Total Products (total number of produced items): The same events generated at the quality control step, can be used to count the total number of products, i.e., the sum of "good" and "bad" produced items.

This quality index of course depends on the definition of a produced item, but it can be applied to single production processes performed at station level, where the quality control is made on the elaboration performed in that specific cell.

6.4.2 OEEE indexes

Energy consumption [Watts per hour]: Calculated as the energy consumed within a time interval. This metric can be calculated at several levels (machine, station, line, shop, etc.) and be associated to different tasks (e.g., *a robot consumes XX W h when moving, while YY W h when idle*). The consumption of each device involved can be retrieved directly from the device driver itself if it is supported, or by installing proper power meters which monitors the consumption of the device.

Energy consumed [Watts]: Calculated as the energy consumed for accomplishing a determined task. This metric can be calculated either at machine or product level (e.g., *a station consumes XX W for a complete cycle, or YY W were consumed for producing item ZZ*). The consumption of each device involved can be retrieved directly from the device driver itself if it is supported, or by installing proper power meters which monitors the consumption of the device. Events regarding the energy consumption may be generated at the end of each production cycle, which can then be associated to the produced item.

6.5 Updated business vocabularies

The initial set of horizontal cross-domain vocabulary was examined and presented in the deliverable D3.2. The aim of this section is to update the initial vocabulary, according to the identification of terms used in the business environment of the manufacturing domain. At the same time, the relevancy to the second, agricultural domain was examined and marked. Identified domain-specific and initial business-related terms, which describe the business framework in the manufacturing domain, will be an extension of the first horizontal vocabulary located on wiki.

Terms	Description and notes	Relevancy to Agricultural domain
Simulation	The imitation of the operation of a real-world process or system over time. The act of simulating something first requires that a model be developed; this model represents the key characteristics or behaviours of the selected physical or abstract system or process.	x
Discrete event simulation	Simulation represents modelling, simulating and analyzing systems utilizing computational and mathematical techniques, while creating a model construct of a conceptual framework that describes a system	x
Monte Carlo simulation	The method of repetitive trials with gambling similarity, utilizes models of uncertainty where representation of time is unnecessary.	x
Optimization	Selection of the best alternative out of the set of all possible alternatives	x
Body in white (BIW)	The stage in automotive design or automobile manufacturing in which a car body's sheet metal components have been welded together — but before moving parts (doors, hoods, and deck lids as well as fenders) the motor, chassis sub-assemblies, or trim (glass, seats, upholstery, electronics, etc.) have been added and before painting	
OEE	Overall Equipment Effectiveness index is a single figure that signifies the utilisation of a machine. This can be at a job level, shift level or overall level.	x
OEEE	Overall Equipment and Energy Efficiency index is an extension of OEE index through adding the energy component.	x
EMS	Energy management system refers to a system in an organization to achieve energy efficiency through well laid out procedures and methods, and to ensure continual improvement, which will spread awareness of energy efficiency throughout an entire organisation.	

Table 1: Extended business vocabulary in manufacturing domain

Note: This is a living, rolling list, which will extend the previous one (on the project wiki) and will be continuously updated during the project.

7.   References

(Albrecht, 2012)   Albrecht, M. 2012. Introduction to Discrete Event Simulation. *www.albrechts.com.* [Online] 9 February 2012. http://www.albrechts.com/mike/DES/Introduction%20to%20DES.pdf.

(AW, 2009)   AutomationWorld, OEE Goes Real Time. [Online] 2009. [Cited: 16 01 2012] http://www.automationworld.com/information-management/oee-goes-real-time

(Blum, 2011)   Blum, A., Linear Programming. Carnegie Mellon University. [Online] 2011. [Cited: 26 10 2011] http://www.cs.cmu.edu/~avrim/451/lectures/lect1027.pdf

(Chen et al 2007)   Chen, Y.C., Liu, C.H., Wang, C.C., Hsieh, M.F. (2007). "RFID and IPv6-enabled Ubiquitous Medication Error and Compliance Monitoring System", 9th International Conference on e-Health Networking, Application and Services, 2007, 19-22 June 2007 Page(s):105 - 108.

(DE, 2011)   DifferentionalEvolution. 2011. Differential Evolution Homepage. *ICSI: International Computer Science Institute | Berkeley, California.* [Online] 2011. [Cited: 28 07 2011.] http://www.icsi.berkeley.edu/~storn/code.html.

(EC, 2007)   European Commission (2007). A lead market initiative for Europe. Brussels. COM (2007) 860 final.

(Effra, 2010)   FACTORIES OF THE FUTURE PPP STRATEGIC MULTI-ANNUAL ROADMAP

http://kap-project.eu/fileadmin/KAP/user_upload/Deliverables/D1.1.pdf

(Evolver, 2011)   Evolver. 2011. Evolver: Genetic Algorithm Optimization Software for Excel - Palisade. *Palisade Corporation: Maker of Risk & Decision Analysis Software using Monte Carlo Simulation.* [Online] 2011. [Cited: 13 09 2011.] http://www.palisade.com/evolver/.

(FICO, 2011)   FICO. 2011. FICO™ Xpress Optimization Suite. *FICO.* [Online] 2011. [Cited: 13 09 2011.] http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx.

(GOS, 2011)   Global_Optimization_Software. 2011. Global Optimization Software. *Universität Wien - Fakultät für Mathematik.* [Online] 2011. [Cited: 09 15, 2011.] http://www.mat.univie.ac.at/~neum/glopt/software_g.html#bb_codes.

(IEA, 2011)   International Energy Agency, World Energy Outlook 2011. ISBN 978-92-64-12413-4.

(IPS, 2011)   Integer_Programming_Software. 2011. Integer_Programming_Software. *NEOS Wiki.* [Online] 2011. [Cited: 15 09 2011.] http://www.neos-guide.org/NEOS/index.php/Integer_Programming_Software.

(Invest, 2011)   Investopedia. Using Genetic Algorithms To Forecast Financial Markets. [Online] 2011. [Cited: 17 09 2011] http://www.investopedia.com/articles/financial-theory/11/using-genetic-algorithms-forecast-financial-markets.asp?partner=YahooEA

(Jensen, 2004)   Jensen, S. An Introduction to Lagrange Multipliers. [Online] 2011. [Cited: 22 09 2011] http://www.slimy.com/~steuard/teaching/tutorials/Lagrange.html

(L/M, 2011)   LOQO/MATLAB. 2011. LOQO/MATLAB Interface. *Princeton University.* [Online] 2011. [Cited: 15 09 2011.] http://www.princeton.edu/~rvdb/matlab.html.

(LDS2, 2011)   Lumina_Decision_Systems_2. 2011. http://www.lumina.com/uploads/main_images/ADE%20EULA.pdf.

(LDS1, 2011)	*Lumina Decision Systems Homepage.* [Online] 2011. [Cited: 28 07 2011.] http://www.lumina.com/uploads/main_images/ADE%20EULA.pdf. Lumina_Desicion_Systems_1. 2011. http://www.lumina.com/uploads/main_images/Analytica%20EULA.pdf. *Lumina Desicion Systems Homepage.* [Online] 2011. [Cited: 28 07 2011.] http://www.lumina.com/uploads/main_images/Analytica%20EULA.pdf.
(LDS, 2011)	Lumnia_Decision_Systems. 2011. Lumnia Decision Systems Homepage. *Lumnia Decision Systems Homepage.* [Online] 2011. [Cited: 28 07 2011.] http://www.lumina.com.
(LG, 2012)	Lanner_Group. 2012. Process Simulation Modelling Software. *Process Simulation Modelling Software.* [Online] Lanner Group, 2012. [Cited: 30 01 2012.] http://www.lanner.com/en/media/witness/witness12.cfm.
(LPS, 2011)	Linear_Programming_Software. 2011. Linear Programming Software. *NEOS Wiki.* [Online] 2011. [Cited: 15 09 2011.] http://www.neos-guide.org/NEOS/index.php/Linear_Programming_Software.
(LTD, 2011)	LTD, Neustro Consulting. http://www.neustro.com/oee.aspx. [Online] Lumina_Decision_Systems. 2011. Lumnia Decision Systems Homepage. *Lumnia Decision Systems Homepage.* [Online] 2011. [Cited: 28 07 2011.] http://www.lumina.com.
(MathW, 2011)	MathWorks. 2011. MATLAB - The Language of Technical Computing. *MathWorks - MATLAB and Simulink for Technical Computing.* [Online] 2011. [Cited: 15 09 2011.] http://www.mathworks.com/products/matlab/.
(Milagro et al 2008)	Milagro, F., Antolin, P., Kool, P., Rosengren, P., Ahlsén M. (2008). SOAP tunnel through a P2P network of physical devices, Internet of Things Workshop, Sophia Antopolis.
(MLOS, 2011)	Modeling_Languages_and_Optimization_Systems. 2011. Modeling_Languages_and_Optimization_Systems. *NEOS Wiki.* [Online] 2011. [Cited: 15 09 2011.] http://www.neos-guide.org/NEOS/index.php/Modeling_Languages_and_Optimization_Systems.
(Neustro, 2011)	http://www.neustro.com/oee.aspx
(Niederlinski, 2011)	Niederlinski, A.: A Quick and Gentle Guide to Constraint Logic Programming via ECLiPSe. 2011, 420 p. ISBN 978-83-62652-08-2.
(Orfano, 2011)	Orafano, F.: Applications of Genetic Algorithms. [Online] 2011. [Cited:15.2.2012] http://www.brighthub.com/science/genetics/articles/102918.aspx.
(Paralič, 2010)	Paralič, J.: Scheduling and Logistics. TU Košice, 2010, (in Slovak).
(Pardalos, 1991)	Pardalos, P.M., Vavasis, S.A.: Quadratic programming with one negative eigenvalue is NP-hard. Journal of Global Optimization, Volume 1, Number 1, 1991, pp. 15-22
(QPS, 2011)	Quadratic_Programming_Software. 2011. Quadratic_Programming_Software. *NEOS Wiki.* [Online] 2011. [Cited: 15 09 2011.] http://www.neos-guide.org/NEOS/index.php/Quadratic_Programming_Software.
(Rockwell, 2012)	Rockwell_Automation. 2012. Arena Simulation Software by Rockwell Automation: Home. *Arena Simulation Software by Rockwell Automation: Home.* [Online] 2012. [Cited: 02 02 2012.] http://www.arenasimulation.com/.
(Rosinová, 2007)	Rosinová, D., Dúbravská, M.: Optimalization. STU Bratislava, 2007, (in Slovak).

(Rossi, 2006)   Rossi, F., Van Beek, P., Walsh, T.: Handbook of Constraint Programming. Foundations of Artificial Intelligence, Elsevier, 2006, ISSN 1574-6525

(Scilab, 2011)   Scilab. 2011. Home - Scilab Web Site. *Home - Scilab Web Site.* [Online] 2011. http://www.scilab.org/.

(Sekaj, 2005)   Sekaj, I.: Evolutionary Computing and Its Utilization in Practice. Iris, Bratislava, 2005, (in Slovak).

(Siemens, 2012)   Siemens. 2012. Tecnomatix: Siemens PLM Software. *Siemens Global Website.* [Online] 2012. [Cited: 02 02 2012.] http://www.plm.automation.siemens.com/en_us/products/tecnomatix/index.shtml.

(Tsang, 1993)   Tsang, E.P.K.: Foundations of Constraint Satisfaction. Academic Press, 1993.

(Visual8, 2012)   Visual8. 2012. AutoMod 3D Process Simulation Software. *Visual8 Simulation Modeling & Decision Support Solutions.* [Online] 2012. [Cited: 02 02 2012.] http://www.visual8.com/content-testing/automod/.

(Vorne, 2008)   Vorne Industries, The Fast Guide to OEE. [Online] 2008. [Cited: 28 02 2012] http://www.oee.com/free-oee-tools.html

(WiseGEEK, 2011)   http://www.wisegeek.com/what-is-quadratic-programming.htm