Journal of Universal Knowledge Management, vol. 0, no. 2 (2005), 148-169
submitted: 2005-10-14, accepted: 2005-11-14, appeared: 2005-11-28 © J.UKM

Towards Self-Organizing Knowledge Intensive Processes

Cornelia Richter-von Hagen, Dietmar Ratz, Roman Povalej
(Institute for Applied Informatics and Formal Description Methods,
University of Karlsruhe, Germany
{cri,dra,rpo}@aifb.uni-karlsruhe.de)

Abstract: In this paper we investigate the capabilities of Genetic Algorithms applied to the domain of Knowledge Intensive Process Improvement (Knowiπ). Knowledge intensive processes (KnowiP) can be seen as sequences of activities based on knowledge intensive acquisition and handling. Such knowledge intensive processes can be implemented in enterprises of different kind regardless of which type, production or service company. In order to measure the performance of knowledge intensive processes, performance indexes are necessary. The processes are evaluated according to these indexes. Two particular Genetic Algorithms (GAs) are applied to improve a special class of knowledge intensive processes. In a single-objective algorithm we aim at improving the duration of the process execution. Moreover, we address the presence of multiple evaluation criteria by a Multi-Objective Genetic Algorithm (MOGA) to find acceptable Pareto solutions as trade-offs. For our case, we investigate the Multi-Sexual GA (MSGA) considering the criteria service time, costs of acquisition and usage of knowledge sources simultaneously.

Key Words: Knowledge Intensive Processes, Process-oriented Optimisation, Operations Research, Genetic Algorithms.

Category: C.3, H.1, I.2.8, J.1

1 Introduction

Business processes as sequences of business activities realise a predefined business goal and create an output with a value for a customer. Today, on the one hand, business processes are more and more based on required knowledge and on the other hand they are supposed to be flexible and adaptable to the changing surrounding conditions.

Genetic Algorithms are stochastical methods that can be used to solve a very broad class of optimisation problems. They are known to solve problems in a heuristic way under consideration of the problem's environment. The surrounding conditions of business processes, especially knowledge based business processes, can be seen as the problem's environment with respect to the application of GAs. Therefore, it is useful to apply Genetic Algorithms to improve and manage business processes as well as knowledge based business processes.

2 Knowledge Intensive Process Improvement (Knowiπ)

2.1 Background

Business processes can be found in any organisation, production or service company. As the economic market conditions change rapidly also business processes need to be more and more flexible and adaptable.

Definition BUSINESS PROCESS

A business process is a sequence of activities aiming at the creation of one or more products or services with a value for a customer. It is started and finished by one or more events. As it proceeds in an organisation there is an underlying organisational structure [Richter-von Hagen and Stucky 04]. Furthermore, the activities usually need one or more resources (like people, processors, data, software, etc.) that belong to predefined resource classes.

Example BUSINESS PROCESS "ORDER PROCESSING"

Figure 1 presents a simple business process of an order processing. There are two start events *direct order* and *order by telephone or email*, one of which starts the business process, because the customer orders some product or service. After the fulfilling of different activities, like e. g. *order registration, registration confirmation* and *distribution* the end event *acceptance by customer* occurs. On the one hand elements out of the organisational structure (people) are assigned to the activities. On the other hand further resources like telephone, documents, laptops or a PDA are assigned to either the same or other activities.

A classification of the resources is reasonable for the resource allocation to activities. A set of resources with similar qualities is called resource class. Resource classes can be divided into organisational units and roles. Organisational units like departments, groups or teams arise from the organisational structure of the company, whereas roles like experts, consultants, administrators, etc. are characterised by their competences and capabilities. Resources like processors, data, software or hardware can be classified into roles. We can define e. g. a role *network printer* that comprises all disposed printers connected to the network.

In this paper we consider special business processes, namely knowledge intensive business processes or simply knowledge intensive processes. These are already defined e. g. in [Gronau and Weber 04]:

Definition KNOWLEDGE INTENSIVE (BUSINESS) PROCESS (KNOWIP)

A process is knowledge intensive if its value can only be created through the fulfillment of the knowledge requirements of the process participants.

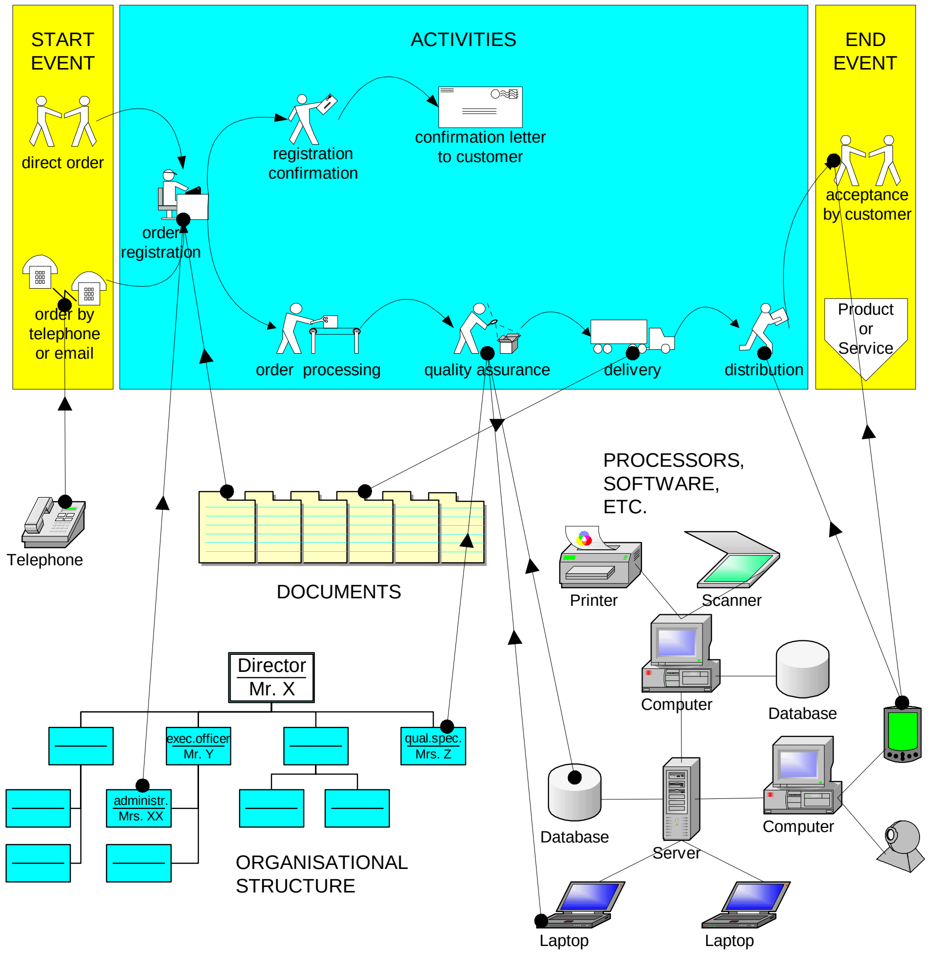


Figure 1: Business Process “Order Processing” and its Resources

The human resources are the bearers of the knowledge. This knowledge of the process participants itself can also be classified into roles and is the most important resource for many companies [Nagel 05].

Business processes can be structured, semistructured and unstructured. Structured processes are completely predefined. There are fixed and non-changeable rules for the execution of every activity. A structured process is repeatable as often as needed.

The business process in Figure 1 is structured. The next step is predefined for every activity in the process. Even though there can be more than one possibility to proceed there are fixed rules for the entire process.

Semistructured processes contain structured parts and non structured parts.

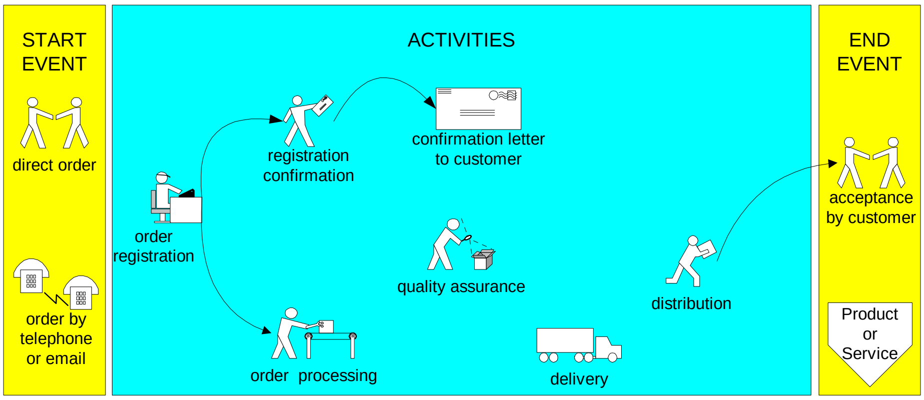


Figure 2: Semistructured Process

In Figure 2 the structured process from Figure 1 is turned into a semistructured process. This time the next step is not predefined for every activity in the process but for some of them like *order registration, registration confirmation* and *distribution*. For these activities the rules for subsequent activities are fixed and non-changeable. For all other activities there are no rules predefined. It is not fixed what kind of order can be handled and if a quality assurance is necessary or not. The subprocess from *order processing* to *distribution* is completely arbitrary and becomes fixed during process execution.

Unstructured processes are completely unpredictable. In Figure 3 the semistructured process from Figure 2 is turned into an unstructured process. This time the next step is not predefined for any activity in the process. No rules for subsequent activities are known in advance and are decided not until the preceding activity has finished. Even the potential activities need not be known in advance. E. g. after completing *order registration* the market conditions can force the decision maker to decide to buy the product or service and then deliver it to the customer which is not envisioned at all before.

Unstructured processes are not suitable for any level of automated procedures but they give plenty of freedom to the users. An unstructured and not

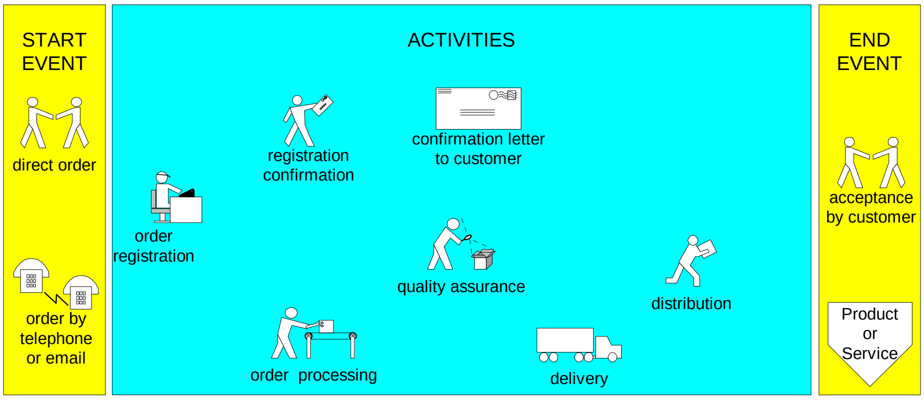


Figure 3: Unstructured Process

modelled process can be reconstructed after the execution. Then it is to decide if it continues as an unstructured process or as a semistructured process with some sound process components or even as a structured process that can be improved after every process execution. Intelligent algorithms can deliver new techniques to facilitate this improvement.

Knowledge intensive business processes can be structured but are often also semistructured processes, because "Knowledge intensive business processes are only partially mapped by the process model due to unpredictable decisions or tasks guided by creativity. Typically knowledge flows and knowledge transfers between media and persons are necessary to achieve a successful process completion." [Gronau and Weber 04].

Figure 4 shows some of the required knowledge flows and transfers between the data contained in a customers database and the users of these data. People working in different departments (like in the customer care or sales departement) or also people working for various subprocesses (like the consulting or support subprocess) need some or all of the customers data stored in the database to finish their work successfully. Data like *contact person* and *project* are required by the project teams, a consulting team needs the knowledge about the *special fields* and the *market profile* of the company, whereas data like *previous dialogues* and *determined prognoses* are indispensable for diverse forward-looking customer dialogues. Referring to the example in Figure 1 knowledge flows are also necessary there. E. g. the administrator who performs the *order registration* requires data like *name, address* and *contact person*. The quality specialist who performs the *quality assurance* can require customer data like *quality demand* of that special customer, but he also needs expert knowledge about the product composition etc. This knowledge may be in the specialist's head or in a knowledge database.

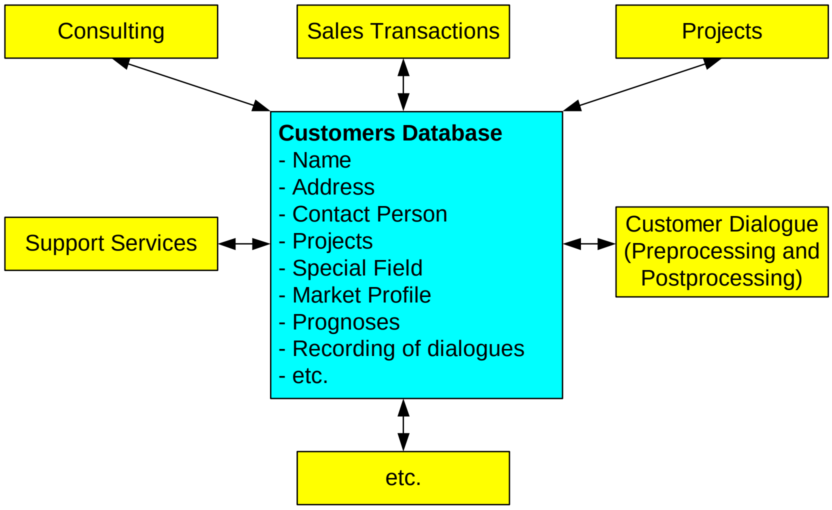


Figure 4: Knowledge Flows and Transfers

The required expenses to reach the aim reduce by reusing the knowledge resources, even if the volume of work increases. Figure 5 shows a possible creation of an object-oriented library. During the start of project 1 only the creation of the library is possible and 100 man-days are required for the project execution of a volume of work of 1 Mio. Euro. During the second project the knowledge from the library can be used already and for a volume of work of 1,5 Mio. Euro only 80 man-days are necessary. In the future the library can be applied and significant savings are possible. If required, the library can be improved continuously.

2.2 Business Process Reengineering vs. Improvement

To improve processes there are two main methods. One is known as Business Process Reengineering (BPR). This method was first mentioned around 1990 in [Hammer and Champy 93] and is the revolutionary method. The intention is to reorganise the entire company and its business processes. Possibly existing processes and organisational structures are disregarded and newly recreated. In [Hammer and Champy 93] BPR is explained as fundamental rethinking of the business processes and their radical and dramatic redesign.

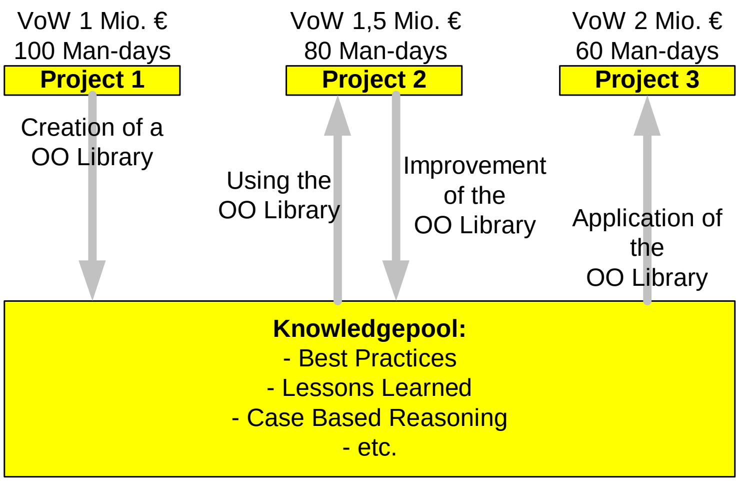


Figure 5: Improvement by a better Knowledge Use

The other method is known as Business Process Improvement (BPI) or Continuous Process Improvement (CPI) and is the evolutionary method. Whereas BPR is a fundamental rethinking and a radical reorganisation of the processes, BPI aims at a continuous and incremental improvement of business processes. BPI is an important component of Total Quality Management (TQM). TQM intends to ameliorate the customer satisfaction by improving the quality awareness in the entire company. One of the principles of TQM addresses the process awareness by emphasising that the quality of a product will improve through process control and a consequent effective assignment of resources to activities.

BPI is a continuous improvement of the process quality. A permanent adaptation and improvement of the process ensures a high process quality and performance and therefore a continuous improvement of the product quality. A high product quality guarantees a high customer satisfaction.

These methods can be transfered to knowledge intensive business processes. The method of reengineering was already transfered by some authors (see e. g. [Allweyer 98]). There is a need for a continuous process improvement because often parts of the existing processes are still effective and it is not possible to reorganise the entire company. Especially knowledge intensive processes are often derived from business processes by requiring more and more knowledge for activity execution.

Therefore, in this paper we focus on the second method, the (BPI/CPI) and call this Knowledge intensive process improvement (Knowiπ).

Definition BUSINESS PROCESS IMPROVEMENT (BPI)
BPI is an evolutionary method to improve business processes aiming at a continuous and incremental improvement of processes.

Definition KNOWLEDGE INTENSIVE PROCESS IMPROVEMENT (KNOWIπ)
Knowiπ is an evolutionary method to improve knowledge intensive business processes aiming at a continuous and incremental improvement.

2.3 Performance Indexes

The use of performance indexes enables the evaluation of the process improvement. Performance indexes are described in detail in [Gadatsch 01] and can be classified (concerning business processes) into process oriented and resource oriented and also into time oriented, value oriented and quantity oriented indexes. We extend this classification list with respect to knowledge intensive processes by a knowledge oriented and a quality oriented view as shown in Figure 6.

Performance Indexes		
process oriented	resource oriented	knowledge oriented
time oriented throughput time execution time service time idle time	time oriented operation time idle time inactive time	time oriented acquisition time adaptation time application time
value oriented costs of process	value oriented costs of used resources costs of unusable resources	value oriented costs of acquisition costs of adaptation costs of application
quantity oriented executed activities not exec. activities	quantity oriented object input object stock object output	quantity oriented used knowledge sources not used kn. sources
quality oriented output quality	quality oriented quality of used resources	quality oriented knowledge quality

Figure 6: Performance Indexes for an Evaluation

Process oriented performance indexes allow an evaluation of business processes with respect to their kind of flow. In a time oriented view there are several indexes that characterise the process. The throughput time of a process is the total time for executing the complete process from its first activity at the very beginning to its final activity at the end of the process. Therefore, the throughput time is usually larger than the execution time, because there may arise idle times when the process cannot proceed for example due to some unavailable resources. The service time index can be used to specify the time in which the process effectively serves a potential customer. In a bank, for example, the service time of a process, in which a customer withdraws money, ends when the money is paid off, whereas the process execution proceeds because of the necessary accountings. In a value oriented view, we consider only one process oriented performance index, which is the cost of process. In a quantity oriented view there are two indexes: the number of executed activities and the number of not executed activities. Since a process creates an output for a customer, we use the quality of this output as a process oriented performance index in a quality oriented view.

Resource oriented performance indexes enable an evaluation of business processes with respect to the required resources. In a time oriented view we consider the operation time, the idle time, and the inactive time to specify the workload of the involved resources. In a value oriented view, two kind of costs are interesting. These are on the one hand the costs of the resources actually used in the process and on the other hand the costs for resources which are currently unusable due to illness of employees or defects of machines. For a quantity oriented view we can count the objects, that must be treated during the process. Object input is the set of objects to be handled by a resource, whereas object output is the set of objects already handled. The set of objects currently in handling by a resource is denoted by object stock. Finally, we have the quality of a used resource (e. g. the skills of an employee) as a resource oriented performance index in a quality oriented view.

For evaluating knowledge intensive business processes, we introduced some knowledge oriented performance indexes. In a time oriented view we look at the time for acquiring knowledge, the time for adapting the acquired knowledge to the special case treated in the current process, and the time for applying the adapted knowledge. In a value oriented view, we distinguish three kind of costs according to the three knowledge based steps of acquisition, adaption, and application. In a quantity oriented view we are interested in the number of used knowledge sources and the number of unused knowledge sources. Similar to the case of resource oriented performance indexes, an important index in a quality oriented view must be the quality of the knowledge delivered by the knowledge sources.

One of the main tasks in improving a business process with respect to the described performance indexes is to determine their values. While some of the indexes' values (e. g. the time oriented indexes) are in many cases measurable and others at least computable (e. g. the value and quantity oriented indexes), it is not very easy to determine values for the quality oriented indexes. Therefore, several practical aspects of business process management should be considered for measuring and computing performance index values. Referring to this, it is worth mentioning techniques and approaches based on *six sigma*, used to achieve special levels of quality, *balanced scorecard* (BSC), a method for measuring a company's activities in terms of its vision and strategies, or *return on quality* (ROQ), where customers' satisfaction plays an important role. Moreover, direct user or customer feedback can be used for measuring the quality oriented indexes.

2.4 Surrounding Conditions and Critical Values

The surroundings of a knowledge intensive business process can either depend on the business area in which the process is implemented or they can be domain independent. Examples for either type of surroundings are preconditions for the process like dependencies between activities, dependencies between resources, dependencies between activities and resources, resource restrictions or even market conditions that effect the strategic decisions of a company, like a changing quality standard or an increasing role of innovation. Moreover, KnowiPs can depend on the used knowledge infrastructure and resources; e. g. the unavailability of a knowledge base or a knowledge resource can significantly change the corresponding process.

The change of surrounding conditions can influence one or more performance indexes and therefore lead to critical and undesirable values. Critical conditions can be all important oversteppings of fixed values for certain performance indexes. For example a throughput time for a contract (or an important letter which needs to be send exactly on time) higher than the available time could lead to a cancellation of the contract. Moreover, a sudden malfunction of a knowledge resource (e. g. an important expert becoming ill etc.) could temporarily or permanently stop a running process. Consequently it will be necessary to define all important performance indexes with their critical values in advance.

A typical input sheet for the critical values should allow to enter bounds and tolerances to specify the feasibility of performance indexes (see Figure 7). Additional efforts are necessary to fix constraints given by functional dependencies between the performance indexes. On this basis, it is possible to immediately react on critical conditions. This procedure can also play an important role in risk management, when planning, leading, and controlling the resources and activi-

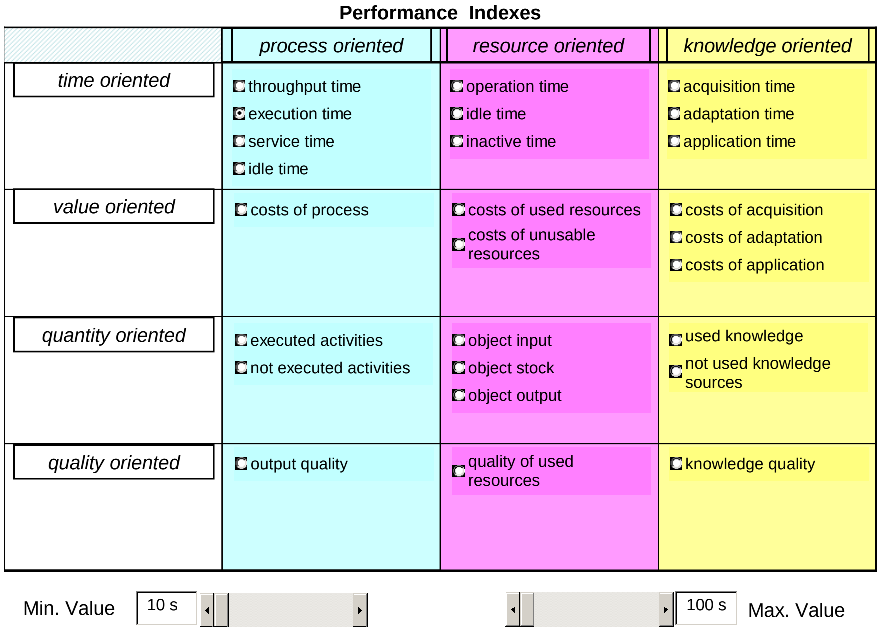


Figure 7: Input of the Critical Values for the Performance Indexes

ties of an organisation in order to decrease its risk of loss effectively. Especially in the context of knowledge intensive processes, for example in financial institutions or insurance companies, we need risk management, but we must also balance the risk of loss from unexpected causes against the economic cost of protection.

3 Trade-off, Multi-Objective Minimisation Problems, and Pareto Solutions

Knowiπ generally requires the consideration of more than one performance index. Therefore BPI is one of the many real-world problems that depend on a handling of multiple criteria. These can be described by a set of objective functions with additional constraints. Usually it will not be possible to improve all of the objective function values without worsening some others. This phenomenon is referred to as Trade-off and is demonstrated in Figure 8. The improvement of one criterion can result in a deterioration of one or more others. The aim is to best increase the imagined area between all objectives.

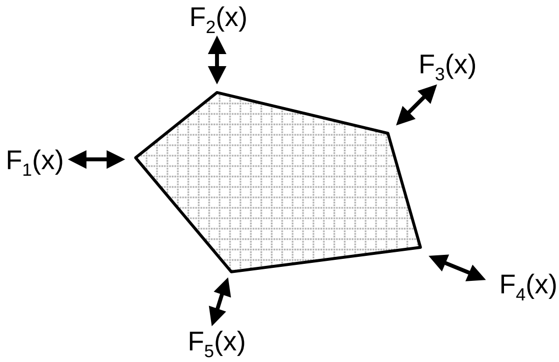


Figure 8: Trade-off of Performance Indexes

So, we have to deal with the problem of finding some kind of optimal configuration for our business process, where the objective functions (i. e. the performance indexes as described in Figure 6), which specify the optimality, are in conflict with each other. Moreover, we know, that the choice of an optimal configuration may be restricted by some conditions. Thus, a solution of our problem is only feasible, if it fulfills all given constraints. If we can describe our business process by a vector $x = (x_1, x_2, \ldots, x_n)$ consisting of n design or decision variables x_i, $i = 1, \ldots, n$, then we must find our optimal solution with respect to the condition $x \in D$, where D is the feasible domain defined by the given constraints. The main goal is to find feasible values (or simply called solutions) for x, for which the k objective functions F_j, $j = 1, \ldots, k$, have some kind of optimal values $F_j(x)$.

Therefore, we can describe our optimization problem mathematically according to the subsequent definition of a multi-objective minimisation problem, where we fixed $D \subset \mathbb{R}^n$ and $F_j : \mathbb{R}^n \to \mathbb{R}, j = 1, \ldots, k$.

Definition MULTI-OBJECTIVE MINIMISATION PROBLEM (MOMP)
Minimise $F(x)$, subject to $x \in D$,
where $x = (x_1, x_2, ...x_n)$ and $F(x) = (F_1(x), F_2(x), ..., F_k(x))$,
with k $(k \geq 2)$ and $F(x) \in \mathbb{R}^k$.

If there is no preference for any of the objectives F_j of the minimisation problem, there does not necessarily exist one unique optimal solution. Instead, it is necessary to describe the relative fitness between any two potential solutions (i. e. feasible values for x). This can be done using the Pareto dominance. A solution is said to dominate another solution when it is better on one objective function, and not worse on the other objectives. Thus a solution x dominates a solution y if and only if there exists an i with $F_i(x) < F_i(y)$ and for all $j \neq i$ it holds $F_j(x) \leq F_j(y)$. A solution is said to be nondominated if no solution can be found that dominates it. The set of solutions, which are (in this sense) better than all other solutions of the feasible region (or searchspace), is called the Pareto-optimal set or the set of nondominated solutions. It can be defined according to the subsequent definition.

Definition PARETO-OPTIMAL (NONDOMINATED) SOLUTION

A vector $x^* \in \mathbb{R}^n$ is called Pareto-optimal or nondominated solution of a MOMP, if $\nexists\, x \in \mathbb{R}^n$ with $F_i(x) \leq F_i(x^*)$ for all $i = 1, ..., k$ and $x \neq x^*$. Vector x^* is nondominated by any vector x.

Thus, the Pareto-optimal set (denoted by X^*) contains all solutions that balance the objective functions in an optimal way. The corresponding set of objective vectors denoted by $F^* = \{g \mid g = F(x) \wedge x \in X^*\}$ forms the so-called Pareto front. The solutions, whose corresponding objective vectors do not lie on the Pareto front, are referred to as dominated solutions. Of course, Pareto-optimal solutions are generally not optimal for any of the multiple criteria.

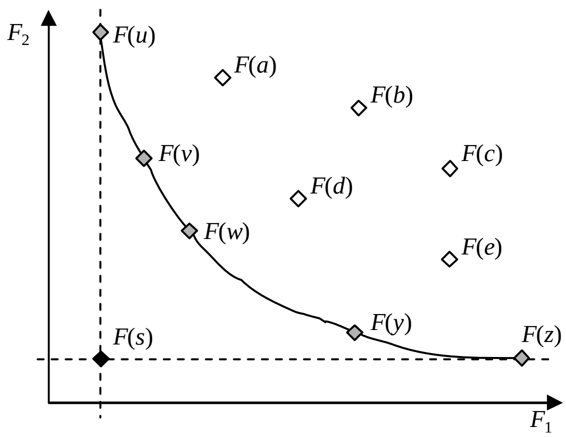


Figure 9: Example for a Pareto Front

Example PARETO FRONT FOR $k = 2$

In Figure 9 a Pareto front for the case $k = 2$ is shown. Here a, b, c, d, and e are dominated solutions, whereas u, v, w, y, and z are nondominated solutions, whose objective vectors $F(u)$, $F(v)$, $F(w)$, $F(y)$, and $F(z)$ lie on the Pareto front. The solution u produces the smallest value for the objective function F_1 and solution z produces the smallest value for the objective function F_2. The "solution" s would theoretically minimise both objective functions, but it is an unfeasible value for x (i. e. $s \notin D$), so it is an unattainable solution for the given MOMP.

4 Genetic Algorithm Methodology

4.1 Background

Given hard optimisation problems we can use probabilistic algorithms. These algorithms do not guarantee to find the optimum, but it is easy to find a much better solution than the existing one. Often also the optimum is reached. "In general, any abstract task to be accomplished can be thought of as solving a problem, which, in turn, can perceived as a search through a space of potential solutions. [...] For small spaces, classical exhaustive methods usually suffice; for larger spaces special artificial intelligence techniques must be employed. Genetic Algorithms (GAs) are among such techniques; they are stochastic algorithms whose search methods model some natural phenomena: genetic inheritance and Darwinian strife for survival" [Michalewicz 92]. Genetic Algorithms solve problems in a heuristic way under consideration of the problem's environment. They copy the idea of natural evolution, that individuals in nature try to adapt successfully to a changing environment. The well adapted individuals will survive better than the others. Genetic Algorithms use the same vocabulary as is used in natural evolution.

Definition GA CONCEPTS

GAs work on a population of n individuals. Each individual can be encoded in a certain representation, called chromosome or string. The chromosome consists of m genes. Genes of certain characters are located at certain places of the chromosome, which are called loci or string positions. The mating members (individuals) also are called parents. The parents produce children.

The individuals in the population represent potential solutions of the problem. These potential solutions can be evaluated with respect to the environment, i. e. an evaluation function is needed that represents the environment's evaluation of a proposed solution. GAs require three main components to be defined: a representation, an evaluation and fitness function, and some operators to be applied to the chosen individuals, basically the selection, crossover, and mutation operator. Figure 10 shows the progress of a GA.

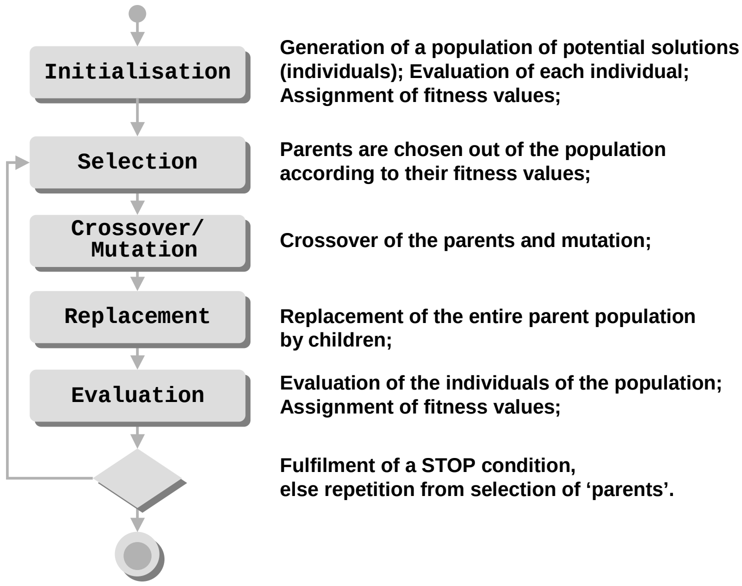


Figure 10: Universal Genetic Algorithm

During *selection* the individuals are chosen according to their fitness values, i. e. the higher the fitness value the higher the probability to get parent. Basically two parents are selected to produce two children through crossover. Crossover occurs as exchange of parts of the chromosome. One or more string positions are determined randomly and the parental parts before, behind and in between are exchanged. A simple form of mutation is the change of one gene in an individual. Whereas crossover has a high probability to occur, the mutation probability is very low.

4.2 GAs for Multi-Objective Optimisation (MoO)

Real-world problems usually necessitate the simultaneous consideration of multiple objectives (see Section 3). These objectives are in conflict with one another in the majority of cases. Trade-offs exist between some objectives, where improvement in one objective will cause deterioration in another. It is very rare for problems to have a single solution. A set of non-dominated solutions will exist.

Several GAs for multi-objective optimisation have already been presented in the literature. Details are discussed e. g. in [Fonseca and Fleming 95], [Fleming and Purshouse 01] and [Zitzler, Deb and Thiele 00]. We distinguish between aggregating and non aggregating approaches. Aggregating approaches aim at the reduction of multiple objectives into a single objective and then to apply a single-objective GA. Multiple objectives are usually combined linearly into a scalar objective by using an aggregating function. However, the determination of the weights often turns out to be very difficult.

Non aggregating approaches have been proposed. One of the first was the *Vector Evaluated GA* (VEGA) by Schaffer in 1985. The Strength Pareto Evolutionary Algorithm (SPEA) presented by Zitzler and Thiele is delivered with a comparison to VEGA and to other six algorithms (RAND, FFGA, NPGA, HLGA, NSGA, SOEA) [Zitzler, Deb and Thiele 00]; this comparison gives a ranking considering some universal problems.

The Multi-Sexual GA (MSGA) presented by Lis and Eiben [Lis and Eiben 96] is an extension of VEGA and was already successfully applied in [Bonissone and Subbu 03] to a flexible manufacturing problem.

5 GA Approach to Knowiπ

5.1 A Special Class of Knowledge Intensive Processes

We want to consider the special class of structured or semistructured knowledge intensive business processes with constrained resources (KnowiP$_{\mathrm{CR}}$). In a structured KnowiP the execution rules are fixed, i. e. all possible activities and all knowledge sources are known and predefined. A semistructured KnowiP consists of structured parts, whereas some parts can be changed during execution. Furthermore, constrained means that there exists at least one resource class containing less resources than required by the activities of the process at the same time. As an example we propose the process of a request for change during a product development project. The availability of a limited number of experts represents the constrained resources. To model this process, it is possible to apply the modelling language of Petri nets (see Figure 11).

Petri nets are known as an exact and formal language to describe business processes and therefore to facilitate an exact mapping of the process. Petri nets are bipartite graphs consisting of places $p_i \in P$ (round) and transitions $t_i \in T$ (square) connected by arrows. Places correspond to object storages like e. g. document storages or to occurred events. Transitions correspond to activities of any kind. They can contain tokens (marked Petri nets) to present the execution of the process. The main execution rules are the AND-split, AND-join, OR-split, and OR-join. Transitions followed by two places imply an AND-split; transitions preceded by two places imply an AND-join. For Places and OR-constructions respectively.

Figure 11 demonstrates the KnowiP *Request for Change*. Different activities require some knowledge resources. If a RfC is available, the customer and the

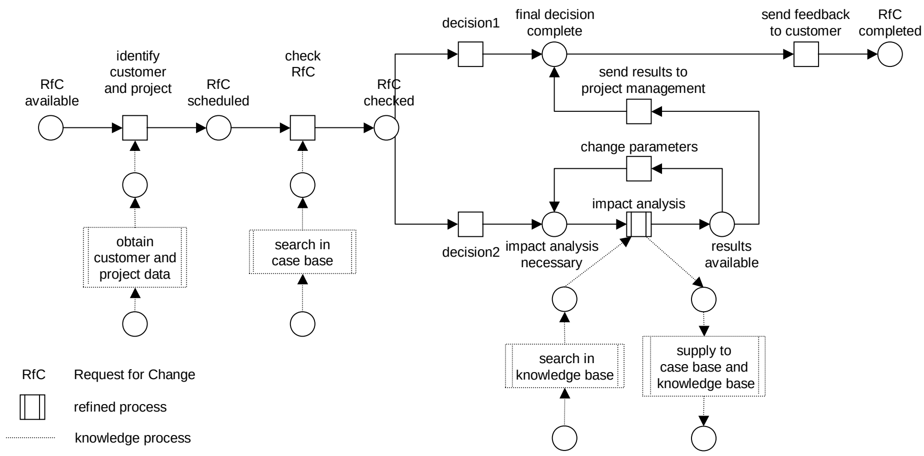


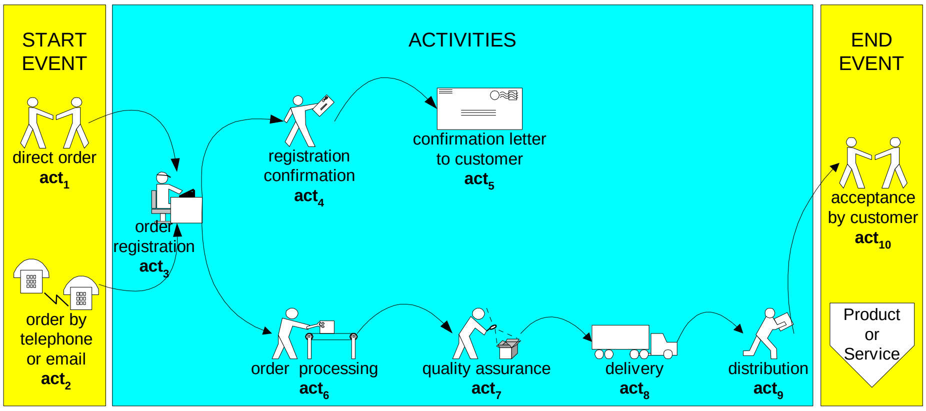
Figure 11: Example for a Knowledge Intensive Process: Request for Change during a Product Development Project

corresponding project must be identified. The customer and project data are acquired during a knowledge process and synchronised with the RfC. The next activity checks the existence of an equal or similar case in the past by searching in a case base. If on the one hand the search in the case base was successful, the only activity left is to send feedback to the RfC to the customer. If on the other hand the search in the case base was not successful, an impact analysis performed by an expert or an expert team is necessary. Another knowledge process supports the impact analysis. The experts gain knowledge from a knowledge base to treat the RfC, but also they can supply to the aforementioned case base for future RfCs and also for the knowledge base, because of acquired knowledge and experiences from the case handling. If the impact analysis is successful, the results are sent to the project management group and the customer obtains the feedback, else the impact analysis will be repeated.

5.2 GA for Single-Objective Knowiπ Problem

A particular GA is applied to the special class of KnowiP$_{CR}$ considering the single criterion *duration of execution*. The design choices of the three main components (see Section 4.1) are presented below.

The representation of the solutions is a ternary string using the ternary alphabet $\{-1, 0, 1\}$. This encoding results from the representation of a graph modelling a business process by an incidence matrix and describes the reachability of all pairs of activities (see matrix I in Figure 12). This representation is redundant, but promises some saving of time because of the maintenance of more information.



$$I = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix} \qquad I_x = \left(I_1, \quad I_2, \quad \ldots \quad I_9, \quad I_{10}, \quad 3 \right)$$

Figure 12: Business Process, Incidence Matrix, and String Representation

The single criterion *duration of execution* delivers the evaluation and thus the fitness value of the solutions. The better — the shorter — the duration of execution, the fitter the individual and the higher the fitness value.

The third of the main components are the genetic operators. Two different possibilities for selection were applied. First, the usual roulette-wheel-operator, and second, a best-half-operator. The roulette-wheel-operator chooses the individuals for a later mating randomly according to their fitness value. The higher the fitness value of an individual, the more probable this individual will be selected for mating. The best-half-operator sorts the individuals of the entire population according to their fitness values. Then all members of the best half are selected with one randomly chosen member. The crossover occurs as general one-point-crossover with a relatively high crossover probability. One string position is selected randomly and the first part to this string position of one parent is combined with the second part after the string position of the other parent. The second child is combined respectively. The mutation occurs as simple random bit flip. Usually, the mutation probability is very low. The transitive closure is included as repair operator to remain in the search space.

Generally, there are two possibilities to handle unfeasible solutions. We can either apply repair methods to move these solutions back into the feasible region of the MOMP, or we can modify the evaluation of the objective function in such a manner, that unfeasible solutions get associated with bad fitness values, to prevent further propagation of these solutions. This method is similar to the techniques applied in penalty methods.

At a first investigation only the process is improved; the resources are not considered yet. There already exists a research work that also takes resources into account (see [Hofacker and Vetschera 01]).

5.3 GA for Multi-Objective Knowiπ Problem

For the multi-objective KnowiP$_{\mathrm{CR}}$ problem the Multi-Sexual GA is applied considering the multiple criteria *service time*, *costs of acquisition* and *usage of knowledge sources*. The applications of the MSGA delivered good experiences (see e. g. [Lis and Eiben 96] and [Bonissone and Subbu 03]). An improved GA performance was already observed for some problems simply using a multi-parent crossover operator, suggested e. g. in [Eiben et al. 94]. One important advantage of multi-parent crossover operators is that they are more explorative and less sensitive to premature convergence.

The main characteristics of the MSGA are:

- as many sexes as optimisation criteria are used and each individual is evaluated according to the optimisation criterion related to its sex;
- the multi-parent crossover (requiring one parent from each sex) is applied to generate offspring;
- a set of nondominated solutions is updated during execution; this set is the output.

The representation of the solutions again is a ternary string out of the ternary alphabet $\{-1, 0, 1\}$, which consist of the rows of the incidence matrix, but this time with a sex marker (an integer) at the last string position (see string I_x in Figure 12). The criteria *service time*, *costs of acquisition* and *usage of knowledge sources* deliver the three partial evaluation functions. The fitness value of each solution is calculated according to Lis and Eiben [Lis and Eiben 96].

The operators for *Selection* are similar to the operators as used in the single-objective Knowiπ problem and used according to [Lis and Eiben 96]. *Crossover* again occurs with a high probability e. g. $p_{cross} = 0.9$ in the following manner:

Two of three chosen parents $P_1(= I_{x1})$ and $P_2(= I_{x2})$ produce child $C(= I'_x)$
for $i = 1$ to n do
if $P_1(i) = P_2(i)$ then $C(i) := P_1(i)$
else $C(i) := P_1(i)$ with probability p_{cr1}
$C(i) := P_2(i)$ with probability $p_{cr2} = 1 - p_{cr1}$.

Mutation occurs with a low probability e. g. $p_{mut} = 0.02$ in the manner that the value at the chosen string position switches from $C(i)$ to $C'(i) = -C(i)$.

Another operator is used after completing these three operators to check the validity of the obtained solution. Then, the same techniques (repair or penalty methods) as in the case of single-objective optimization described above.

Changed surrounding conditions of the KnowiP can result in critical conditions. The prevention of these demands another operator that checks the fixed critical values of the performance indexes (see Section 2.4).

If a solution is evaluated and it violates any of the constraints for the critical values of the performance indexes listed in Figure 6, the postprocessing is similar to the one practised for unfeasible solutions. So, it is also possible, even though in most cases difficult, to apply repair methods to modify these solutions in such a way, that they do not violate any performance index constraints anymore. The easier way would be to assign bad fitness values, as described above for unfeasible solutions.

6 Conclusion and Future Work

The customised MSGA was successfully applied to one of the manyfold Knowiπ problems, the KnowiP$_{\mathrm{CR}}$ (see Section 5.1). It remains to discover the potential of the described MSGA as well as other MOGAs featuring specially customised operators to other Knowiπ problems. Open questions are the fitting of MSGAs to semi- or even unstructured KnowiPs. Because of the occurrence of many constraints (maybe for the critical conditions) it should be worthwhile to investigate the implementation of a rule base. For diffuse constraints even a fuzzy rule base maybe a promising approach.

Furthermore, it is interesting to extend our version of the MSGA to the control and the automated improvement of KnowiPs. This can be achieved by first using data analysis and data mining techniques on collected data. Afterwards GA techniques can improve the KnowiPs which again delivers data for another analysis and so on. This process can be called self-organizing and promises an automated adaptation without user interaction.

We developed a prototype software environment that integrates all components (genetic optimisation kernel, transformation, backward transformation, evaluation, graphical visualisation, constraints specification, etc.) required to graphically model business processes and improve them by genetic algorithms with respect to multiple optimisation criteria. The system can be used as experimental environment for theoretical and practical studies of techniques for automated business process improvement and to further develop and test different algorithmic approaches. The system must be extended and optimized for a practical suitability. Therefore it is our plan to perform tests on real-world process improvement problems in collaboration with companies interested applying our techniques.

References

[Allweyer 98] Allweyer, T.: "Knowledge Process Redesign - Modellierung und Gestaltung der Wissensverarbeitung im Unternehmen"; Technical report, IDS Prof. Scheer GmbH, Saarbrücken (1998).

[Bonissone and Subbu 03] Bonissone, S. and Subbu, R.: "Exploring the Pareto Frontier using Multi-Sexual Evolutionary Algorithms: An Application to a Flexible Manufacturing Problem"; Technical Report 2003GRC083, GE Global Research, Niskayuna (2003).

[Eiben et al. 94] Eiben, A. E., Raué, P.-E., and Ruttkay, Z.: "Genetic Algorithms with Multi-Parent Recombination"; Proc. of the 3rd Conference on Parallel Problem Solving from Nature, Springer, (1994), 78–87.

[Fleming and Purshouse 01] Fonseca, C. and Fleming P.: "Genetic Algorithms in Control Systems Engineering"; Research Report No. 789, Department of Automatic Control and Systems Engineering, University of Sheffield, (2001).

[Fonseca and Fleming 95] Fonseca, C. and Fleming P.: "An Overview of Evolutionary Algorithms in Multiobjective Optimisation"; Evolutionary Computation 3 (1), Springer, (1995), 1–16.

[Gadatsch 01] Gadatsch, A.: "Management von Geschäftsprozessen"; Vieweg, Braunschweig/Wiesbaden (2001).

[Gronau and Weber 04] Gronau, N. and Weber, E.: "Management of Knowledge Intensive Business Processes"; Business Process Management, Springer, (2004).

[Hammer and Champy 93] Hammer, M. and Champy J.: "Reengineering the Corporation: A Manifesto for Business Revolution"; Harper Business, (1993).

[Hofacker and Vetschera 01] Hofacker, I. and Vetschera, R.: "Algorithmical Approaches to Business Process Design"; Computers and Operations Research, Elsevier, (2001).

[Lis and Eiben 96] Lis, J. and Eiben, A. E.: "A Multi-Sexual Genetic Algorithm for Multiobjective Optimization"; Proc. of the 1996 International Conference on Evolutionary Computation, IEEE, (1996), 59–64.

[Michalewicz 92] Michalewicz Z.: "Genetic Algorithms + Data Structures = Evolutionary Programs"; Springer, (1992).

[Nagel 05] Nagel, C.: "KAM.sys - Wissen strategisch nutzen"; Wissen schafft Werte: Praktische Lösungsansätze zur Organisation von Wissen, C. Nagel Verlag, Hof (2005).

[Richter-von Hagen and Stucky 04] Richter-von Hagen, C. and Stucky, W.: "Business-Process und Workflow-Management. Prozessverbesserung durch Prozess-Management"; Teubner, Stuttgart, Leipzig, Wiesbaden (2004).

[Zitzler, Deb and Thiele 00] Zitzler, E., Deb, K. and Thiele, L.: "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results"; Evolutionary Computation 8(2), Massachusetts Institute of Technology, (2000), 173–195.