Volume 28 Number 4 December 2004 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Information Society in 2004

Guest Editors:

Andrej Brodnik

Matjaž Gams

Ian Munro

The Slovene Society Informatika, Ljubljana, Slovenia

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Introduction

In 2004, the Informatica journal again invited authors of the best papers, presented at the Information society multiconference (is.ijs.si), to rewrite their papers for the journal publication. Curiously enough, the conference itself does not face any significant decline in terms of papers or the number of single conferences. It seems that the field is quite stable. Furthermore, the new sub-conferences were added following the original goal of the conference to discuss complete spectrum of topics from fundamental research upon which the Information society builds up to final application in society.

But where, then, is the enormous growth of the information society? Is it that the breakthrough has already happened and that we now face "just" a steady progress? Whatever the case, more and more fundamental results are appearing which represent a solid basis for computer and information systems that are emerging and more and more intelligent functions are performed by computer systems in order to help humans.

In a way, there has been no fault in technical systems in spite of .com failures – it is just a result of market activities going up and down in waves. To the special editors, the technical progress, be it in terms of the Moore's law or progress of software or algorithms, is quite steady. The possibilities of information society are growing faster than they get implemented; therefore the new brave world is in front of us even more than it used to be. The question we as technical editors cannot answer is – why the introduction of new possibilities into human every-day life is not as enormous as it was expected to be? The tentative answer is that we humans are just not capable to encapsulate and apply the enormous possibilities due to whatever reason, probably due to subjective reasons related to human social interrelations.

For example, one of the important aspects of society informatization is the ability of citizens to access or better to communicate with the public and governmental institutions using modern information means. In not too distant past the only information support citizen could get was possibility to apply for or at best print out certain forms issued by these institutions. Since then the situation substantially changed in some places while in the others not that much. In particular the later is the case in Central and moreover in Eastern Europe where citizens still have to physically go to specific offices and stand there in rows. It is not the case that it is harder to implement such systems in this part of the world than in, for example USA, Canada or Western Europe, in technical terms, since research and commercial institutions are quite advanced. Furthermore, no major new breakthrough is needed. It is much more in terms of the mental state of leadership and the system, therefore the social interrelations represent the problem. Researchers often enthusiastically point out advanced solutions in the most advanced countries, yet their ideas are met with bureaucratic resistance and misunderstanding.

In response to these challenges, the Information Society Multiconference was created seven years ago, with a goal to promote and exchange information and knowledge about technical and social solutions, needed to further develop information society. It is a place where researchers can present, analyze, and verify new discoveries in the scientific community first, and later prepare a ground for their enrichment and development in practice. The Multiconference utilizes its Central-European location to became a bridge, through which the Western European experiences are transferred to Central and Eastern Europe, while enriching Western thought by the unique experiences brought about by the transformations happening in Eastern and Central Europe. It is supported and co-organized by several major research institutions and societies. In 2003, we proudly announced cooperation with ACM Slovenia, i.e. the Slovenian chapter of the ACM, the largest worldwide society in computer science and informatics.

In 2004 the programming committees of single conferences proposed best papers according to the established procedure. The proposals were reviewed first by the conference team. In a discussion, majority of papers were accepted for rewriting for the journal while around 20% were not. The rewritten papers were again reviewed and then accepted for publication. Most of them are published in this issue of Informatica, while contributions from the Theoretical Computer Science sub-conference will appear in one of the following issues of Informatica due to a slightly different refereeing process that is usual for this area.

The first group of papers is related to evolutionary computing. These algorithms offer advantage of high flexibility and adaptability. These papers describe several important industrial applications where evolutionary algorithms produced sometimes quite novel solutions to experts themselves. The papers include Bäck, Willmes, and Krause*:* "*Industrial Optimization by Evolution Strategies: A Bioinspired Optimization Algorithm*"; Schönemann, Emmerich and Preuss: "*On the Extinction of Evolutionary Algorithm Subpopulations on Multimodal Landscapes";* Filipič*:* "*Optimizing Production Schedules and Energy Consumption with an Evolutionary Algorithm"; and* Koroušič-Seljak: "*Evolutionary Balancing of Healthy Meals*".

The second group of papers is concerned with advances in information society, often in relation to a specific problem, e.g. e-governing, visualization of news and data, semantic difference between words and finally information modeling. The papers in this group are: Berce: *"eGovernance: Relation Theory of the Impact*

Factors"; Grobelnik and Mladenič: *"Visualization of News Articles";* Mahkovec: *"An Agent for Categorizing and Geolocating News Articles"*; Ferlež and Gams: *"Shortest-Path Semantic Distance Measure in WordNet v2.0";* Bohanec, Džeroski, Žnidarsic, Messéan, Scatasta and Wesseler: *"Multi-attribute Modelling of Economic and Ecological Impacts of Cropping Systems"*.

The following two papers deal with software development and information systems. Jurič, Tekavc and Heričko: *"Information Systems Integration Process Model";* Kosar, Mernik, Žumer, Henriques and Pereira: *"Software Development with Grammatical Approach"*.

The next papers deal with multi-lingual approaches, specialized for Central Europe, where special languages are used, quite different than, say, English. Steingerger: *"Providing Cross-Lingual Information Access with Knowledge-Poor Methods";* Hajdinjak and Mihelič: "*Conducting the Wizard-of-Oz Experiment"*; Krstev, Vitas and Erjavec: *"Morpho-Syntactic Descriptions in MULTEXT-East — the Case of Serbian"*.

Finally, a short paper describes social changes in Slovenian schools due to information society. Pivec, Rajkovic, Jus: *"Computer Education and Social Changes in Slovenia"*.

We would also like to use this opportunity to thank the Slovenian government for cooperation and support, in particular through the Ministry of Education, Science and Sport, and the Ministry of Information Society.

Andrej Brodnik, Matjaž Gams and Ian Munro

Industrial Optimization by Evolution Strategies: A Bioinspired Optimization Algorithm

Thomas Bäck
NuTech Solutions GmbH
Martin-Schmeißer-Weg 15, 44227 Dortmund, Germany
and
Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
baeck@nutechsolutions.de

Lars Willmes, Peter Krause NuTech Solutions GmbH
Martin-Schmeißer-Weg 15, 44227 Dortmund, Germany
{willmes,krause}@nutechsolutions.de

Keywords: Evolutionary algorithm, multi-criterion optimization, airfoil design

Received: July 3, 2004

The basic variants of evolution strategies, a special instance of evolutionary algorithms, are discussed in this paper. Gleaned from the model of organic evolution, evolution strategies are characterized by the additional self-adaptive process that fine-tunes their strategy parameters during optimization. This property is a fundamental ingredient for the application to challenging engineering applications involving resource-intensive simulation runs. For one instance of such applications, the single-criterion and multi-criterion airfoil design problem, the results of an evolution strategy are presented and discussed in this paper.

Povzetek: Članek predstavlja evolucijske strategije kot poseben primer evolucijskih algoritmov z zmožnostjo samoprilagajanja in prikazuje njihovo uporabnost v optimiranju oblike profila letalskega krila.

1 Introduction

Evolution strategies [Bäc96, Rec94, Sch95] are one of the main paradigms in the field of *evolutionary computation*, focusing on algorithms for adaptation and optimization which are gleaned from the model of organic evolution. Evolution strategies are nowadays a widely accepted method for optimization in the field of engineering.

In the following sections we will give an overview of the working principles of evolution strategies and then demonstrate its capabilities with an example of the field from airfoil design. Finally, some conclusions are discussed.

2 The Algorithm

2.1 Working Principle

In general, evolutionary algorithms mimic the process of natural evolution, the driving process for the emergence of complex and well adapted organic structures, by applying variation and selection operators to a set of candidate solutions for a given optimization problem. The following structure of a general evolutionary algorithm reflects all essential components of an evolution strategy as well (see e.g. [BHS97]):

Algorithm 1:

$t := 0$;
initialize $P(t)$;
evaluate $P(t)$;
while not *terminate* **do**
  $P'(t) :=$ *variation*$(P(t))$*;*
  evaluate$(P'(t))$*;*
  $P(t+1) :=$ *select*$(P'(t) \cup Q)$;
  $t := t + 1$;
od

In case of a (μ,λ)-evolution strategy, the following statements regarding the components of algorithm 1 can be made:

- $P(t)$ denotes a population (multiset) of μ individuals (candidate solutions to the given problem) at generation (iteration) t of the algorithm.
- The initialization at $t = 0$ can be done randomly, or with known starting points obtained by any method.
- The evaluation of a population involves calculation of its members quality according to the given objective function (quality criterion).

– The variation operators include the exchange of partial information between solutions (recombination) and its subsequent modification by adding normally distributed variations (mutation) of adaptable step sizes. These step sizes are themselves optimized during the search according to a process called *self-adaptation*.

– By means of recombination and mutation, an offspring population $P'(t)$ of $\lambda \gg \mu$ candidate solutions is generated.

– The selection operator chooses the μ best solutions from $P'(t)$ (i.e., $Q = \emptyset$) as starting points for the next iteration of the loop. Alternatively, a (μ+λ)-evolution strategy would select the μ best solutions from the union of $P'(t)$ and $P(t)$ (i.e., $Q = P(t)$).

– The algorithm terminates if no more improvements are achieved over a number of subsequent iterations or if a given amount of time is exceeded.

– The algorithm returns the best candidate solution ever found during its execution.

In the following, these basic components of an evolution strategy are explained in some more detail. For extensive information about evolution strategies, refer to [Bäc96, Rec94, Sch95].

Using a more formal notation following the outline given in [SR95, SB97], one iteration of the strategy, that is a step from a population $P^{(T)}$ towards the next reproduction cycle with $P^{(T+1)}$, can be modeled as follows:

$$P^{(T+1)} := opt_{ES}(P^{(T)}) \tag{1}$$

where $opt_{ES} : I^\mu \to I^\mu$ is defined by

$$opt_{ES} := \mathbf{sel} \circ (\mathbf{mut} \circ \mathbf{rec})^\lambda \ , \tag{2}$$

operating on an input population $P^{(T)}$ according to

$$opt_{ES}(P^{(T)}) = \mathbf{sel}(P^{(T)} \sqcup \left(\sqcup_{i=1}^{\lambda} \{\mathbf{mut}(\mathbf{rec}(P^{(T)}))\} \right) \tag{3}$$

(here, $\sqcup$ denotes the union operation on multisets). Equation (3) clarifies that the population at generation $T + 1$ is obtained from P^T by first applying a λ-fold repetition of recombination and mutation, which results in an intermediate population P' of size λ, and then applying the selection operator to the union of $P^{(T)}$ and P'. Recall that the recombination operator generates only one individual per application, which can then be mutated directly.

In the following, both the formal as well as the informal way of describing the algorithmic components will be used as it seems appropriate.

2.2 The Structure of Individuals

For a given optimization problem

$$f : M \subseteq \mathbb{R}^n \to \mathbb{R} \quad , \quad f(\vec{x}) \to \min$$

an individual of the evolution strategy contains the candidate solution $\vec{x} \in \mathbb{R}^n$ as one part of its representation. Furthermore, there exist a variable amount (depending on the type of strategy used) of additional information, so-called *strategy parameters*, in the representation of individuals. These strategy parameters essentially encode the n-dimensional normal distribution which is to be used for the variation of the solution.

More formally, an individual $\vec{a} = (\vec{x}, \vec{\sigma}, \vec{\alpha})$ consists of up to three components $\vec{x} \in \mathbb{R}^n$ (the solution), $\vec{\sigma} \in \mathbb{R}^{n_\sigma}$ (a set of standard deviations of the normal distribution), and $\alpha \in [-\pi, \pi]^{n_\alpha}$ (a set of rotation angles representing the covariances of the n-dimensional normal distribution), where $n_\sigma \in \{1, \ldots, n\}$ and $n_\alpha \in \{0, (2n-n_\sigma)\cdot(n_\sigma-1)/2\}$. The exact meaning of these components is described in more detail in section 2.3.

2.3 Mutation

The mutation in evolution strategies works by adding a normally distributed random vector $\vec{z} \sim N(\vec{0}, \mathbf{C})$ with expectation vector $\vec{0}$ and covariance matrix $\mathbf{C}^{-1}$, where the covariance matrix is described by the mutated strategy parameters of the individual. Depending on the amount of strategy parameters incorporated into the representation of an individual, the following main variants of mutation and self-adaptation can be distinguished:

– $n_\sigma = 1$, $n_\alpha = 0$: The standard deviation for all object variables is identical (σ), and all object variables are mutated by adding normally distributed random numbers with

$$\sigma' = \sigma \cdot \exp(\tau_0 \cdot N(0,1)) \tag{4}$$
$$x_i' = x_i + \sigma' \cdot N_i(0,1) \quad , \tag{5}$$

where $\tau_0 \propto (\sqrt{n})^{-1}$. Here, $N(0,1)$ denotes a value sampled from a normally distributed random variable with expectation zero and variance one. The notation $N_i(0,1)$ indicates the random variable to be sampled anew for each setting of the index i.

– $n_\sigma = n$, $n_\alpha = 0$: All object variables have their own, individual standard deviation σ_i, which determines the corresponding modification according to

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \tag{6}$$
$$x_i' = x_i + \sigma_i' \cdot N(0,1) \quad , \tag{7}$$

where $\tau' \propto (\sqrt{2n})^{-1}$ and $\tau \propto (\sqrt{2\sqrt{n}})^{-1}$.

– $n_\sigma = n$, $n_\alpha = n \cdot (n-1)/2$: The vectors $\vec{\sigma}$ and $\vec{\alpha}$ represent the complete covariance matrix of the n-dimensional normal distribution, where the covariances are given by rotation angles α_j describing the coordinate rotations necessary to transform an uncorrelated mutation vector into a correlated one. The details of this mechanism can be found in [Bäc96]

(pp. 68–71) or [Rud92]. The mutation is performed according to

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \quad (8)$$

$$\alpha_j' = \alpha_j + \beta \cdot N_j(0,1) \quad (9)$$

$$\vec{x}' = \vec{x} + N(\vec{0}, \mathbf{C}(\vec{\sigma}', \vec{\alpha}')) \quad (10)$$

where $N(\vec{0}, \mathbf{C}(\vec{\sigma}', \vec{\alpha}'))$ denotes the correlated mutation vector and $\beta \approx 0.0873$.

The amount of information included into the individuals by means of the self-adaptation principle increases from the simple case of one standard deviation up to the order of n^2 additional parameters in case of *correlated mutations*, which reflects an enormous degree of freedom for the *internal models* of the individuals. This growing degree of freedom often enhances the global search capabilities of the algorithm at the cost of the expense in computation time, and it also reflects a shift from the precise *adaptation* of a few strategy parameters (as in case of $n_\sigma = 1$) to the exploitation of a large *diversity* of strategy parameters.

One of the main design parameters to be fixed for the practical application of the evolution strategy concerns the choice of n_σ and n_α, i.e., the amount of self-adaptable strategy parameters required for the problem.

2.4 Recombination

In evolution strategies recombination is incorporated into the main loop of the algorithm as the first variation operator and generates a new intermediate population of λ individuals by λ-fold application to the parent population, creating one individual per application from ϱ ($1 \leq \varrho \leq \mu$) individuals. Normally, $\varrho = 2$ or $\varrho = \mu$ (so-called global recombination) are chosen. The recombination types for object variables and strategy parameters in evolution strategies often differ from each other, and typical examples are *discrete recombination* (random choices of single variables from parents, comparable to uniform crossover in genetic algorithms) and *intermediary recombination* (arithmetic averaging). A typical setting of the recombination consists in using discrete recombination for object variables and global intermediary recombination for strategy parameters. For further details on these operators, see [Bäc96].

When $\mu > 1$ is chosen, the recombination operator needs also be specified for a (μ,λ)-evolution strategy .

2.5 Selection

Essentially, the evolution strategy offers two different variants for selecting candidate solutions for the next iteration of the main loop of the algorithm: (μ,λ)-selection and (μ+λ)-selection.

The notation (μ, λ) indicates that μ parents create $\lambda > \mu$ offspring by means of recombination and mutation, and the best μ offspring individuals are deterministically selected to replace the parents (in this case, $Q = \emptyset$ in algorithm 1). Notice that this mechanism allows that the best member of the population at generation $t + 1$ might perform *worse* than the best individual at generation t, i.e., the method is not *elitist*, thus allowing the strategy to accept temporary deteriorations that might help to leave the region of attraction of a local optimum and reach a better optimum. Moreover, in combination with the self-adaptation of strategy parameters, (μ,λ)-selection has demonstrated clear advantages over its competitor, the (μ+λ) method.

In contrast, the (μ+λ)-strategy selects the μ survivors from the union of parents and offspring, such that a monotonic course of evolution is guaranteed ($Q = P(t)$ in algorithm 1).

For reasons related to the self-adaptation of strategy parameters, the (μ,λ)-evolution strategy is typically preferred.

2.6 Termination Criterion

There are several options for the choice of the termination criterion, including the measurement of some absolute or relative measure of the population diversity (see e.g. [Bäc96], pp. 80–81), a predefined number of iterations of the main loop of the algorithm, or a predefined amount of CPU time or real time for execution of the algorithm.

3 Airfoil design

3.1 Introduction

Airfoil design provides a wealth of multi-criteria optimization problems. The layout of a wing heavily influences its efficiency regarding e.g. fuel consumption, etc. Efficiency of an airfoil design can not be measured independently of the anticipated use of the wing, since even the most efficient design must still be able to produce enough lift at low speeds to allow a plane to take off. Different flight conditions like starting and landing or cruising at high altitudes, induce different conditions for optimality. This naturally leads to the formulation of a multi-criteria optimization problem where each flight condition (flight point) states its own objective function. The airfoil design problem considered in this study and the resulting objective function are introduced in section 3.2 and 3.3. Section 3.5 introduces basic features of the *Strength Pareto Evolutionary Algorithm 2* (SPEA2) and the *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) which for this study represent the state of the art in evolutionary multi-criteria optimization.

3.2 The Design

One of the main characteristics of a wing design is its pressure profile, i.e. the distribution of pressure over the chord. The inverse design problem under study is to find the wing profile that produces a given pressure profile at given flow conditions. The test problem at hand was proposed for the AEROSHAPE (`http://aeroshape.cira.it`) project and consists of two target wing designs, namely the

standard NACA0012 wing at typical starting flow conditions and the standard NACA4412 wing at typical cruise flow conditions. The flow conditions for the two wings are given in table 1; the target wing designs and their respective pressure profiles are shown in figure 1.

	High Lift	**Low Drag**
Target wing	NACA0012	NACA4412
Mach number	0.2	0.77
Reynolds number	$5.1 \cdot 10^6$	10^7
Angle of Attack	$10.8°$	$1.0°$
c_w	$2.252 \cdot 10^{-2}$	$1.682 \cdot 10^{-2}$
c_a	1.252	0.5794

Table 1: Flow conditions for high lift with the NACA4412 wing and for low drag with the NACA0012 wing.

The optimization goal is the identification of a set of wing designs whose pressure distributions provide a certain performance for the lift off situation at the expense of cruise condition efficiency. This set is supposed to contain designs very similar to the NACA0012 design on the one hand and the NACA4412 design on the other hand as extreme solutions. Ultimately, an engineer would select one design from this collection of wing profiles that fits best to a given aeroplane concept where neither the standard NACA0012 nor the standard NACA4412 would be an optimal choice.

The rational behind using the NACA0012 and NACA4412 designs is to construct a test case that contains all major difficulties of fluid dynamics and its simulation, but at the same time produces verifiable and comprehensive results. In a real world application, the target pressure distribution may be given independently from a standard airfoil.

3.3 Objective Function

The pressure distribution $p(s)$ at position s of the chord is computed by solving the two-dimensional Navier-Stokes Equations. Together with the pressure p_∞, the density ρ_∞ and the speed $\vec{v}_\infty$ of the surrounding stream the value of

$$c_p(s) = \frac{p(s) - p_\infty}{\frac{\rho_\infty}{2} \cdot \vec{v}_\infty^2} \qquad (11)$$

is calculated, so that the two objective functions to be minimized are the cumulated squared differences between the actual pressure profile of the current configuration and the target pressure profiles:

$$f_1(\vec{x}) = \int_0^1 \left[c_p^{\vec{x}}(s) - c_p^{ld}(s)\right]^2 ds \qquad (12)$$

$$f_2(\vec{x}) = \int_0^1 \left[c_p^{\vec{x}}(s) - c_p^{hl}(s)\right]^2 ds \qquad (13)$$

where s is normalized by the chord length ($s \in [0,1]$), $c_p^{\vec{x}}$ is the pressure profile of the current design, c_p^{ld} is the low-drag pressure profile of the NACA4412 wing and c_p^{hl} is the high-lift pressure profile of the NACA0012 wing.

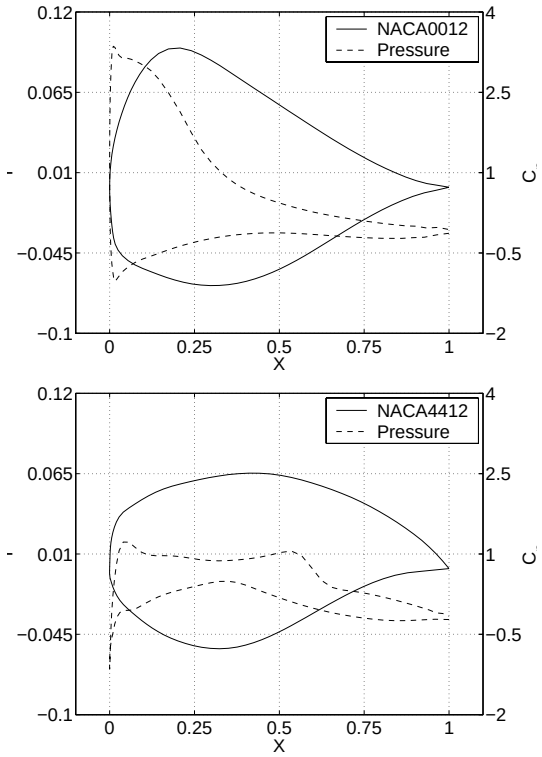


Figure 1: NACA0012 with pressure profile (top) and NACA4412 with pressure profile (bottom).

Since a single evaluation of the objective function is costly the total number of evaluations was restricted to 1000. Figure 2 give some indication of the quality of the problem: For both objective functions the function values are plotted against the first two design variable dimensions. It is quite clear from figure 2 that any optimization algorithm may easily get trapped in local optima.

3.4 Single-Criteria Optimization

The first goal is to re-design one of the extreme solutions introduced in 3.2. Here, we have chosen the NACA0012 wing for typical starting flow conditions. Thus, we need only the objective function 12.

Starting with a standard implementation of an evolution strategy [BFM97] different parameter settings have been studied. Different numbers of parents and offspring have been tested with standard evolution strategies using comma and plus selection schemes. Varying the proportion of parents to offspring directly influences the selection pressure of the algorithm and the goal of search. Decreasing the proportion leads to a smoother selection pressure and thus to more exploring the search space. On the other hand a larger proportion leads to a larger selection pressure and more exploiting the search space.

Furthermore 1 and n step sizes for the mutation have

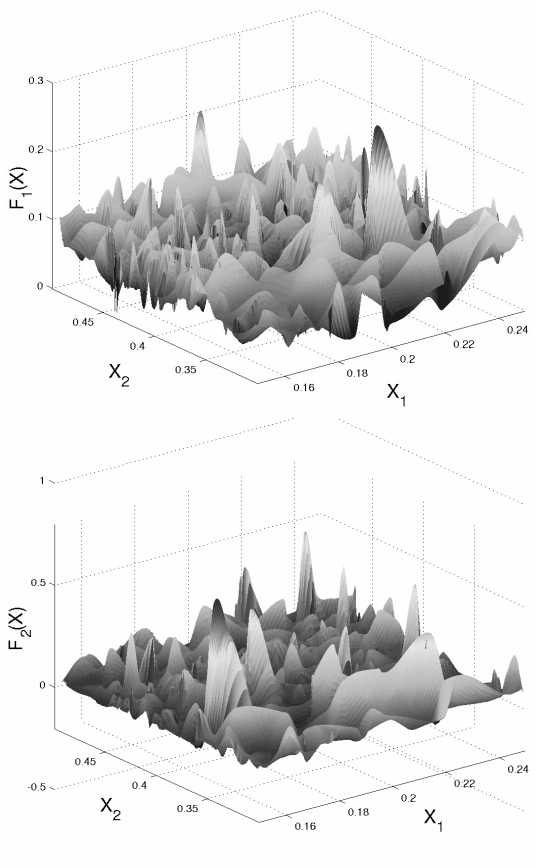


Figure 2: Objective function landscape of f_1 (top) and f_2 (bottom), projected on the first 2 dimensions.

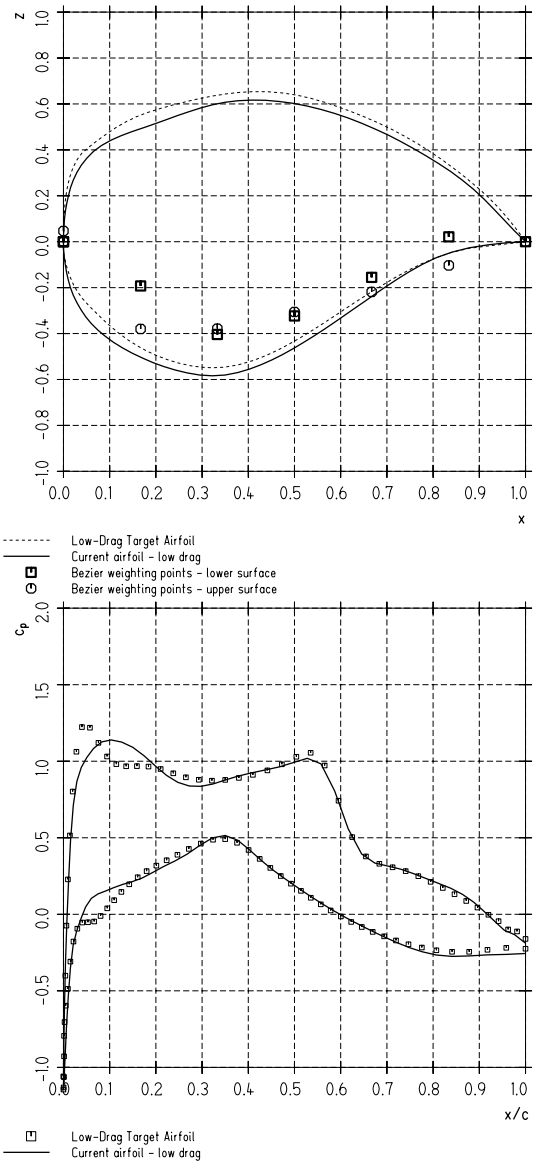


Figure 3: Airfoil shape (top) and pressure distribution (bottom) found with the ES variant

been tried to have one global step size or one step size in each direction of the search. Different kinds of recombination have been performed, especially intermediate recombination on the strategy parameters in combination with intermediate or discrete recombination on the object variables.

First results showed that the (5+20)-ES seems most promising for further investigations. Recombination was applied to the object variables in a discrete way and to the set of strategy parameters in an intermediate way.

Using derandomized mutation step size control [OGH94] the results could be further improved. This best ES variant outperformed the best values known so far in 90% of all tests.

The airfoil shape and pressure distribution of the best result achieved with the derandomized step size control mutation is presented in Fig. 3. In the left part of the figure the airfoil shape is shown and in the right part the pressure can be seen.

A major difference in the airfoil shape between the results so far and the ones presented here is hard to recognize. The experts in the field recognize the advantage of the ES variant in the lower surface. This advantage is much clearer visible in the pressure distribution. Here the pressure generated with the ES variant is much more in coincidence with the given target than the ones so far. This result is in deed 14 % better than the best known so far.

Another major difference is the ability to start the optimization process with randomly chosen individuals, i.e. pressure distributions and airfoil shapes. This fact will become very important when leaving the re-design testcase and looking for really new airfoil designs. Here starting with already predefined airfoils will narrow search to a specific search space area and maybe exclude the best solutions.

3.5 Multi-Criteria Optimization

The main problem in evolutionary multi-criteria optimization lies in the selection operator that chooses the parent

individuals for the next reproduction cycle. The single-criteria selection operators of $(\mu + \lambda)$ and (μ, λ) Evolution Strategies rely on the total order of scalar fitness values. Since in general, there is no such total order given for vector valued objective functions, it must be derived by additional selection criteria. SPEA2 and NSGA-II both prefer non-dominated individuals to dominated ones and they both try to establish evenly spread parent populations by preferring parents in sparsely populated regions of the objective function space.

The SPEA2 operator accumulates information on dominance relationships by summing so called strength values that are computed for an individual by counting the number of individuals it dominates. To enable comparison of individuals with identical strength count, a density value $0 \leq \rho \leq 1$ based on a k-nearest neighbor method is added that is low for sparsely populated regions of the objective function space. Additionally, a special "exclude worst" method based on smallest pairwise distances is applied, if a Pareto front must be reduced to a given size. The details of the algorithm can be found in [LZT01].

The NSGA-II operator uses the non-dominated sorting method to split a population in disjunct Pareto fronts, such that in each front individuals do not dominate each other. NSGA-II then adds new parent individuals front wise, starting with the global Pareto front. If the addition of the next Pareto front would yield more than the prescribed number of new parents, each individual of the current front is assigned a density value based on the city block distance of its closest neighbors. As with SPEA2, individuals from sparsely populated regions of the objective function space are preferred. Details on the methods used in NSGA-II can be found in [Deb01, DG01].

Since the computational cost of the CFD-Simulation does not allow numerous experiments even for the two-dimensional models, we show representative results of single runs from a number of experiments. We do not try to average several runs in any way, because the small number of experiments that could be conducted does not allow meaningful statistics. Additionally, from visual inspection, the runs made did not significantly differ from each other.

The pareto-fronts displayed in figures 4, 5 and 6 contain a reference Pareto front (denoted "Reference") that is the common Pareto front of all optimization experiments that were conducted for this study and results computed with a multi-criteria genetic algorithm (MOGA) as described in [Pol99]. The reference set was included to show the quality of a single run optimization compared to an aggregated pareto-set that needs much more fitness function evaluations and to supply a benchmark line the individual algorithms have to approach.

Figure 4 demonstrates that SPEA2 and NSGA-II are closer to the reference set than the MOGA. SPEA2 is slightly better in reconstructing the NACA4412 profile, while NSGA-II is slightly more successful for the NACA0012 profile. In the compromise region there is hardly a difference between SPEA2 and NSGA-II.

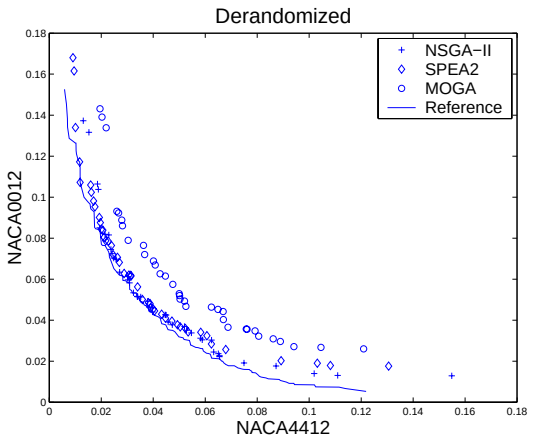


Figure 4: Pareto front with derandomized mutation and NSGA-II- and SPEA2-Selection.

Figure 5 displays a rather surprising result: The SPEA2 selection operator in combination with the pooling mutation [SS96] converges close to the reference pareto-set in the compromise region, but fails to place good solutions at the tails of the reference set. The NSGA-II selection operator, contrarily, has a better spread of solutions when combined with pooling mutation, while failing to closely approach the reference pareto-set. But still, both Evolution Strategies outperform the MOGA.

Figure 6 finally shows that both selection methods produce nicely spread solutions when combined with Schwefel's mutation [NWH$^+$00] which are not as close to the reference set as with the derandomized mutation. Once again, the pareto-set produced by the MOGA is worse than the Evolution Strategies' solutions.

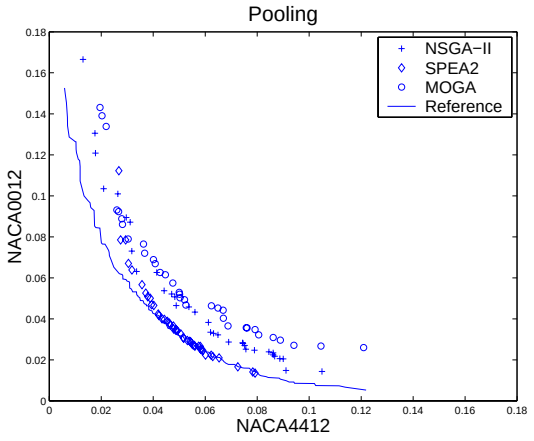


Figure 5: Pareto front with pooling mutation and NSGA-II- and SPEA2-Selection.

It is difficult to make serious judgements on the basis of the few data available. Comparing the results available from the experiments, the derandomized mutation operator seems to work very well, but neither Schwefel's mutation nor the pooling mutation are clearly inferior. Clarification

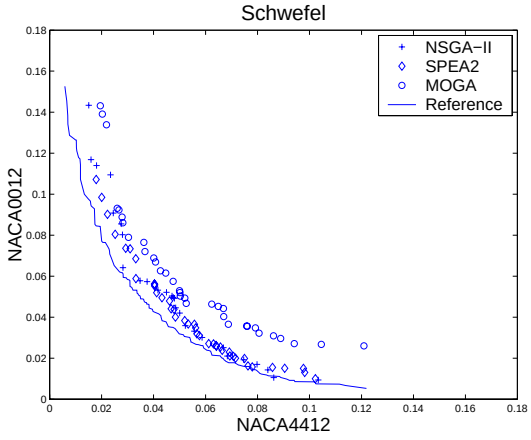


Figure 6: Pareto front with Schwefel's mutation and NSGA-II- and SPEA2-Selection.

of the question which mutation operator to choose will involve further tests with extended computational effort that was beyond the scope of this study.

The same statement holds for the choice of selection method. The results with the pooling mutation as shown in figure 5, where the NSGA-II selection provides better spread of solutions than the SPEA2 selection and the SPEA2 selection converges closer to the reference set than the NSGA-II selection, should be considered with some suspicion, as there seems to be no obvious reason for this behavior that could be attributed to the selection method. In fact the effect was less obvious in the other experiments with pooling mutation. For the airfoil design problem, as stated in this study, SPEA2 and NSGA-II perform comparably well.

It can clearly be seen from the results, though, that using NSGA-II or SPEA2 in combination with an Evolution Strategy yields better performance than using the MOGA as described in [Pol99]. This result is supported by all experiments that were made for the airfoil design problem.

4 Conclusions

As demonstrated by numerical experiments on scientific test functions (see e.g., [Sch95]), evolution strategies yield very good optima in case of nonlinear, high-dimensional global optimization problems. The self-adaptation of strategy parameters (i.e., variances and covariances of the normally distributed mutation operator) is an important and mandatory component of the algorithm to achieve this quality as a global optimization method. Self-adaptation is clearly the most distinguishing feature of evolution strategies, and has been increasingly studied and accepted by the evolutionary computation community over the past decade.

In many practical applications to engineering optimization problems, evolutionary algorithms have been widely neglected for a quite long period of time, mostly because the number of function evaluations - i.e., simulator calls - required seemed unacceptable to practitioners working with these simulators. Given the fact that simulator runtimes can range from several minutes to several hours, and the short project cycles in industry, this seems completely understandable.

Evolution strategies with self-adaptation, however, can be used with very small population sizes (e.g., a (1, 7)-strategy) and few generation cycles, such that they can achieve the desired balance between minimization of resource utilization (i.e., simulator calls) and finding an optimum as good as possible. Indeed, the application to airfoil design problems as discussed in this paper, with 10^3 simulator calls, by far is not representative of the extreme end of that spectrum: At NuTech Solutions, we are currently using evolution strategy variants with as few as around 10^2 simulator calls for high-dimensional problems (around 100 to 150 design parameters), and the results beat any of the optimizers that have been used previously by clients of NuTech Solutions.

These variants of evolution strategies, being under constant further development (both from an academic and a commercial point of view), exploit methods such self-adaptation and meta-modelling and are clearly defining the state-of-the-art in global optimization under tight resource constraints.

References

[Bäc96] Th. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.

[BFM97] Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.

[BHS97] Th. Bäck, U. Hammel, and H.-P. Schwefel. Evolutionary computation: History and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.

[Deb01] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, New York, 2001.

[DG01] Kalyanmoy Deb and Tushar Goel. Controlled elitist non-dominated sorting genetic algorithms for better convergence. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001)*, pages 67–81, 7–9 March 2001.

[LZT01] Marco Laumanns, Eckart Zitzler, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. TIK-Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, May 2001. -.

[NWH+00] B. Naujoks, L. Willmes, W. Haase, Th. Bäck, and M. Schütz. Multi-point airfoil optimization using evolution strategies. In *ECCOMAS 2000, European Congress on Computational Methods in Applied Sciences and Engineering*, 2000.

[OGH94] A. Ostermeier, A. Gawelczyk, and N. Hansen. Step-size adaptation based on non-local use of selection information. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature — PPSN III International Conference on Evolutionary Computation*, volume 866 of *Lecture Notes in Computer Science*, pages 189–198. Springer, Berlin, 1994.

[Pol99] Carlo Poloni. Multi objective aerodynamic optimisation by means of robust and efficient genetic algorithm. In Kozo Fujii and George S. Dulikravich, editors, *Recent development of aerodynamic design methodologies : inverse design and optimization*, volume 68 of *Notes on numerical fluid mechanics*, pages 1–24. Vieweg, Braunschweig/Wiesbaden, 1999.

[Rec94] I. Rechenberg. *Evolutionsstrategie '94*, volume 1 of *Werkstatt Bionik und Evolutionstechnik*. Frommann–Holzboog, Stuttgart, 1994.

[Rud92] G. Rudolph. On correlated mutations in evolution strategies. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 105–114. Elsevier, Amsterdam, 1992.

[SB97] H.-P. Schwefel and Th. Bäck. Artificial evolution: how and why ? In D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms in Engineering and Computer Science*, chapter 1, pages 1–19. Wiley, Chichester, 1997.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.

[SR95] H.-P. Schwefel and G. Rudolph. Contemporary evolution strategies. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Advances in Artificial Life. Third International Conference on Artificial Life*, volume 929 of *Lecture Notes in Artificial Intelligence*, pages 893–907. Springer, Berlin, 1995.

[SS96] M. Schütz and J. Sprave. Application of parallel mixed-integer evolution strategies with mutation rate pooling. In L.J. Fogel, P.J. Angeline, and T. Bäck, editors, *Proceedings of the 5th Annual Conference on Evolutionary Programming (EP-96), San Diego, CA, 29. February - 2. March*, pages 345–354, 1996.

On the Extinction of Evolutionary Algorithm Subpopulations on Multimodal Landscapes

Lutz Schönemann
Systems Analysis Group,
University of Dortmund, Germany
lutz.schoenemann@cs.uni-dortmund.de

Michael Emmerich
Leiden Institute for Advanced Computer Science
University of Leiden, Netherlands
emmerich@liacs.nl

Mike Preuss
Systems Analysis Group,
University of Dortmund, Germany
mike.preuss@cs.uni-dortmund.de

Keywords: Evolutionary algorithms, multimodal fitness landscape, niching techniques, random genetic drift

Received: July 17, 2004

Population based evolutionary algorithms (EA) are frequently used to optimize on multimodal functions. A common assumption is that during search several subpopulations might coexist in different attraction regions of the search space. Practical experience and takeover–time considerations suggest that this is not true in general. We therefore analyze the stability of subpopulations within a simplified EA on a two-attractor model, focusing on two extreme cases: (1) Function values of both local minima are exactly the same and (2) function values on the first attractor are always better than on the second. Realistic scenarios for bimodal optimization are assumed to be located in between these two extremes, such that upper and lower bounds for extinction times can be estimated, e. g. by Markov chain analysis and empirical studies. The obtained results provide new insights into the effect of $(\mu \dot{+} \lambda)$ selection on the stability of subpopulations and the effect of genetic drift. Moreover, the effect of idealized niching on the same scenarios is investigated, leading to an immense increase of the EA's ability to perform concurrent search. Our model and the findings based thereupon do not depend on the number of problem dimensions.

Povzetek: Članek podaja analizo stabilnosti podpopulacij v evolucijskem algoritmu za optimiranje funkcij z več lokalnimi ekstremi.

1 Introduction

EA are preferable tools for optimization on multimodal functions. It has often been assumed that the strength of EA stems from the ability to search concurrently in different high performance regions of the search space. Contrary to this, experimental results on multimodal function optimization suggest that EA using panmictic $(\mu \dot{+} \lambda)$ selection tend to rapidly concentrate on a single attractor, even if all optima have the same size and function values. Without employing niching techniques, it seems impossible to maintain individuals in different regions at the same time.

In this paper we trace back the effect of extinction on neutral landscapes to random genetic drift dynamics, which can be observed in a simplified scenario. Though, dealing with an theoretical issue, the paper provides valuable insights for the practitioner on how to design niching techniques, when searching on multimodal landscapes.

The paper is structured as follows: First, we demonstrate the effect on a simple test-case (two-sphere model) (Sect. 2). Second, we ascribe extinction to random genetic drift dynamics that can be reproduced and analyzed with a simple Markov model which is set up and analyzed in Sect. 3. Based on the theoretical observation we motivate a design principle for niching techniques and demonstrate its benefit on the test case (Sect. 4).

2 Population Dynamics on a Bimodal Test-Case

As an example, the extinction of subpopulations has been observed for the minimization of a simple two-sphere prob-

lem (cf. Figure 1)

$$f(\mathbf{x}) = \min((x_1 - 1)^2, (x_1 + 1)^2) + x_2^2 + \ldots + x_n^2 \ . \quad (1)$$

Note that this model is defined for variable problem dimensions n. However, to enable empirical testing, we fixed $n = 2$. The only required assumption for the argumentation following in Sect. 3 is that exactly two attractors exist. In further studies it may be interesting to investigate the case of more than two attractors.

Algorithm 1 describes the $(\mu \dot{+} \lambda)$-EA [2] that will be studied in this paper. Let $\mathbb{I}$ define the individual space. Each individual $a \in \mathbb{I}$ consists of information on its position in the search space and its objective function value. Furthermore, let $P_t \in \mathbb{I}^\mu, Q_t \in \mathbb{I}^\lambda$ and $M_t \in \mathbb{I}^\nu$ denote multisets of individuals (or *populations*) with $\nu = \mu + \lambda$ for the $(\mu + \lambda)$ selection and $\nu = \lambda$ for the (μ, λ) selection. P_t will be termed the *parent populations*, while Q_t will be termed the *offspring population* for $t = 0, \ldots, t_{\max}$. The EA starts with initializing the population of parents P_t

Algorithm 1

$t \leftarrow 0$
$P_t \leftarrow \text{init}()$ /* Initialize population $P_t \in \mathbb{I}^\mu$*/
while $t < t_{\max}$ **do**
  $Q_t \leftarrow \text{gen}(P_t)$ /* Generate offspring */
  $M_t \leftarrow \begin{cases} Q_t & \text{for } (\mu, \lambda) \text{ selection} \\ Q_t \cup P_t & \text{for } (\mu + \lambda) \text{ selection} \end{cases}$
  $P_{t+1} \leftarrow \text{sel}(M_t)$ /* Select new parents */
  $t \leftarrow t + 1$
end while

in the individual space $\mathbb{I}$. Then the following procedure is repeated while the generation counter does not exceed a user defined maximum $t_{\max}$: Generate a multiset of λ offspring by means of variation operators (usually recombination and mutation), then select the best μ individuals out of M_t. Here $M_t = Q_t$ in case of (μ, λ) selection and $M_t = Q_t \cup P_t$ in case of $(\mu + \lambda)$ selection. Finally increase the generation counter and jump to the beginning of the loop.

We make the convention that in case of equal objective function values for M_t, $\text{sel}(M_t)$ randomly draws k out of the M_t individuals, without choosing one of the individuals twice and without preferring the offspring in case of $(\mu+\lambda)$ selection.

Figure 2 shows the average extinction time and probabilities for the $(\mu+\lambda)$-EA. It can be observed that the takeover probability of one subpopulation grows proportionally with its ratio in the initial population.

3 Markov Model for the Extinction Dynamics

Imagine an objective function (for minimization) with two large local optimal regions with equal or slightly different

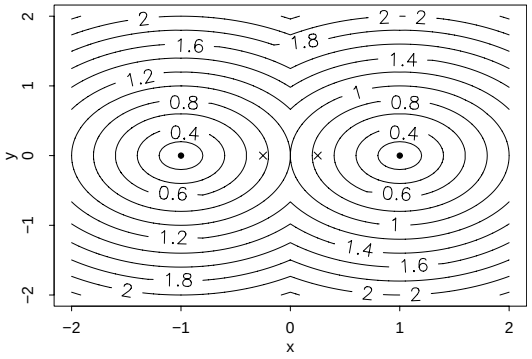


Figure 1: Two-sphere model: Crosses mark starting points for the black and white subpopulations, respectively.

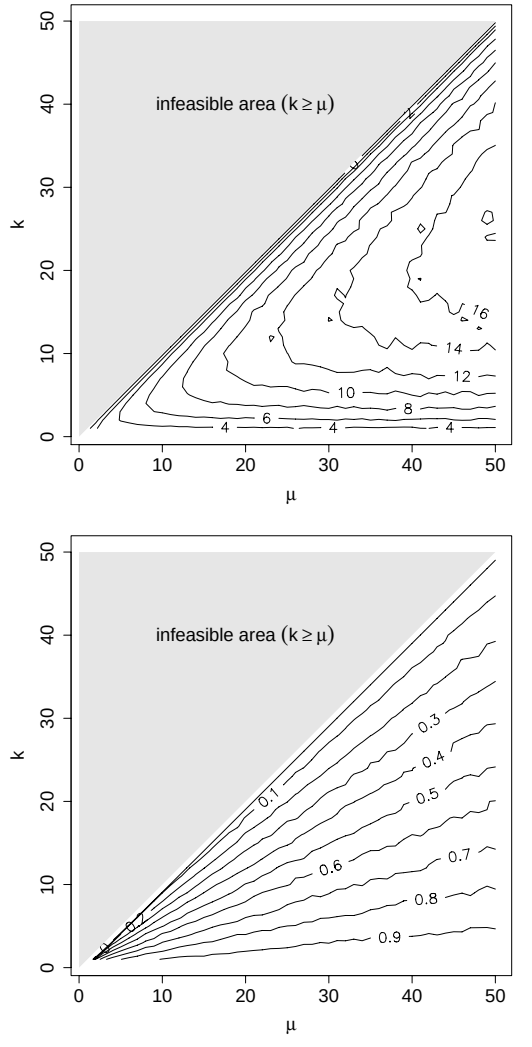


Figure 2: Extinction times (left) and probabilities (right) for a $(\mu + 7\mu)$-ES with Gaussian mutation on the two-sphere model (with 2 dimensions) averaged from 5000 runs. The number of black individuals in the population of size μ is denoted by k.

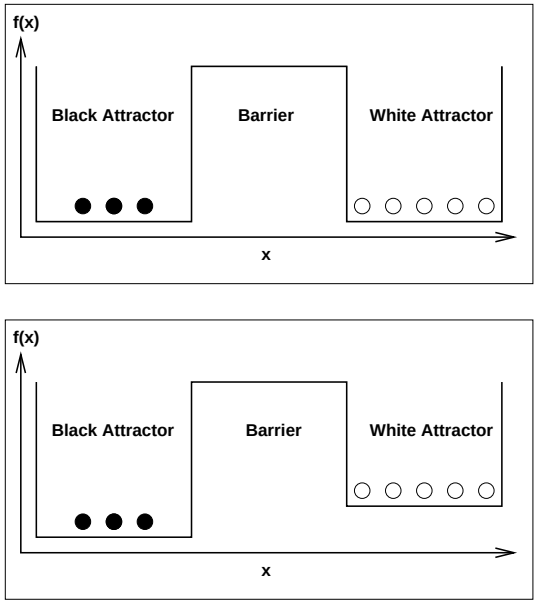


Figure 3: Schematic draw of the instantiations of the *two-attractor model*. The left figure describes the case with equal function values in both attractor basins and the right figure describes the case of better function values for the black individuals than that for the white individuals.

optimal function values. In between these plateaus there is a large barrier with extremely high function values (Fig. 3), such that it is very improbable that an individual from one area crosses the barrier by a single mutation. This is similar to the case that the optimization has reached the bottom of two equal or similar local optima of a bimodal function with flat bottoms.

In order to simulate the dynamics of the $(\mu \overset{+}{,} \lambda)$-EA on such a system, let us define the following rules of the game:

For a population P_t at a time t let black(P_t) define the number of individuals on the first attractor (we will call them *black individuals*). Accordingly, $\mu - \text{black}(P_t)$ individuals are located on the second attractor (we call them *white individuals*). Furthermore, let us assume that all individuals on an attractor have the same function value. Individuals cannot move from one attractor to another attractor or leave their attractor by means of mutation. Hence, the reduced mutation operator simply results in a copy of the individual.

Assuming an initial population with a specified number of black individuals, we are now interested in the dynamics of the simplified EA for the case that (1) the function value for both plateaus is equal and (2) the function value for the plateau that contains the black individuals is better than the function value on the plateau that contains the white individuals. Markov chain analysis can be a powerful tool for understanding simple models of evolution [3, 5]. Next, we provide the reader with the derived Markov chain model.

3.1 Deriving the Transition Probabilities

Let k denote the number of black individuals in the initial population. Then we are interested in the probability $p_j(k)$ for j black individuals in the subsequent population. This can be obtained by dividing the EA into two steps - the generation of individuals and the selection of P_{t+1}. A possible generational transition could be described as

$$P_t = \{\underbrace{\bullet,\ldots,\bullet}_{k},\underbrace{\circ,\ldots,\circ}_{\mu-k}\}$$

$$\overset{\text{generate}}{\longrightarrow} \quad Q_t = \{\underbrace{\bullet,\ldots,\bullet}_{l},\underbrace{\circ,\ldots,\circ}_{\lambda-l}\} \tag{2}$$

$$\overset{\text{replace}}{\longrightarrow} \quad P_{t+1} = \{\underbrace{\bullet,\ldots,\bullet}_{j},\underbrace{\circ,\ldots,\circ}_{\mu-j}\} \tag{3}$$

Then the transition matrix of the whole evolution step reads:

$$\mathbf{P} := (p_{k,j})_{k\in\{0,\ldots,\mu\},j\in\{0,\ldots,\mu\}}\,, \tag{4}$$

with

$$p_{k,j} = \sum_{l=0}^{\lambda} p_l^{\text{gen}}(k)\cdot p_j^{\text{sel}}(l,k)\ . \tag{5}$$

Here $p_l^{\text{gen}}(k)$ describes the transition probabilities of the procedure gen(P_t)

$$p_l^{\text{gen}}(k) = \Pr(\text{black}(Q_t) = l|\text{black}(P_t) = k) \tag{6}$$

and $p_j^{\text{sel}}(l,k)$ describes the transition probabilities for the procedure sel(M_t)

$$p_j^{\text{sel}}(l,k) = \Pr(\text{black}(P_{t+1}) = j|\text{black}(P_t) = k \wedge \text{black}(Q_t) = l)\ . \tag{7}$$

The transition probabilities $p_l^{\text{gen}}(k)$ for the generate function are the same for all selection schemes studied here:

$$p_l^{\text{gen}}(k) = \left(\frac{k}{\mu}\right)^l \cdot \left(\frac{\mu-k}{\mu}\right)^{\lambda-l} \cdot \binom{\lambda}{l}\ . \tag{8}$$

Table 1 shows the transition probabilities that are instantiated for different selection methods and assumptions about the function values on the two attractors.

3.2 Markov Chain Analysis

Now, we can apply Markov chain analysis [7] in order to analyze the dynamics of the system. Recall from probability theory, for $t > 0$ and any given state vector $\mathbf{p}_t$ we can calculate the probability distribution for the resulting subsequent state j by means of $\mathbf{p}_{t+1} = \mathbf{P}\cdot\mathbf{p}_t$. The limit value for $\mathbf{p}_t$ as $t \to \infty$ can be obtained by means of the fundamental matrix:

Table 1: Selection probabilities $p_j^{sel}(l,k)$ for the $(\mu+\lambda)$ selection (top) and the (μ,λ) selection (bottom) (I is the indicator function).

equal function values	black better than white
$\frac{\binom{k+l}{j}\cdot\binom{\mu+\lambda-k-l}{\mu-j}}{\binom{\mu+\lambda}{\mu}}$	$I(j=\min(\mu,k+l))$
$\frac{\binom{l}{j}\cdot\binom{\lambda-l}{\mu-j}}{\binom{\mu+\lambda}{\mu}}$	$I(j=\min(\mu,l))$

The Markov process of the two-attractor model has absorbing boundaries $k=0$ and $k=\mu$. If one of the absorbing states has been reached the system remains stable. The absorption probabilities and mean absorption times correspond to the extinction probabilities and mean extinction times. Both can be derived from the transition matrix $\mathbf{P}$ and the initial state k_0. First, let us partition the transition matrix as follows:

$$\mathbf{P} = \begin{pmatrix} 1 & \mathbf{0}_{\mu-1} & 0 \\ \mathbf{A}^t & \mathbf{C} & \mathbf{B}^t \\ 0 & \mathbf{0}_{\mu-1} & 1 \end{pmatrix} . \tag{9}$$

where $\mathbf{A} := (a_k)_{k\in\{1,\ldots,\mu-1\}}$, $\mathbf{B} := (b_k)_{k\in\{1,\ldots,\mu-1\}}$, $\mathbf{C} := (c_{k,j})_{k\in\{1,\ldots,\mu-1\},j\in\{1,\ldots,\mu-1\}}$, and $\mathbf{0}_{\mu-1}$ is the $(\mu-1)$-dimensional 0-vector.

Now, the fundamental matrix $\mathbf{T}$ of the transition matrix $\mathbf{P}$ reads:

$$\mathbf{T} := (\mathbf{I}-\mathbf{C})^{-1} , \tag{10}$$

and from Markov chain theory ([7], Chap. 3) an expression for the extinction of black individuals, i. e. for reaching the absorbing state $k=0$, can be derived as

$$p_E(k_0) = \sum_{i=1}^{\mu-1} a_{i,1} t_{k_0,i}, \quad k_0 = 1,\ldots,\mu-1 \tag{11}$$

It is also known that $t_{i,j}$ of $\mathbf{T}$ equals the mean number of iterations that the system is in state i when starting in state j before absorption takes place. Thus

$$E(k_0,\mu) = \sum_{i=1}^{\mu-1} t_{k_0,i}, \quad k_0 = 1,\ldots,\mu-1 \tag{12}$$

is the mean absorption time, or - translated to our model - the average time that two subpopulations in the evolutionary system can coexists when working with the generational transition described by $\mathbf{P}$ and starting with k individuals.

4 Analysis of Selection Mechanisms

Now, we can use the Markov chain techniques proposed in the previous section to determine some characteristics of selection mechanisms on the two-attractor model. We start with the case of equal fitness.

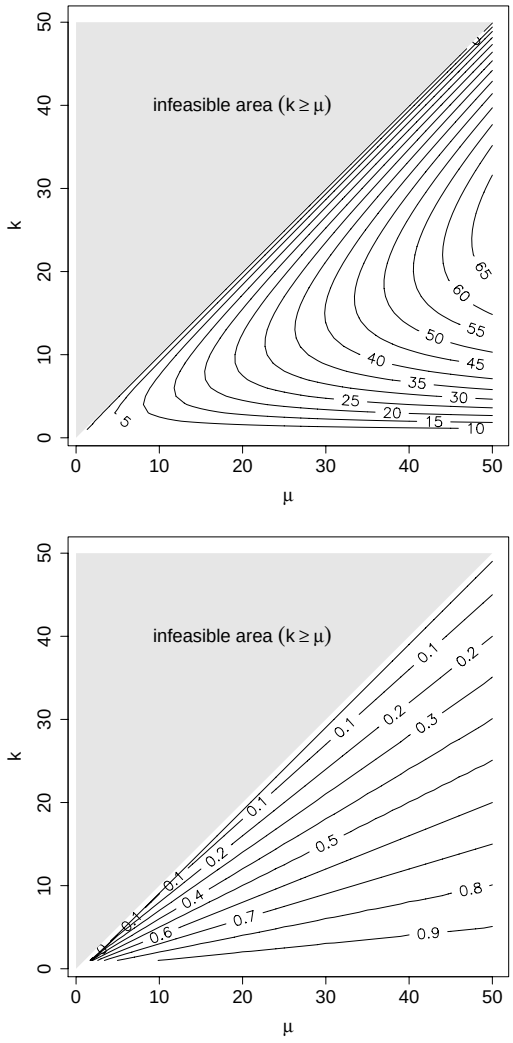


Figure 4: Expected extinction times (top) and probabilities (bottom) for a $(\mu+7\mu)$-EA obtained by Markov theory.

Figures 4 and 5 show the expected extinction times (EET) and probabilities of extinction (PE) for some frequently used EA strategies. The figure reveals that extinction times increase linearly with a growing μ if λ and k are set to a constant proportion of μ. Note, that extinction times are measured in generations. In the case of $(\mu+7\mu)$ selection and $\mu=40, k=20$, this means that one population dies out on average after 53 generations or about $15,000$ offspring. Contrary to this, for the $(\mu+1)$ selection and the same settings for μ and k only 550 offspring are generated until one population dies out.

In addition to the expected extinction time we are interested in the probability of extinction for one population. A closer look at the underlying data of the bottom diagram in Fig. 4 instantly leads to the following conjecture: *The extinction probability is defined by* $\frac{k}{\mu}$. This is an astonishing

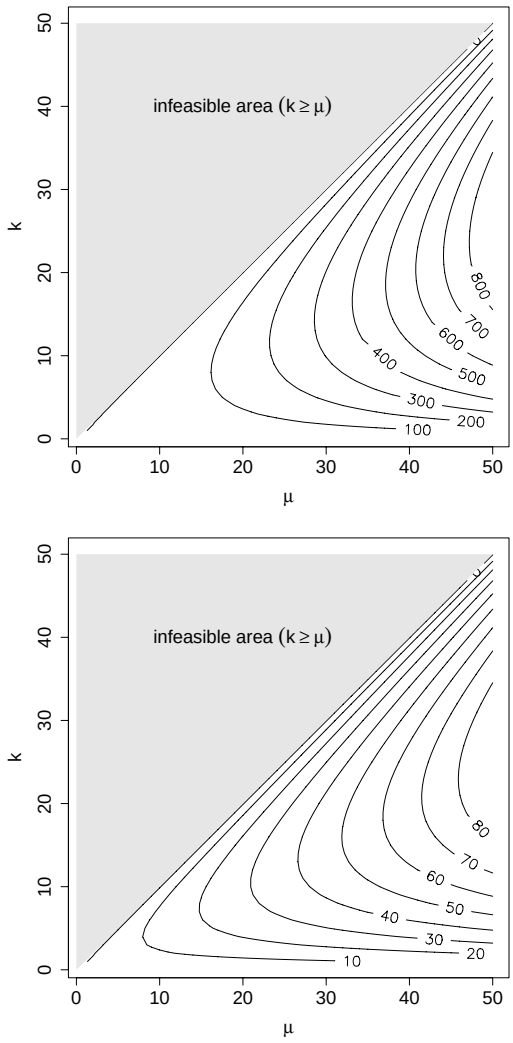


Figure 5: Expected extinction times for a $(\mu + 1)$-EA (top) and a $(\mu + \mu)$-EA (bottom).

simple formula, and, moreover, it is especially independent from λ.

Investigating the model of equal function values gives fundamental insights into the behavior of EA on a bimodal fitness landscape. But it is assumed that a real EA will produce different offspring on a two-attractor landscape. Therefore, we study the two-sphere model depicted in Fig. 1 as a more realistic case.

For this function, a Markov chain analysis can not be applied directly. Hence, we obtain the results presented in Fig. 2 by a monte carlo simulation with a real EA. In order to prevent an acceleration of the extinction process caused by recombination, the EA shall apply variation only by mutation. This mutation operator adds a normally distributed random variate to the object variables. Small mutation strengths (step sizes) assure that no individual can jump from one attractor to the other.

The outcome documents that due to the stochastic mutation the extinction times are smaller than in the former model. This is an expected result because this scenario is lying in between the two models of equal and different function values. In contrast to this, the probabilities of extinction are the same. Note that although we did most of the simulations on the 2 dimensional two-sphere model, futher tests indicate that the resulting EA behavior does not differ much for higher dimension counts, as predicted by the more general Markov model.

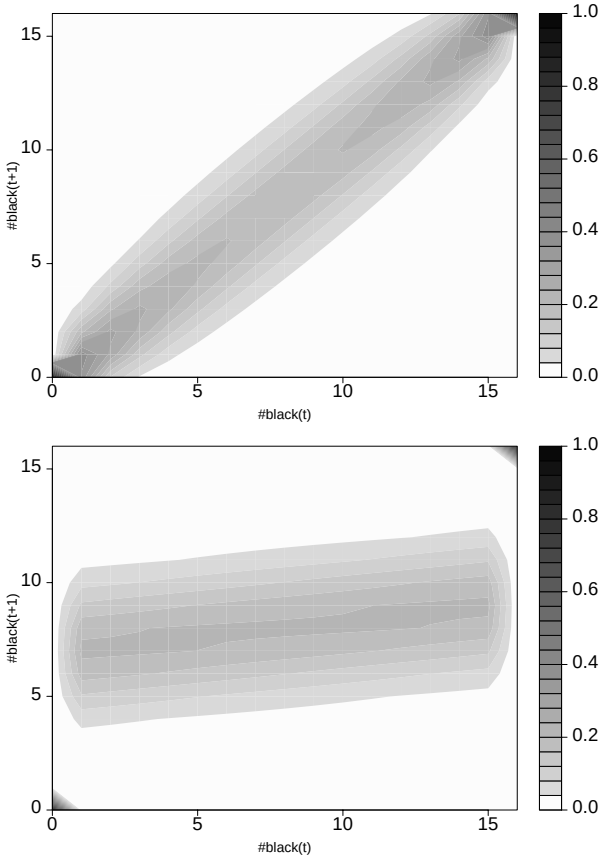


Figure 6: Visualization of transition matrices for a $(16 + 112)$-EA as obtained from the Markov model. Top: Without niching, transition probability to an absorbing state (black subpopulation counts 0 or 16 individuals) increases towards the borders, bottom: with recombinative niching, transition to an absorbing state is highly unlikely.

Our investigations show that — due to the effect of genetic drift — one subpopulation will die out quickly even if both have similar fitness values. To guarantee the survival of a potentially fitter subpopulation and to prevent the extinction of subpopulations located on attractors of similar quality, one could use several techniques. One of it is niching [4, 6, 8]. Thus, some of the former experiments were repeated with a simple niching technique we name *recombinative niching*: Attraction areas to which individuals belong are identified by some clustering approach (cf. [9]) and we generate the same number of offspring individuals for each subpopulation. In contrast to many other

Table 2: Illustrative cases for expected extinction times (EET) and probabilities (PE).

Strategy	Model	Niching	k	EET	PE
(16,112)	boolean	-	1	2.0	0.007
(16,112)	neutral	-	1	6.7	0.94
(16,112)	neutral	x	1	$1.1 \cdot 10^5$	0.5
(16,112)	boolean	x	1	1.0	0
(4,28)	neutral	x	1	10.2	0.5
(50+1)	boolean	-	1	224.0	0
(16,112)	two-sph.	x	1	$9.8 \cdot 10^4$	0.48
(16+112)	two-sph.	x	1	$> 10^6$	0.52

niching techniques, the suggested selection process is panmictic (alternative selection schemes are presented in [1, Sect. C2]). It was observed that using this kind of simple niching, the subpopulations were able to coexist for a much longer time (Table 2, Fig. 6), even for the more realistic example of the two-sphere function. Hence, the results in this paper affirm that for optimization on multimodal landscapes niching is a preferable technique to avoid the loss of information gained by subpopulations [9], even in the presence of equality.

The results for some selected test cases are shown in Table 2. It contains the mean extinction times as well as the extinction probabilities. The cases were chosen because they reflect some frequently observed situations and provide the reader with some borderline cases. The mean extinction time shows the number of reproduction cycles both subpopulations survive. In contrast to this PE measures the probability that the subpopulation consisting of black individuals dies out. Whereas the results of the first six strategies were got by a Markov Chain analysis, the results for the two-sphere model were reached by experiments. Some remarks need to be spend on the results of the (16+112)-ES on the two-sphere model. Due to time limitations the runs were limited to 10^7 generations. Until then, in only 15% of all runs one population died out. On average, a single run lasts longer than 10^6 generations. In the case that one population died out, the black subpopulation was eliminated in 52%.

5 Conclusions

By means of this paper a better understanding of the extinction process of subpopulations on multimodal landscapes has been achieved. In detail our investigations give evidence for the following conjectures:

– For $(\mu \dot{+} 7\mu)$-EA and the equal fitness model the extinction time grows linearly with μ. Thus, even in the best-case scenario, co-existence of populations will not occur for a long time. This result can be interpreted also in a way that mating restrictions alone will not suffice to prevent subpopulations from extinction. This is because in the studies on the simplified model, the recombination has been omitted and thus it cannot accelerate the extinction of subpopulations.

– For different function values and comma selection it was observed that better individuals survive with probability near 1. Hence, the effect of random genetic drift unlikely biases the direction of evolution, if fitness values are clearly different.

– The extinction time for a $(\mu + 1)$-EA are long even in the case of equal function values. However, if we regard the number of function evaluations as a criterion for the extinction time, proportions change and the mean extinction time for the $(\mu + 1)$-EA is significantly smaller than for an EA with birth surplus.

– The extinction time is increased substantially by using the suggested niching technique.

In this paper only the bimodal case has been considered. However, we conjecture that similar results can be obtained for landscapes with more than two attractors. Future research will have to clarify this point. Furthermore, the effect of the recombination operator deserves further attention.

Acknowledgments This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the *Collaborative Research Center* "Computational Intelligence" (SFB 531).

References

[1] Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: Handbook of Evolutionary Computation. Oxford University Press, New York, and Institute of Physics Publishing, Bristol (1997) Release 97/1.

[2] Beyer, H.–G., Schwefel, H.–P.: Evolution strategies: A comprehensive introduction. Natural Computing **1** (2002) 3–52

[3] Chakraborty, U., Deb, K., Chakraborty, M.: Analysis of selection algorithms: A markov chain approach. Evolutionary Computation **2** (1996) 133–167

[4] Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In Schaffer, J.D., ed.: Proc. 3rd Int'l Conf. on Genetic Algorithms ICGA 89, San Mateo, CA, Morgan Kaufmann (1989) 42–50

[5] Goldberg, D.E., Segrest, P.: Finite markov chain analysis of genetic algorithms. In Grefenstette, J.J., ed.: Proc. 2nd Int'l Conf. on Genetic Algorithms ICGA 87, L. Erlbaum Associates (1987) 1–8

[6] Horn, J.: Finite markov chain analysis of genetic algorithms with niching. In Forrest, S., ed.: Proc. Fifth Int'l Conf. on Genetic Algorithms ICGA 93, San Mateo, CA, Morgan Kaufmann (1993) 110–117

[7] Kemeny, J.G., Snell, J.L.: Finite Markov chains. D. Van Nostrand Company, Ltd., London (1969)

[8] Mahfoud, S.W.: Niching Methods for Genetic Algorithms. PhD thesis, University of Illinois at Urbana Champaign (1995)

[9] Streichert, F., Stein, G., Ulmer, H., and Zell, A.: A Clustering Based Niching EA for Multimodal Search Spaces. Proceedings of the 6th International Conference on Artificial Evolution, Marseille, France, October 27-30, 2003, p. 169-180

Optimizing Production Schedules and Energy Consumption with an Evolutionary Algorithm

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
E-mail: bogdan.filipic@ijs.si

Keywords: evolutionary algorithm, production scheduling, energy consumption, peak energy demand, cost minimization, automobile industry

Received: November 15, 2004

We present an evolutionary algorithm approach to schedule optimization for a group of production lines in a car factory. Schedules are evaluated with respect to the energy consumption over peak demand periods, while the task is to minimize the energy costs by appropriately scheduling the interruptions of processes on the lines. Tests on real problem instances show this approach gives near-optimal schedules in acceptable time.

Povzetek: v članku opisujemo učinkovito sestavljanje urnikov obratovanja z evolucijskim algoritmom in optimiranje porabe energije na proizvodnih linijah v avtomobilski tovarni.

1 Introduction

Scheduling deals with allocating activities to resources over time in such a way that given objectives are optimized, while temporal constraints and resource limitations are satisfied. Problems of this kind appear in manufacturing, timetabling, vehicle routing, design of computer operating systems and other domains. Because of its great practical importance, scheduling has permanently attracted research interests. Following the attempts in the fields of Operations Research and Artificial Intelligence with limited success in practice, Evolutionary Computation [1] has recently offered means of producing near-optimal schedules for complex problems at reasonable computational costs [3]. A number of applications of evolutionary algorithms in scheduling have been reported [2, 6, 11]. Nevertheless, there are still open issues to be addressed in the development of evolutionary scheduling systems. Above all, real-world problems and realistic criteria for schedule optimization should be considered [5].

Using evolutionary computation techniques, we deal with a class of real-world problems with schedule cost related to resource management. Our previous application oriented studies include scheduling of operations in a production unit of a textile factory, where the objective was to ensure optimal energy consumption [7], and scheduling activities in ship repair in order to balance the work load for workers of various trades [8].

In this paper we describe production scheduling on a group of production lines of an automobile factory. The problem is non-typical in two respects. First, it requires process interruptions to be scheduled rather than processes themselves. Second, the optimality criterion is not based on a traditional schedule performance measure, such as overall processing time, but related to energy consumption. The objective is to schedule interruptions of the running processes in such a manner that energy consumption over the peak demand periods is minimized. In addition, the schedules are subject to time and resource constraints that have to be strictly satisfied.

Design of an evolutionary scheduling system for this problem and the initial practical results were presented in [9, 10], while here the focus is on an improved version of the system and its evaluation. The paper explains the scheduling problem and the schedule cost that is related to energy consumption, describes the employed scheduling system, provides the results of its evaluation on real problem instances, and discusses them in view of regular exploitation at the plant.

2 Production Scheduling Based on Energy Costs

Production systems relying on intense energy consumption, such as steel plants and other heavy industries, are faced with peak demand periods. These are the time periods over which their power demand exceeds a given limitation and the excess has to be paid at a higher rate. This measure is imposed by the energy supplier to minimize the total energy consumption over critical periods. There are several ways of reducing the peak power demand: activation of internal energy sources, interruption of energy-intensive processes, and appropriate production scheduling.

2.1 The Scheduling Task

The focus of energy consumption management in the considered factory is in the car-body production unit. The unit consists of six lines of hydraulic presses that perform cutting and shaping. A line in operation is regarded as an individual work process. Power demands of the processes vary from 20kW to 370kW. The unit operates according to a daily production plan that specifies which of the lines are in operation and what is their work time. Power demand of the unit equals to the sum of power demands of the running processes. Other energy consumers at the plant contribute to the so called background power demand, P_b. The total power demand of the plant, P, consists of the demand of the pressing unit and the background demand. To asses the energy costs, the total demand is related to the prescribed limitation $P_{\max}$, also called the target load. Figure 1 shows an example of daily profiles for background demand, total power demand and target load.

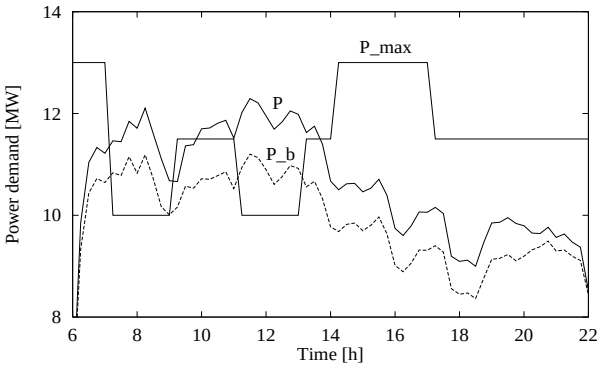


Figure 1: Background demand P_b, total power demand P, and target load $P_{\max}$ on the production lines

The efforts to reduce the target load excess are concentrated on line production in the pressing unit, since it is an intense energy consumer and also more suitable for scheduling than background processes. Two approaches are combined in the unit: process interrupting and scheduling. Process interruptions are either intended as breaks for the staff or can be spent to change machine tools and perform maintenance on the lines. The idea is to schedule these activities in such a way that the daily production plan is realized, while the contribution of the unit to the target load excess is minimized.

To balance between the conflicting requirements of plan fulfilment and reduction of the target load excess, the following constraints have been imposed on schedules:

- duration of process interruptions, T_0,
- minimum period of time between two interruptions of a process, T,
- maximum number of processes that can be interrupted simultaneously, M.

Taking into account these constraints, process interruptions have to be scheduled so as to minimize the target load excess contributed by the production lines.

2.2 Schedule Cost

The schedule cost to be minimized is formally defined as follows. Let $P_i(t)$, $i = 1, \ldots, N$, denote the power demands of the operating production lines in the pressing unit. The total power demand of the plant is

$$P(t) = \sum_{i=1}^{N} P_i(t) + P_b(t) \tag{1}$$

where $P_b(t)$ represents the background demand. Then the contribution of the considered processes to the target load excess at time t is

$$P_{\text{exc}}(t) = \begin{cases} \sum_{i=1}^{N} P_i(t); & P_b(t) \geq P_{\max}(t) \\ P(t) - P_{\max}(t); & P_b(t) < P_{\max}(t) \;\&\; P(t) > P_{\max}(t) \\ 0; & \text{otherwise} \end{cases} \tag{2}$$

and the energy consumption resulting from the target load excess equals to

$$W_{\text{exc}} = \int_t P_{\text{exc}}(t)\, dt. \tag{3}$$

W_{exc} represents the cost of interruption schedules which is to be minimized. It is to be noted, however, that power demands are in practice sampled using certain time interval Δt and integral (3) is approximated by

$$\sum_t P_{\text{exc}}(t)\, \Delta t. \tag{4}$$

3 The Scheduling System

The scheduling system generates daily schedules of process interruptions and calculates the expected reduction of the target load excess. It accepts the following input information:

- estimates of power demand profiles for the processes to be executed,
- an estimate of the background demand profile,
- the target load profile, and
- constraints to be considered in schedule construction.

The power demand estimates are based on data recorded over previous days and on production plan for the current day. As the production does not change rapidly, the estimates are rather accurate and make it possible to generate realistic production schedules.

The scheduling algorithm is designed to solve problem instances with arbitrary power demand profiles and can operate at various time discretizations. The core of the algorithm is a $(\mu + \lambda)$ evolution strategy [14]. It iteratively improves the schedules through the following sequence of steps:

Step 0: Generate an initial population of μ schedules by randomly assigning starting times to process interruptions.

Step 1: Generate λ descendants from μ parents by applying local transformations to schedules.

Step 2: Select μ best solutions out of $\mu + \lambda$ available, and make them parents for the next generation.

Step 3: If maximum number of generations is reached, exit, otherwise go to Step 1.

The best schedule found during this search process is returned as a suboptimal solution to the problem. Both, inserting the interruptions into a schedule at the initialization step and local schedule transformations are performed in such a way that constraints imposed on schedules remain satisfied. This is achieved by maintaining a direct representation of schedules within the algorithm and checking the constraints. Schedules are represented as two-dimensional arrays with the number of rows equal to the number of running processes, and columns to time intervals considered during scheduling. Each element of the array holds a value denoting the process status at the corresponding time interval. The status can be: *interrupted*, which means the process is interrupted, *interruption possible*, which means the process is running and it is possible to interrupt it, or *interruption not possible*, which means the process is running but cannot be interrupted due to the constraints.

Schedule transformations are carried out on random basis and include:

– inserting an interruption into a schedule,

– deleting an interruption from a schedule,

– shifting an interruption within a schedule.

Insertion of an interruption consists of finding a random time slot in the schedule with *interruption possible*, changing its status to *interrupted* and updating the status of the slots affected through constraint values T and M to *interruption not possible*. Deletion of an interruption includes random selection of an *interrupted* slot, changing its status to *interrupt possible*, and updating the status of the slots that are no more effected through constraint values T and M to *interruption possible*. Shifting of an interruption consists of its deletion at the current time slot and insertion at another time slot.

4 Evaluation and Results

4.1 Tests on Real Scheduling Problems

The scheduling system was initially tested on a set of problem instances based on real data recorded at the plant. The data were used as input to optimize daily schedules for the production lines. The constraints for schedule construction were set as follows. Duration of process interruptions, T_0, was 30 minutes. Each process had to run continuously for at least four hours between two interruptions ($T = 240$ min), and at most three process interruptions were permitted to take place simultaneously ($M = 3$). Time step used during search for assigning starting times to interruptions was 5 minutes.

The scheduling algorithm was run for 200 generations. The population size and the number of offspring generated in each generation were $\mu = \lambda = 20$. For each problem instance, the algorithm was executed 10 times, and both the best and average results were recorded. The optimized schedules of process interruptions were produced in the form shown in Table 1.

Table 1: An optimized interruption schedule for the production lines

Line number	Interruption times	
1	8:00–8:30	12:15–12:45
2	7:25–7:55	11:55–12:25
3	7:00–7:30	12:20–12:50
4	7:15–7:45	11:45–12:15
5		11:00–11:30
6		11:30–12:00

The evaluation confirmed that schedule optimization can substantially contribute to the decrease of energy costs in the production unit. Energy consumption resulting from the target load excess on the lines was reduced by at least 25% on workdays, but in most cases by about 30%. Table 2 shows the achieved reduction averaged over 10 runs of the optimization algorithm for each day in a two-week period. The reproducibility of the reduction in kWh obtained in 10 runs for each problem instance was within 2%.

4.2 On the Optimality of Schedules

Additional numerical experiments were carried out to check how close the optimized schedules are to the true optimal ones. For this purpose a selected scheduling problem representing a typical situation at the plant was used. All six production lines were required to operate and estimates of power demand profiles shown in Fig. 1 were used. If case of no process interruptions, the target load excess would amount to 3218.3 kWh. For this power demand situation, test problem instances of various complexity were defined. Their complexity was varied through constraint values T_0, T, and M.

Table 2: Average reduction of the target load excess obtained by the scheduling system

Day	Target load excess [kWh]	Reduction [kWh]	[%]
Mon	2616.5	1000.4	38.2
Tue	2569.6	970.5	37.8
Wed	3218.3	1012.6	31.5
Thu	2892.2	926.2	32.0
Fri	3055.1	931.6	30.5
Sat	655.0	413.0	63.0
Sun	0.0	0.0	0.0
Mon	2461.2	810.1	32.9
Tue	2117.7	636.6	30.1
Wed	2910.3	836.8	28.7
Thu	2752.8	803.3	29.2
Fri	2523.5	869.8	34.5
Sat	0.0	0.0	0.0
Sun	0.0	0.0	0.0

To denote a problem instance with particular constraint values, will use the notation (T_0, T, M), where times T_0 and T are given in minutes, and $M \in [1..N]$. The test set of problem instances consisted of $(30, 240, 3)$, $(30, 240, 1)$, $(30, 120, 3)$, $(30, 120, 1)$. Note that $(30, 240, 3)$ is the default setting of constraint values used at the plant, while additional settings resulting in more demanding problems were chosen to further check the performance of the developed scheduling system.

For the evaluation purposes, optimal schedules for the selected problem instances were produced by the Constraint Logic Programming approach. Constraint Logic Programming (CLP, [4, 12]) is a generalization of logic programming [13] where unification is replaced by a more general mechanism of constraint satisfaction over a specific computation domain, such as Boolean, finite or real. It is capable of finding optimal solutions to the problems of manageable size. We used the ECL^{i}PS^e CLP environment and its finite domain solver CLP($\mathcal{F}$). Unfortunately, the scheduling task introduced in Subsection 2.1 is too complex to be treated generally. However, particular problem instances can be handled individually by considering their specificities during problem solving.

Schedule costs found by this tool and by the evolutionary scheduling system are compared in Table 3. More clear picture of the evolutionary algorithm performance can be obtained from Table 4 which shows the deviation of the schedule improvement from the optimum gained with CLP.

These results are very informative for practical assessment of the evolutionary scheduling algorithm. While the initial tests on real problems showed that potential decrease of energy costs is at the expected level, we now have an absolute measure of the scheduling algorithm performance. It is particularly encouraging, that under constraint setting $(30, 240, 3)$, which is usually used at the plant, the result is very close to the optimum. For hypothetical problems with more complex spaces the gap to the optimum increases, but we believe that initial results given in this paper can still be improved.

Table 3: Optimal schedule costs in kWh found with CLP and suboptimal costs obtained with the evolutionary algorithm (EA)

Problem instance (T_0, T, M)	CLP	EA best	EA average
(30, 240, 3)	2209.1	2211.5	2213.1
(30, 240, 1)	2374.3	2385.1	2419.7
(30, 120, 3)	2185.8	2187.1	2187.4
(30, 120, 1)	2345.8	2365.8	2390.4

Table 4: Deviation of schedule cost improvement by the EA from the optimal improvement obtained with CLP

Problem instance (T_0, T, M)	EA best	EA average
(30, 240, 3)	0.2%	0.4%
(30, 240, 1)	1.3%	5.4%
(30, 120, 3)	0.1%	0.2%
(30, 120, 1)	2.3%	5.1%

Certainly, one may ask whether it is possible to apply the CLP system for regular scheduling at the plant. It turns out that its advantage of guaranteed optimal solutions comes at some other costs. Solving problems of this type with CLP is only efficient on individual basis, where additional constraints for schedules are derived from input data (e.g. feasible time intervals for process interruptions) and implemented to prune the search space. Further increase of problem complexity would sooner or later exceed the capabilities of the system. The CPU time spent to obtain optimal solutions with the CLP system depends very much on the problem instance, and ranges from a few minutes to several hours on a Pentium computer. On the other hand, the execution of the evolutionary algorithm on the same computer requires about half a minute for each problem instance, and only slightly increases with problem complexity.

5 Conclusion

An evolutionary algorithm was developed to schedule process interruptions on car-body production lines in an automobile factory where the objective is to decrease power demand over critical periods. In a comparative study the results of the evolutionary algorithm were assessed with regards to optimal results found by a CLP system. The comparison was beneficial in that it confirmed the evolutionary algorithm is capable of finding near-optimal results for a typical scheduling task appearing on the lines.

The approach was implemented as a process scheduling

module within a system for energy consumption management at the plant. It facilitates monitoring and control of energy consumption, while its primary role is to assist the process supervisor in preparing daily production schedules for the pressing unit.

In practical exploitation it has turned out the system is beneficial under certain amount of work load for the production lines. When the amount of orders is low, the lines are not heavily loaded and the resulting power demand does not exceed the target load. Hence there is no need for interruption scheduling and energy consumption optimization. On the other hand, the plant may get numerous orders and tight deadlines to accomplish them, and therefore deliberately decides not to interrupt the line processes as the additional energy costs are less than the penalties for not fulfilling the orders in time. Between these two extremes, there are however modes of operation where the system is regularly used and contributes to the decrease of production costs.

Acknowledgement

This work was supported by the Ministry of Education, Science and Sport of the Republic of Slovenia, and by the companies Revoz and INEA.

References

[1] T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.): *Handbook of Evolutionary Computing*. Institute of Physics Publishing, Bristol, and Oxford University Press, New York, 1997.

[2] J. Biethahn, V. Nissen (Eds.): *Evolutionary Algorithms in Management Applications*. Springer-Verlag, Berlin, 1995.

[3] R. Bruns: Scheduling. Chapter F1.5 in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computing*. Institute of Physics Publishing, Bristol, and Oxford University Press, New York, 1997.

[4] J. Cohen: Constraint logic programming languages. *Communications of the ACM*, 33 (7), 52–68, 1990.

[5] D. Corne, P. Ross: Practical issues and recent advances in job- and open-shop scheduling. In D. Dasgupta, Z. Michalewicz (Eds.), *Evolutionary Algorithms in Engineering Applications*, pp. 531–546. Springer-Verlag, Berlin, 1997.

[6] D. Dasgupta, Z. Michalewicz (Eds.): *Evolutionary Algorithms in Engineering Applications*. Springer-Verlag, Berlin, 1997.

[7] B. Filipič: Enhancing genetic search to schedule a production unit. In B. Neumann (Ed.), *Proceedings of the 10th European Conference on Artificial Intelligence ECAI '92*, pp. 603–607, Vienna, Austria, 1992. John Wiley, Chichester. Also published in J. Dorn, K. A. Froeschl (Eds.), *Scheduling of Production Processes*, pp. 61–69. Ellis Horwood, Chichester, 1993.

[8] B. Filipič: A genetic algorithm applied to resource management in production systems. In J. Biethahn, V. Nissen (Eds.), *Evolutionary Algorithms in Management Applications*, pp. 101–111. Springer-Verlag, Berlin, 1995.

[9] B. Filipič: A hybrid optimization algorithm for energy consumption management at a motor plant. In *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing EUFIT '97*, vol. 1, pp. 717–721, Aachen, Germany, 1997.

[10] B. Filipič, D. Zupanič: Near-optimal scheduling of line production with an evolutionary algorithm. In *Proceedings of the 1999 Congress on Evolutionary Computation CEC 99*, pp. 1124–1129, Washington, D.C. IEEE, Piscataway, 1999.

[11] K. S. Hindi, H. Yang, K. Fleszar: An Evolutionary Algorithm for Resource-Constrained Project Scheduling. *IEEE Transactions on Evolutionary Computation*, 6 (5), 512–518, 2002.

[12] J. Jaffar, J.-L. Lassez: Constraint logic programming. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, pp. 111–119, Munich, Germany, 1987.

[13] J. W. Lloyd: *Foundations of Logic Programming*. Springer-Verlag, Berlin, 1987.

[14] H.-P. Schwefel: *Evolution and Optimum Seeking*. John Wiley, New York, 1995.

Evolutionary Balancing of Healthy Meals

Barbara Koroušić Seljak
Computer Systems Department, Jožef Stefan Institute
Jamova 39, SI-1000 Ljubljana, Slovenia
Barbara.Korousic@ijs.si

Keywords: Optimization, multiconstrained fractional knapsack problem, genetic algorithms.

Received: July 7, 2004

In this paper we present an evolutionary algorithm for solving the nutrition problem of composing and balancing healthy meals. We treat this problem as a single-objective and multiconstrained fractional knapsack problem that is easy to formulate, yet, its decision problem is in the class of NP-complete problems. In other words, some heuristic algorithm is required to provide good problem solutions in reasonable (polynomial) computational time. We applied a genetic algorithm and modified its parameters to yield high-quality and reliable solutions (healthy and balanced meals) that respect multiple weakly-correlated dietary recommendations and guidelines and include as much seasonal functional foods as possible. Functional foods contain physiologically active compounds that provide health benefits beyond their nutrient contributions.

Povzetek: V članku predstavljamo evolucijski algoritem za reševanje problema optimalne sestave jedilnika.

1 Introduction

Nutrition is the process of nourishing, by which our body obtains nutrients and non-nutrients. These are chemical substances obtained from food and used in the body to provide energy, structural materials, and regulating agents to support growth, maintenance and repair of the body's tissues. Foods are composed of water and solids. Solid materials include carbohydrates, lipids, protein, vitamins, minerals and other compounds. Water, carbohydrates, lipids, protein, vitamins and some of the minerals found in foods are nutrients, while components of foods that contain alcohols, phytochemicals, pigments, aditives and others are non-nutrients. Some non-nutrients are beneficial (like flavonoids, isoflavones or lignans), some are neutral, and a few are harmful. Food choices influence our mental performance, emotional well-being and physical performance (health).

There exist many advices (dietary recommendations and guidelines) for proper nutrition based on objective scientific medical and nutrition research that may prevent and control nutritional deficiencies, infectious diseases and even chronic diseases [1]. They define the amounts of energy, nutrients and other dietary components that best support health. Although the intent of these advices may seem simple enough, they are the subject of much misunderstanding and controversy. Using a reliable nutrition software could help determine which advices should be applied and adjusted to meet our individual needs.

In this paper we introduce the mathematical aspect of such a nutrition software that is used as an optimization tool for composing and balancing healthy meals. A healthy meal provides sufficient energy and enough of all the nutrients and beneficial non-nutrients. Balancing a meal involves using enough, but not too much, of each type of food.

In Section 2 we provide a formulation of the problem of composing and balancing meals as a single-objective and multiconstrained fractional knapsack problem; in Section 3 we describe a genetic algorithm for the single-objective and multiconstrained knapsack problem; in Section 4 we evaluate the method; and in Section 5 we list conclusions and suggest possible future work.

2 Problem of balancing healthy meals

Finding an optimal composition of foods that could be served as a healthy and balanced meal is a complex problem for two reasons: there are many problem constraints (advices for proper nutrition) that are weakly-correlated, and the search space (the set of all possible combinations) is complex. The problem is even more difficult because the food quality may vary by season. Anyhow, solutions of the problem are trade-off solutions. For such solutions no improvement in any constraint is possible without violating at least one of other constraints.

We treat the problem of composing and balancing healthy meals as a single-objective and multiconstrained (multidimensional) fractional knapsack problem (MFKP) that is easy to formulate, yet it can be solved efficiently only by using some optimization techniques. It is also called the multi-knapsack problem. Many practical problems can be formulated as a multiconstrained knapsack problem, for example, the capital budgeting problem. Other applications of the problem include

allocating processors in a distributed computer system, project selection, and cutting stock problems. In our case, the MFKP is defined as follows.

Given food items of different values (qualities) and volumes (data about energy, nutrients, and non-nutrients), find the most valuable (healthy-and-balanced) composition of foods which fit in a knapsack (meal) of fixed volumes. Values are defined subjectively with respect to food functionality, seasonal availability and price. Knapsack volumes are defined by weakly-correlated dietary recommendations and guidelines, such as:

- recommended intakes of energy, nutrients and non-nutrients;
- adequate carbohydrates : protein : fat energy ratio;
- adequate ratio of essential fatty acids;
- recommended consumption of fruits and vegetables;
- restricted intake of fats and dietary cholesterol; etc.

2.1 Formal definition

We are given a knapsack of m capacities C_k for $k = 1, 2,..., m$, and n objects (food items). Each object has a value $v_i \in I^+$, $v_i > 0$, and a set of volumes $\omega_{i,k} \in R^+$, $\omega_{i,k} \geq 0$, one for each capacity. We would like to find a selection of objects $i \in I^+$, $i \geq 0$, such that $\sum_{i=1}^{n} \omega_{i,k} x_i \Theta C_k$ (Θ can be $\leq, =$ or $\geq$, $k = 1, 2,..., m$) and for which the total value, $\sum_{i=1}^{n} v_i x_i$, is maximized. The parameter $x_i = f_i P_i$ denotes the quantity of the selected object, where $f_i \in F$, F={0.25,0.5,0.75,1,1.5,2,3,…,10}, is a fraction of its portion size $P_i \in R^+$, $P_i \geq 0$.

The decision problem of the MFKP is NP-complete [2]. The only two exact algorithms that deliver optimum solutions to multiconstrained knapsack problems are based on the branch-and-bound [3] and the dynamic programming [4] approaches. On the other hand, heuristic methods for solving knapsack problems that have time complexity bounded by a polynomial in the size parameters of the problem have been known for many decades. A comprehensive review of the multiconstrained 0-1 knapsack problem and the associated heuristic algorithms is given by Chu and Beasley [5].

3 Genetic algorithm for the MFKP

We decided to compose and balance healthy meals in a heuristic way by using a genetic algorithm (GA) [6]. The theoretical foundations of this effective optimization technique were originally developed by Holland [7]. GAs are different from traditional optimization techniques as they simulate nature at a very abstract level to get solutions for a variety of demanding problems. They have been shown to be well suited for solving problems characterized by local minima. In recent years, a number of papers involving the use of GAs to solve multiconstrained knapsack problems have appeared.

The basic characteristic of GAs is that they search through an arbitrary search space both for exploration and exploitation purpose. The evolutionary process of biological organisms in nature is simulated by taking an initial population of individuals and applying genetic operators to the selected (normally highly-fit) individuals. Each individual in the population is encoded into a string (chromosome) that represents a possible (candidate) solution to a given problem. The fitness of an individual is evaluated with respect to a given objective function. Highly-fit individuals are reproduced by exchanging genetic information with other highly-fit individuals. This produces new offspring solutions that replace either less-fit individuals or the whole population. This procedure is repeated until a satisfactory solution is found.

Although the exploration of the search space is driven by random decisions, GAs are far from random search routines. The random decisions made in GAs can be modelled using Markov chain analysis. In this way, it can be shown that GAs will converge to globally optimum solutions [8].

3.1 Direct encoding

The first step in designing a GA is to encode the candidate solutions to the problem. We applied a real-valued coding of candidate solutions to the problem.

Because people consume several thousands of food items, we decided to separate n objects into G groups, where $G \leq n$ and G is few tens. Creating a composite meal, we select at most one item from each group. Hence, in our representation, a chromosome contains G' pairs (i_g, x_{i_g}), where i_g denotes the code of the selected object from a food group g and $x_{i_g} = f_{i_g} \cdot P_{i_g}$ its quantity (Figure 1), f_{i_g} being the fraction size of i_g and P_{i_g} its default portion size. A null value $i_g = 0$ implies that no item is selected for a group g. Each of the G food groups may be omitted or repeated within a chromosome. Normally, food groups are selected with respect to the food guide pyramide [9], which is an outline of what to eat each day based on the dietary recommendations and guidelines. It is not a rigid prescription, but a general guide that let us choose healthful meals that are right for us.

The number of possible solutions is approximately $(\frac{n}{G})^{G'|F|}$, where n denotes the number of food items, G the number of food groups, G' the number of food groups captured within a chromosome and F the set of

fraction sizes. Normally, n is in the range of few hundreds to few ten thousands, G is few tens, and $|F|$ is approximately ten.

g	1		…	G'	
(ig, x_{ig})	1060 milk	0.5·244		19034 popcorn	5·8

Figure 1: Description of a chromosome (candidate solution).

In our implementation, the GA starts either with a random population of candidate solutions or a population of solutions (healthy meals) known from experience. The population size is few tens and remains constant over all populations.

3.2 Fitness evaluation

Each solution of the candidate population is evaluated using the following fitness (objective) function:

$$f(\vec{i}) = \sum_{g=1}^{G} v_{i_g} f_{i_g}, \quad (1)$$

where higher value of v_{i_g} means better food quality. The aim of the GA is to maximize this fitness function.

3.3 Infeasible solutions

A chromosome might represent an infeasible solution. This is a solution for which at least one knapsack constraint is violated. There are several ways of dealing with infeasible solutions in GAs [10], i.e.,

- by penalizing infeasible solutions,
- by incorporating a repair operator that transforms an infeasible solution into a feasible one, or
- by using an order-based encoding instead of a direct one.

In our case, we applied the first two approaches. First, we incorporated a penalty term into the fitness function (1) to penalize the fitness of infeasible solutions, without distorting the fitness landscape:

$$f(\vec{i}) = \sum_{g=1}^{G} v_{i_g} f_{i_g} - p(\vec{i}). \quad (2)$$

The penalty term $p(\vec{i})$ is defined in a static way by adding a metric based on a number of constraints violated:

$$p(\vec{i}) = \sum_{k=1}^{m} \mathrm{X}_k \cdot \delta_{\vec{i},k}, \quad (3)$$

where $\delta_{\vec{i},k} = \begin{cases} 1, \text{if } \vec{i} \text{ violates a constraint } k \\ 0, \text{if } \vec{i} \text{ satisfies a constraint } k \end{cases}$

and X_k is the weight of a constraint k.

The next step was to transform a percentage of infeasible solutions into feasible ones by using a greedy repair operator. This operator consists of the following phases:

1. Rank the problem constraints violated by the infeasible solution in the decreasing order of their weights X_k, $k = \{1, ..., m\}$;
2. For each violated problem constraint, starting with the most critical one, sort food items in the chromosome in
   a. the increasing order of their values, and
   b. the decreasing or increasing order of their weights for the exceeded or deceeded constraints, respectively;
3. For each food item in the sorted chromosome, starting with the weakest one, find an alternative item with better nutrient profile for a given constraint. The alternative is searched either in the item's neighborhood or by random in a given food group, depending on the probability of repair. Namely, food items are ordered in each food group so that similar foods are close to each other.
4. Repeat the local-improvement operation of Step 3 at a given repair rate, or until a given problem constraint is satisfied.
5. Go to Step 2.

To prevent premature convergence of the GA, some infeasible solutions were left unrepaired in the population. Allowing a small percentage of infeasible solutions to join the population is longed-for because optimum solutions frequently lie on the boundaries of feasible regions [11].

3.4 Parent selection

To create new candidate solutions, two chromosomes have to be selected from the current population as parents. In our implementation of the GA, best-ranked (highly-fit) feasible solutions are more likely to be selected for reproduction because we apply the elitism strategy, where a number of least-fit members of the current candidate population are interchanged with an equal number of the best-ranked chromosomes. This strategy increases the performance of the GA, because it prevents losing the best-found feasible solutions.

The parent selection is realized via the tournament approach. This is based upon an idea of forming two pools of candidate solutions, each consisting of the same number of chromosomes. Two solutions with the best fitness, each taken from one of the tournament pools, are chosen to be parents. Using a larger size of the tournament pools has the effect of increasing selection pressure on the more-fit solutions. The problem of getting stuck in a local-optimum solution can happen. To avoid this problem, we adopt the standard (binary) tournament selection technique and realize the elitism through the interchange ratio of least-fit to best-ranked solutions. This ratio is in the order of 4 or 6 down to 1 chromosome per population, depending on the population size.

3.5 Crossover and mutation

In crossover, a selected pair of chromosomes are mated to produce an offspring that replaces the least-fit solution in a given population if its fitness ranks above. This steady-state approach may perform better than generational GAs because it better retains feasible solutions found in the populations and may have higher selection pressure [12]. We apply a uniform crossover operator to produce a solution that preserves the genetic material from both parents. Each element of the offspring's chromosome (a pair of the food item's code and its quantity) is created by copying the corresponding element from one of the parents, chosen according to a binary random number generator $[0,1]$. In our implementation of the GA, using a two-point crossover operator can also perform crossover. In this approach, copying the corresponding elements from one parent, and all the others by copying the corresponding elements from other parent, creates elements of the offspring's chromosome between two points, selected by using a crossover probability.

Once the offspring has been generated through the selection and the crossover, mutation is performed on few randomly selected elements of the chromosome. Each element is mutated in one of the following ways chosen at random,

- by local-improvement operation of the greedy repair algorithm (Step 3, Section 3.3), i.e., the code of a given food item and its quantity are replaced with a close alternative item from the same food group and its recommended quantity (with the default fraction size of 1), respectively;
- by multiplying the size of the food item's portion by a randomly selected fraction factor from F={0.25, 0.5, 0.75, 1, 1.5, 2, 3, …, 10}.

The fixed rate of mutation is set to be a small value (in the order of 1 or 2 elements per chromosome).

3.6 Termination criteria

The GA terminates its operations when the system is assumed to be in a stable state, i.e., an optimum feasible solution has been found (a wanted-solution approach), or a certain number of populations have been generated and evaluated (a time-out approach).

4 Evaluation of the GA for the MFKP

In order to evaluate the proposed evolutionary method for composing and balancing healthy meals, we optimized a set of randomly generated meals using the GA. We used the USDA nutrition database, Release 16 [13], which is the major source of food composition data in the United States and is available free of charge. It includes nutrient profiles for more than 6500 food items that are grouped into 23 food groups. The items are ordered so that similar foods are grouped together. Each nutrient profile contains more than 30 values, such as macronutrients, elements, vitamins, etc. We used these data as weights. Values of food items were defined subjectivelly, considering their functionality and availability in a given season. Seasonal foods from the list of functional foods (e.g., apples, broccoli and red wine in automn, avocado in winter, blueberries in summer, etc.) were assigned the highest values. We considered the following representative dietary recommendations and guidelines for:

- the energy value of a meal;
- the carbohydrates : protein : fat energy ratio;
- the intake of dietary cholesterol per meal;
- the intake of vitamin E per meal;
- the linoleic (omega-6) fatty acid : alpha-linoleic (omega-3) fatty acid ratio.

These constraints are equality constraints, except the one on dietary cholesterol that is an inequality (less-than-equal) constraint.

We developed software that implements the GA for composing and balancing healthy meals using the Borland Delphi programming tool. It runs under the Microsoft Windows operating systems on a Pentium PC. The USDA database is used in a Microsoft Access format.

4.1 Experiments and results

After several runs of the program, the most advantageous settings for the GA were defined (Table1), and a set of good solutions to the problem of composing and balancing healthy meals was collected.

Table1: Settings for the GA parameters.

Parameter	*Value*
Population size	20
Repair probability	0.5
Repair rate	2
Elitism ratio	$0.2 \div 0.05$
Tournament pool size	2
Crossover probability	0.7
Mutation rate	0.05
Termination criterion	1000 evaluations

It has proved that a repair operator has to be used in addition to the static penalty function. Otherwise, the search gets stuck in a local minimum without finding a good feasible solution before the termination happens. We estimated that approximately each third candidate solution was infeasible and required repairing. Although the worst-case time complexity of the repair algorithm is $\sum_{g=1}^{G'} O(n_g \cdot m)$, where n_g denotes the number of food items in a food group g and m the number of constraints, in practice $O(m)$ steps were needed to find a close feasible solution. However, the most difficult task in greedy repair was to derive the proper constraint weights X_k, $k = \{1, ..., m\}$ because some of the constraints are weakly-correlated. We defined the highest weight to the

constraint on dietary cholesterol and the lowest to the constraint on the energy intake. In between were the second and the fourth constraints, i.e., the ratios of macronutrients and essential fatty acids, respectively.

In Tables 2 and 3, a good (feasible) daily-meal solution generated from an initial population of random candidate solutions that satisfies the selected problem constraints and its nutritional profile are presented. In fact, these initial candidate solutions were all infeasible. In the solution, the quantities of foods were selected as multipliers of the portion sizes. Larger portions of more than 100 grams were multiplied by a fraction factor from $\{0.25, 0.5, 0.75, 1, 1.5, 2\}$ and smaller portions of few grams by a factor from $\{1, ..., 10\}$, respectively. In Table 2, selected foods are specified with a short description instead of a code.

Table 2: A good daily-meal solution.

Grams	*Food item*
	Breakfast
72	ORANGE DRK,BRKFST TYPE,W/ JUC & PULP, FRZ CONC
38	CEREALS,MALTEX,DRY
	Lunch
141	WEIGHT WATCHERS ON-THE-GO CHICK,BROCLI&CHDR POCKT SNDWCH,FRZ
	Supper
496	SOUP,TOMATO,LO NA,W/H2O
80	FAST FOODS,POTATO,MASHED
31	TURKEY,YOUNG HEN,SKN ONLY,CKD,RSTD
1	GINGER,GROUND
123	BEETS,CND,REG PK,SOL&LIQUIDS
28	CAKE,CHERRY FUDGE W/CHOC FRSTNG
31	ENSURE PLUS,LIQ NUTR
	Snack
150	BANANAS,RAW
21,5	SOYBEANS,MATURE SEEDS,RSTD,SALTED
	Dinner
195	RICE,BROWN,LONG-GRAIN,CKD
85	MACKEREL,ATLANTIC,RAW
42	ALMOND PASTE

Table 3: Nutritional profile of the daily-meal solution.

.

RESULTS	*Recommended*	*Achieved*
Value	$\geq (3 \cdot 0{,}5 \cdot 23)$	✓
Energy (Kcal)	1800 ± 100	1875,7
Protein (% of energy)	$10 \div 15$	14,5
Total lipids (% of energy)	$20 \div 30$	29,8
Carbohydrates (% of energy)	$50 \div 60$	55,7
Dietary cholesterol (mg)	≤ 300	120,4
Vitamin E (mg)	15 ± 2	14,1
Saturated FAs (% of energy)	≤ 10	7,5
ω-6 FA + ω-3 FA (g)	$(11 + 5{,}5) \pm 2$	17,4

In Figure 1 performance of the proposed balancing method, based on measurements of the number of times the candidate solutions were evaluated to come within a certain fraction of the optimum, are presented. From the direct comparison, it can be seen that the repair technique for dealing with infeasible solutions performed better than the penalty one. Namely, repair of infeasible candidate solutions assured faster generation of best-ranked feasible solutions.

The program takes 8 seconds to generate an initial population of 20 candidate solutions and 300 milliseconds to calculate the quality, the fitness and the penalty of a given solution. Repair is more time consuming as each local-improvement operation (Steps 3 and 4, Section 3.3) takes 8 seconds at most, replacing one third of objects in a given infeasible solution at the selected repair rate. Considering one third of infeasible candidate solutions in each population, the total repair time is approximately 1 minute per population.

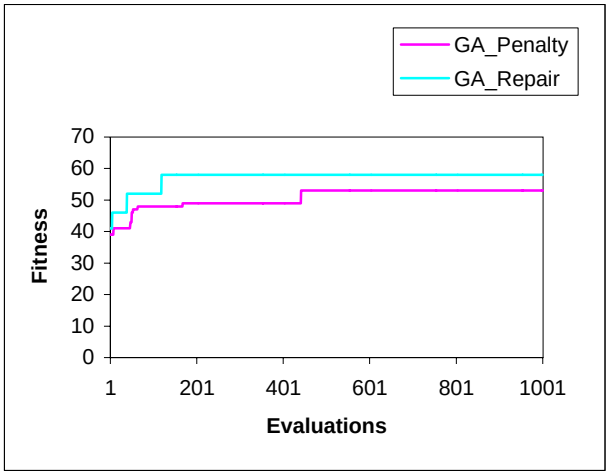


Figure 1: Performance of the meal composing and balancing GA method.

5 Conclusions

In the paper we have presented a heuristic method for composing and balancing healthy meals that considers several constraints and the quality of foods, with the tendancy of including as much as seasonal functional foods as possible. The method is based on a steady-state genetic algorithm that is a particular evolutionary algorithm. It uses direct (real-valued) coding of problem solutions, the elitism and the tournament approach for selection of parents, uniform and two-point crossover, and local-improvement mutation. All the GA parameters are fixed for all populations and candidate solutions. Infeasible solutions are penalized by a static function based on the number of constraints violated. Some infeasible solutions are further repaired so that the least-quality food items are replaced with more appropriate items or their quantities are modified by a multiple of the predefined portion size. Repair is performed in a greedy way. We collected the experimental results by running the GA from an initial population of random candidate solutions. From the results it has been shown that the problem of infeasible solutions could be solved more efficiently by applying a repair operator than merely by

penalizing. We also proved that few infeasible solutions are welcome to the population.

Including some real meals known to be healthy from experience into the initial population, the method could perform better. We could also experiment with other (self-adaptive) penalty functions to reduce the cost of repair, and with an order-based encoding of candidate solutions to the problem. Last but not least, the USDA database need to be replaced with the Slovene national nutritional database.

Acknowledgement

The work presented in this paper has been supported by the Slovenian Ministry of Health (project Development of a public domain server application for food analysis and optimization that considers modern dietary recommendations).

References

[1] POKORN, D., *Dietetika*, DZS, 1999 (in Slovene).

[2] GAREY, M.R. and JOHNSON, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[3] GAVISH, B. and PIRKUL, H., "Efficient Algorithms for Solving Multiconstrained Zero-One Knapsack problems to Optimality", *Mathematical Programming* 31, pp. 78-105, 1985.

[4] SOYSTER, A.L., LEV, B., and SIVKA, W., "Zero-One Programming with Many Variables and Few Constraints", *European Journal of Operational Research* 2, pp. 195-201, 1978.

[5] CHU, P.C. and BEASLEY, J.E., "A Genetic Algorithm for the Multidimensional Knapsack Problem", Journal of Heuristics, 4: 63-86, 1998.

[6] GOLDBERG, D.E., *Genetic Algorithms in Search, optimization, and Machine Learning*. Addison-Wesley, 1989.

[7] HOLLAND, J.H., *Adaptation in Neutral and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.

[8] KARR, C.L., YAKUSHIN, I., and NICOLOSI, K., "Solving inverse initial-value, boundary-value problems via genetic algorithm", *Eng. Applicat. Artif. Intell., 13(6):625-633*, Dec. 2000.

[9] WHITNEY, E.N., CATALDO, C.B., and ROLFES, S.R., *Understanding Normal and Clinical Nutrition*. Wadsworth, Thomson Learning, 2002.

[10] MICHALEWICZ, Z., *Genetic Algorithms+Data Structures=Evolution Programs*. Springer Verlag, 1996.

[11] SIEDLECKI, W. and SKLANSKY, J., "Constrained genetic optimization via dynamic reward-penalty balancing and its use in pattern recognition". In *Proc. of the Third International Conference on Genetic Algorithms*, pp.141-150, 1989.

[12] CHAFEKAR, D., XUAN, J., and RASHEED, K., "Constrained Multi-Objective Optimization Using Steady State Genetic Algorithms". In *Erick Cantú-Paz et al (Editors): Genetic and Evolutionary Computation---GECCO 2003, Proceedings, Part I*, pp. 813--824, Springer. Lecture Notes in Computer Science, Vol. 2723, July 2003.

[13] http://www.nal.usda.gov/fnic/foodcomp

eGovernance: Relation Theory of the Impact Factors

Jaro Berce
RS Government Office for EU Affairs,
Subiceva 11, 1000 Ljubljana, Slovenia,
Tel: +386 1 4782421, fax: +386 1 4782431
e-mail: jaro.berce@gov.si

Keywords: iGovernment, eGovernment, knowledge management, learning organization, information communication technology, eGovernance, public administration

Received: June 26, 2004

A research work presented in the paper establishes that the overall goal in public sector is better governance - helping to narrow the knowledge divide and its consequences between societies and also within each society so to stipulate eDemocracy. The learning process narrow divides between those societies who can accumulate, manage, when/how to share, and when to use knowledge and those who face obstacle and challenges at one ore more stages in the sequence from acquiring knowledge to use. A strong linking of all factors (social, economic, technology) and their proper transform order are the winning solutions. The answer lies with recognition of peoples dimension: knowledge (people's and organization's), sharing and networking knowledge and information, learning, and organizational issues encouraged with economic dimension (rewarding, budget, and financing) and driven by new technology solutions. Therefore, Information and Communication Technology is a general "driving" force that makes Knowledge Management, Learning Organization, and eGovernance more important activity than in the past.

Povzetek: članek opisuje pomen informacijskih in komunikacijskih tehnologij za upravljanje v informacijski družbi.

1 Introduction

The strategic impact of new technology, new organizational principles, and new knowledge paradigm is bringing influence to our everyday life. Main characteristic of usage of new technology is introducing intelligent environment where information creation and communication and as well intercommunication can be provided through different technologies. This new digital venue also affects a public sector and within it a public administration. Studies of public sector organizational governance and related building blocks have been published as strategies and research papers, Internet sites, books, since development of new technologies. Presented paper is adding new dimensions in the field of digital venue and impact factors that influence good eGovernance.

The information communication technology (ICT) is an infrastructure that supports an electronic venue of an organizational social environment. In the study it is represented by: iGovernment - converting existing processes and paper objects to digital form and eGovernment - converting literal services to virtual services. Here I would like to stress the difference in notion of eGovernment. It is most of the time understood as a whole digitalization of a government; this conception brings only one dimension in the government practice - digital service orientation, skipping two important organizational dimensions: knowledge management (KM) and learning organization (LO). Therefore, next researched issue was knowledge and knowledge management. Knowledge is the most important "asset" of today's organizations. Further on, a non-stable environment is shaping the enveloping organizational culture - a learning organization that is suitable to apply occurring changes. To govern the whole system of a modern public organization all three previously mentioned items (ICT, KM and LO) are mandatory for good governance and not only eGovernment solutions. In such a manner can be stipulated better democracy that is in favour of the whole society.

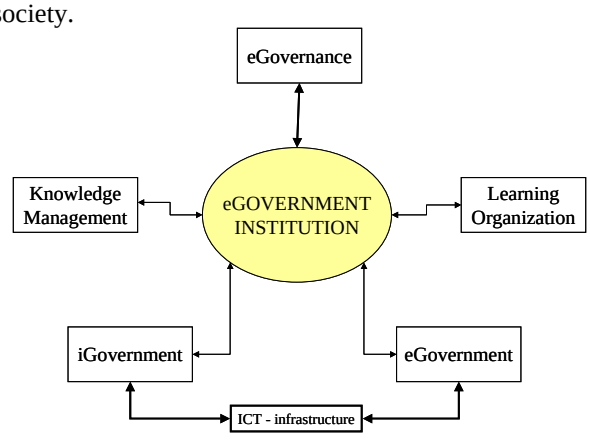


Fig. 1: Research model: "building blocks" and their relations

The research work focuses on the study of a public administration. A practice that is best understood as a system, with multiple external and internal factors, which shape the behaviour of participants, activities of programs and ultimate outcomes [5]. The terminus "public administration" was used in the study for public entities and their relationships with each other and with the larger world through eGovernance system narrowed to: how public sector organizations are organized and managed to accomplish their responsibility.

1.1 Introduction of Terms

As a definition, **Information and Communication Technology** (ICT) can be defined as "electronic means of capturing, processing, storing, and communicating information" and is based on digital information held as 1s and 0s, and comprise computer hardware, software, and networks [1]. Two breakthrough inventions formed the Information Society's foundation: computers and telecommunications. Computers deprive humans of their monopoly on "intelligence" and perform the predictable, routine intellectual tasks. Telecommunications, in turn, ensures common access and information (1s and 0s) spread to all computers connected to network and/or Internet. **Knowledge management** (KM) typically means the systematic management and use of the knowledge in an organization; more abstrusely: "the leveraging of collective wisdom to increase responsiveness and innovation" [Delphi Consulting Group[1]]. The **Learning Organization** (LO) [are] organizations where people continually expand their capacity to create the results they truly desire, where new and expansive patterns of thinking are nurtured, where collective aspiration is set free, and where people are continually learning to see the whole together [6]. **Government** is the system or form by which a community or other political unit is governed. Governments produce huge volumes of information and an increasing amount of it is available through electronic venues, the Internet, and other electronic means. **Governance** understood in presented research work is the exercise of economic, political, and administrative authority to manage the nation's affairs at all levels. It comprises all the mechanisms, processes, and institutions through which the citizens and groups articulate their interests, exercise their legal rights and obligations and mediate their differences. The concept of **eGovernance** relates to the preparation of government as it reacts to information, technology and communication trends on its traditional governance role in society [4].

1.2 Reflection On Research Frame

Governance is the main process done within government. For this process, several factors are important. Learning organization is suitable solution where new confronts are implemented and brought by an information technology. As well new eDemocracy paradigm is stimulating novel way in interactions between public sector and citizens. Knowledge and knowledge management are obviously of fundamental ingredients in supporting correct decisions and through that governance. Therefore, how one can establish a knowledge sharing culture (meaning accumulate, store, access, share, use tacit knowledge in the heads of individuals on the organizational level, as well as "corporate knowledge") within public institutions and as well with citizens and organizations? What impact does this knowledge-sharing culture have on governance? What is the role of information communication technology in this respect? Are civil servants suitable equipped (knowledge, infrastructure, etc.) to take care of eGovernance? Are the organizational incentives important in workers behaviour? Is learning process essential to govern? Conventionally, ICTs have been used within governments for automating processes, replacing clerical labour with its digital equivalent (writing documents, written instructions etc.). eGovernance should be therefore seen as a compound made up of different blocks [7]:

- ICT that support communication between government and civil society,
- Managing institution(s), governmental information and knowledge, and
- The societal and economic effects of electronic democracy.

In 1999, the European Commission proposed a new *e*Europe (within pursuit for a communication on the role of eGovernment for Europe's future[2]) initiative to speed up Europe's entry into the digital age and to ensue coherence in the pace of progress of its Member States. The objective of the *e*Europe initiative was an ambitious one: to bring every citizen, school and business on line and to exploit the potential of the new economy for growth, employment and inclusion. The Commission presented the initiative to the Lisbon European Council in March 2000.

1.3 Scope

The primary aim of the research was to study the objectives and relation impacts of the ICT on the performance of public organizations. A further aim was to analyse the correlation between knowledge, learning organization and improvements in organizational performance (eGovernance) within the public institutions. A public administration influences through its efficiency and effectiveness the whole value-chain (taxation: rise or lowering) on state economic scale. While the efficiency and effectiveness analysis, [or estimating "return on investment" (ROI)] is important it was not studied within the scope of this research. The same apply for the value-added effects.

[1] See: http://www.delphigroup.com/coverage/knowledge_management.htm, Feb. 2004

[2] See: http://europa.eu.int/information_society/eeurope/2005/all_about/egovernment/index_en.htm, Oct. 2003

V1_1 organization

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1,00 Ministry	14	9,8	9,8	9,8
	2,00 constituent office	10	7,0	7,0	16,8
	3,00 local government	53	37,1	37,1	53,8
	4,00 munincipality	46	32,2	32,2	86,0
	5,00 government office	20	14,0	14,0	100,0
	Total	143	100,0	100,0	

Table 1: *Demographics of research population*

For the purpose of the study a survey was drawn. It was used to gather needed data and prepare ground for a theory development that supports interpretation of strategic influences. Based on information gathered from respondents (e.g. institutional information, education level, information infrastructure, budgeting process, etc.) the survey also sought to clarity influence of information communication technology on digital information (iGovernment) and digital services (eGovernment). Analysis more broadly represents also an accumulation of information on the respondents' level of agreement with statements concerning knowledge management and information communication technology. The examination of background information is important because it was hoped it would be possible to focus on the characteristics and factors that enhance the importance of knowledge, ICT infrastructure, and budgeting as well as intangible resources to organizational performance. In the last part of the survey, respondents were asked to pick out those factors and elements of staff motivation that are of special concern to them. Educational level of employees and their training habits were also included in survey questionnaire. Clarifications of stated problems are anticipated through gathered information about respondent.

A total of 288 different public organizations participated in the survey. The breakdown of replies received was as follows:

- From ministries: 100%.
- Local government authorities: 94.6%.
- Government offices: 83.3%.
- In total of 22.9% of answers came from municipalities.

1.4 Theory Background

Through explanation of stated research frame, a pattern development was examined. Research work then comprised of merging predefined "organizational segments" in blocks to investigate theirs relations and enable study of impacts. To aim or navigate the correct procedures that attain this type of research work a **block relationship theory** [2] (BReT) was deployed. The origin of developed BReT model is in an empirically tested pattern developed through study case that make possible to investigate important relations within organization and offers some explanatory outputs:

- Information communication technology (ICT) (Infrastructure, eService, Internet, Policy)
- Organizational issue (Organizational initiative, Educational Level, Permanent Education (Training) Rewarding System)
- Knowledge Management Issues (eCapture of Staff Competency, Information and Knowledge Network, Knowledge Responsibility)
- Budgeting.

It is designed to estimate and test through Structural Equation Model, multivariable linear relationships among *latent* - explanatory variables that operate basic theoretic concepts and *manifest* - dependent variables measured by latent ones. A computer tool - Lisrel "Structural Equation Model of multivariable linear relationships" was used in calculating pre-defined relations and to "weight" their correlations.

2 Theory: Reflections and Findings

The emergence of the new *information communication technologies*, organizational issues (*learning organization*, etc), and consciousness of *knowledge management and networking* importance are driving factors that propel organizations through changing real and virtual environment. In addition, governments worldwide, on all levels - from state to local, are seeking to harness the potential offered by these new technologies and organizational concerns to create new dimensions that describe good eGovernance. A new paradigm is emerging: Knowledge networks and communities transform the digital convergence into people-centered development that beat the digital divide by narrowing the digital gap (also) through eGovernance. The primary role of eGovernance supported by learning, knowledge management and networking is to ensure that the stress is placed on decision-making (doing the right things) and not on increased efficiency (doing things right). Good eGovernance integrates information communication technology that enables governance to carry out the tasks with better control across time and space. An important aspect of eGovernance is the

relationship between government (state and local) and other society's stakeholders (citizens, business, civil society organizations).

Within BReT the manifested variables were constructed to study relation and behavior, of gathered data. The iGovernment variable shows how organizations register working processes: from paper to digital expert systems. Variable shows that digital compilation is well expanded. Confirmation of researched results of well-applied ICT infrastructure within Slovenian organization [3] can be shown also with high statistical reliability of this variable. The eGovernment variable shows how public organizations internally use digital services provided by technology and theirs participation with joint eAdministration project(s). An even distribution was found, meaning that most Slovenian organizations use digital services in a very dedicated form. They are using mostly fundamental (e-mail, e-calendar, etc.) digital services. Only one quarter of organizations, exploit bundled digital services at their work. eGovernment variable in the field of e-services as: approvals of car licenses, tax payments, or other life situations, and legislation, budget spending, etc. confirms that institutions are not using them for external user.

The Knowledge Management variable is a viewpoint type of scale. It is measured with Likert scaling method and represents average of opinions of top management. Important factors (rewarding system, decentralization, knowledge officer, etc.) that accelerate KM and are not well enforced, therefore top management of Slovenian administration organizations has somehow loosen perception on how to exercise in practice KM by their employees. Survey results shows that they have their self-evaluated perception that is higher than can be proven by other survey data.

The Learning Organization variable represents an organizational assist and track to its employees' educational necessity. It is a composition of data (in percentage) representing professional educational aim. Research proved that gaining higher professional knowledge is more applied within Slovenian organizations than other types [languages, ICT (learning text processors, spread sheets etc.), and other] of study programs. Comparing with results shown from other correlations derivate from study that lower educated staffs attain less training than those with higher education can indicate that gaining better or new expert knowledge is limited to those that already have their profession. Others do not yet realize or do not have possibilities to upgrade their education.

The final studied manifest variable is eGovernance. It is composed with different indicators. Three types of data groups are used to calculate this variable. First consist of dichotomous-like [YES/NO] data and shows if organization has its own Internet connection. Next group is composed with organizational issues concerning how institution acts outward(s) [measured through replays to incoming e-mail(s)]. Through this, principles to understand new tools and new digital venue philosophy that are important for good digital governance can be tested. Last group illustrates a level of service digitalization used by institution. It is based on OECD four-stage measuring principle and weighted. Due to complexity of calculation of eGovernance manifest variable, an explanation of it can be only shown through a BReT model with which measurement of non-direct measurable conceptions (eGovernance) can be measured with direct measurable indicators (policy, budget, etc.).

The model has proven the researched hypothesis. Although, it has some limitations:

- A relative adequacy presentation of an actual phenomenon
- A model and a real presentation are not identical
- A "working" model means a relative suitability of a real observable fact(s).

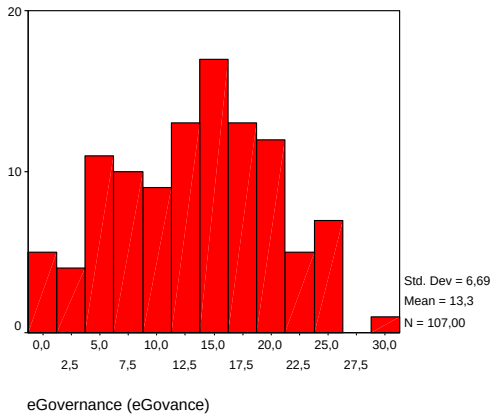


Fig. 2: *eGovernance - variable*

A BReT is a LISREL type "covariance structure" model and has 150 connections - linear structural equations. An impact between all variables is present within a model. Impacts between variables are shown on arrows that connect variables. Higher values represent impact that is more significant. Only for the eGovernment manifest variable an impact from the iGovernment manifest variable was build by the LISREL showing that an importance of digitalization of documents is influencing eServices. Arrows with values that are oriented to manifest (green box) variables are showing unexplained (external) influences. A problem concerning 143 items that were observed in 150 equations influences a reliability of the model. Nevertheless, a null hypothesis value of the "root-mean-square error of approximation" (RMSEA = 0.040) index being less than 0.080 shows that

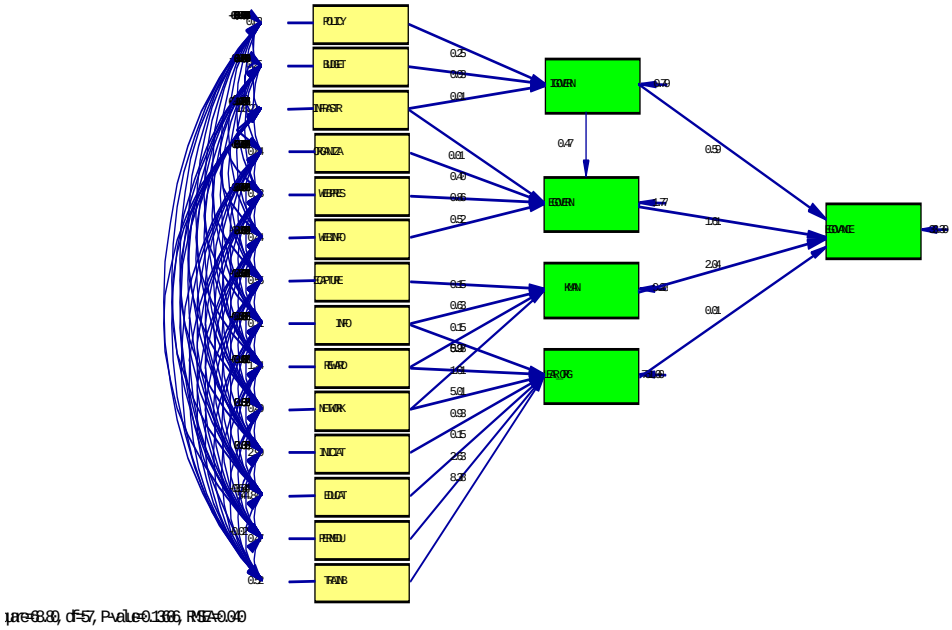


Fig. 3: *Block relation theory – BReT: variables model representation*

model stands well. The "significance" (P-value = 0.13606) greater than 0.05 is typically considered "close fit". The "degrees of freedom" (df = 57) greater than zero allows power to calculate a model. Therefore, a statistical value for a Gauss distribution (χ^2 = 68.80) significantly **validates an acceptation of the model.**

3 Conclusion

From research work a strong linking of all mentioned factors (social, economic, technology) and their proper change orders are the winning solutions for better eGovernance. The solution lies with recognition of *peoples dimension*: knowledge (people's and organization's), sharing knowledge and information, learning, and organizational issues back-up then with *economic dimension*: rewarding, budget, and financing and driven by *new technology solutions*. If only one is missing the effectiveness and efficiency of good eGovernance cannot be achieved. Therefore, ICT is a general "driver" force that makes KM, LO, and eGovernance a more important activity than in the past.

As suggested in the analysis and the research results [8], the way organizations reward workers for their behaviour is generally a good indication of where organizations stand in terms of promoting this behaviour. As is to be expected from the results of the overall achievements of KM practices, the organizational structure of central government bodies seems not to have entirely accompanied or supported cultural changes in within their staff. A strategy of knowledge management is sterile if it does not also provide for a continuous learning process.

The lack of a reward structure for knowledge sharing and the apparent focus of organizations on technology while (sometimes) underestimating the importance of the human factor, as well, relative managerial resistance to the implementation of Knowledge Management and Learning Organization strategies (middle managers have the most to lose from more horizontal knowledge-sharing) and the absence of new governance mechanisms accompanying the changing responsibilities, are classic limitations on the implementation of good eGovernance policies. These topics, all of which relate to change strategies and the obstacles to change, are suggested for further research.

Overall, the challenge to good eGovernance, as faced by the public administration sector, is how to implement a broad change strategy that goes well beyond the introduction of an ICT-enabled digital venue in which to locate many of the structures and processes of governance. It includes changes to the culture of service provision, as well as to the cultures of knowledge management and learning organization behaviour.

References

[1] Ayala Jose J.: *Information Society & Development: Review; Latin America - Final Report*; Contract Cadre: STG-CEAL; (May 2000); http://europa.eu.int/comm/external_relations/info_soc_dev/doc/latin_america_report_en.pdf ; (Sept. 2000)

[2] Berce, Jaro: *eGovernance: A block relation theory*; VITEL – 2003 (Ljubljana); pp. 91-96 [COBISS.SI-ID 2185682]

[3] Berce, Jaro: *The realm of knowledge management at the public sector*; V: Rajkovic, Vladislav (ur.),

Urbancic, Tanja (ur.), Bernik, Mojca (ur.), Rozman, Ivan (ur.), Hericko, Marjan (ur.), Vilfan, Boštjan (ur.), Bavec, Cene (ur.), Bucar, Maja (ur.). Zbornik B 6. mednarodne multi-konference Informacijska druzba IS 2003, October 13. - 17., 2003; Ljubljana; Institut "Jozef Stefan"; 2003; p. 346-350

[4] Clift Steven: *E-Governance to E-Democracy: Progress in Australia and New Zealand toward Information-Age Democracy*; March 2002; http://www.Publicus.Net; (Apr., 2003)

[5] Johnson, Gail: *When World's collide: Public Administration within the University*; 23rd National Conference on Teaching Public Administration; Old Dominion University, College of Business and Public Administration, Norfolk, Va; January 29-31, 2000 http://teachpol.tcnj.edu/conference_papers/_manuscripts/Johnson_G._00.PDF; (February 2004)

[6] Senge, M. Peter: *The Fifth Discipline. The art and practice of the learning organization*, London: Random House. 424 + viii pages, 1990;

[7] The International Teledemocracy Centre; http://itc.napier.ac.uk/ITC_Home/ITC/Research.asp

[8] Berce, Jaro: Doctoral Dissertation - *The influence of Information Society Technologies on Evolution in the Public Administration*, 2004

An Agent for Categorizing and Geolocating News Articles

Žiga Mahkovec
Faculty of Computer and Information Science
Trzaska cesta 25
SI-1001 Ljubljana
Slovenia
ziga.mahkovec@klika.si

Keywords: Text categorization, support vector machine, RCV1, geolocation, GIS, SVG

Received: June 1, 2004

We present a software agent for categorizing and geolocating news articles. The articles are retrieved from different on-line news sources, such as Google News, Reuters and BBC News. They are parsed, categorized based on crime threat, geographically located and rendered in an SVG widget.
The agent is implemented in Java, using the Scalable Vector Graphics markup language to render the user interface. Text categorization is performed using the Support Vector Machine (SVM) method, with test data from the Reuters RCV1 corpus. The GEONet Names Server and Digital Chart of the World databases are used to geolocate the news articles.

Povzetek: Članek opisuje inteligentnega agenta za lokalizacijo novic.

1 Introduction

The U.S. Department of Homeland Security maintains a special Homeland Security Advisory System, with a 5-stage threat level ranging from green (”Low level of terrorist attacks”) to red (”Severe risk of terrorist attacks”). There are few similar systems available for other countries.

Using on-line news sources, text categorization and geographical localization we were able to develop an agent capable of presenting such information for the entire world. News articles are categorized as ”good” and ”bad” — specifically, to news related to terror, war and violence and other news. They are also parsed to retrieve the exact location of the article’s subject. Using various worldwide news sources and applying categorization and location enables the visualization of the threat level for the entire world.

Text categorization is performed using Support Vector Machines (SVM [3]). This method has proved to be successful in solving several text categorization problems, since it avoids over-fitting when presented with a large number of attributes. For learning and classification we used the SVM-Light implementation [4].

The training set consisted of the Reuters Corpus Volume 1 (RCV1), as modified by David Lewis et al. [5]. The data set consists of over 800,000 news articles, published by Reuters between the years 1996 and 1997.

The *GEOnet Names Server* [6] and *Geographic Names Information System* [7] databases were used to locate the news articles. The articles were then rendered within an SVG widget containing a satellite image of the world and vector political boundaries.

2 News sources

The following news sources were used:

- Google News: `http://news.google.com`
- Reuters: `http://reuters.com`
- BBC News: `http://news.bbc.co.uk`

2.1 Google News

Google News is an aggregator of more than 4,500 news sources from all over the world. The service provides automatic news grouping, pulling together related headlines. The number of related news articles is a good indicator of the importance of a news article.

Google News does not provide RSS feeds [8]. Content retrieval is therefore based on HTML parsing, using the `java.util.regex` package for regular expressions. The parsing engine requires periodical testing, since the format of the Google News articles may change in the future.

Geolocation is performed by parsing the headlines of all related news articles. Most often, the headlines will include the location header, e.g. ”BASRA, Iraq –”. By searching for the most frequent geographical name in the headlines, the article can be accurately located.

Although there are several localized editions available, only the U.S. edition was used for news retrieval.

2.2 Reuters

Reuters, being the largest news agency, publishes some 11.000 stories daily. Its website offers news in 13 differ-

ent categories. RSS feeds are provided, greatly simplifying content retrieval, since the XML format is structured and easy to parse. Figure 1 shows an example of an RSS feed item.

```
<title>
  Blast Near U.S. Compound in Iraq's Basra Kills 2
</title>
<guid isPermaLink="false">6210014</guid>
<link>
  http://www.reuters.com/newsArticle.jhtml?storyID=6210014
</link>
<pubDate>Sat, 11 Sep 2004 13:55:26 GMT</pubDate>
<description>
  BASRA, Iraq (Reuters) — A car bomb exploded near the U.S.
  embassy office in the southern Iraqi city of Basra on
  Saturday, killing two people and wounding three, but no
  Americans were injured, officials and witnesses said.
</description>
</item>
```

Figure 1: An example of a Reuters RSS feed

The Reuters article headers consistently include the location header (e.g. ”BASRA, Iraq”)

2.3 BBC News

BBC News also provides RSS feeds. The articles are already partly geographically located (the geographical categories include Africa, Americas, Asia-Pacific, Europe, Middle East, South Asia and UK).

3 Text categorization

The parsed news articles are represented as a Java class containing the following attributes:

- title
- headline
- full text
- news category
- source URL

The full text is used to categorize the news articles.

3.1 Test collection

The Reuters Corpus Volume 1 (RCV1) is an archive of over 800,000 categorized news articles. Lewis et al. [5] used it to produce the RCV1-v2 corpus, containing some corrections. They also provide vectors for training with SVM classifiers.

The documents are available in XML format. A sample document is shown in figure 2.

The RCV1 documents are coded into three category sets: industry, topics and regions. To separate the ”bad” news from the rest, we used specific topic categories:

- GCRIM: crime, law enforcement
- GVIO: war, civil war

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<newsitem itemid="2440" id="root" date="1996-08-20"
  xml:lang="en">
<DOCNO> RC-2440 </DOCNO>
<title>
  SINGAPORE: Philippines wary of terrorist threats to
  APEC meet.
</title>
<headline>
  Philippines wary of terrorist threats to
  APEC meet.
</headline>
<dateline>SINGAPORE 1996-08-20</dateline>
<text>
<p>The Philippines' top military officer said on Tuesday
Manila was exchanging intelligence information with the
United States and other countries on potential terrorist
threats to the APEC summit later this year.</p>
...
</text>
<copyright>(c) Reuters Limited 1996</copyright>
<metadata>
<codes class="bip:countries:1.0">
  <code code="PHLNS"/>
  <code code="SINGP"/>
  <code code="USA"/>
</codes>
<codes class="bip:topics:1.0">
  <code code="GCAT"/>
  <code code="GVIO"/>
</codes>
<dc element="dc.date.created" value="1996-08-20"/>
<dc element="dc.publisher" value="Reuters Holdings Plc"/>
</metadata>
</newsitem>
```

Figure 2: An example of an RCV1 document

3.2 SVM categorization

The *LYRL2004 split* of the RCV1-v2 corpus contains 23,149 documents. 2,087 of these were labeled as ”bad” using the above criterion. This split was then used to train the SVM-Light classifier.

The news articles were preprocessed using the same technique as the one described in [5]. Stop words were removed; stemming was performed using the Porter stemmer. The $TF \times idf$ weights of the terms in the SVM vectors were computed as follows:

$$w_d(t) = (1 + \log_e n(t,d)) \times \log_e(|\mathcal{D}|/n(t)),$$

where $n(t)$ is the number of documents containing the term t; $n(t,d)$ is the number of occurrences of term t in document d, $|\mathcal{D}|$ is the number of documents used in computing the idf weights.

The feature vectors were also cosine normalized:

$$w_d^{'}(t) = \frac{w_d(t)}{\sqrt{\sum_u w_d(u) \times w_d(u)}}$$

The agent would then run the SVN-Light classifier for each new news article. The articles categorized as ”bad” are specifically marked on the map in the user interface, thus identifying the dangerous regions of the world.

4 Geolocating

The agent tries to accurately locate the venue of each news article. The latitude and longitude of the location are also retrieved, enabling visualization within the map of the world.

Most news sources use a standard header, consisting of the proper location, e.g. "BASRA, Iraq (Reuters)". Reuters is very consistent in producing these headers. Google News aggregates several news sources, thus displaying different header formats. However, by using the most frequent header of a set of related news, the location can be accurately defined.

Two databases were used when finding geographical names and their coordinates: the GEOnet Names Server (GNS) [6] and the Geographic Names Information System (GNIS) [7]. The former contains more than 4 million geographical features for the entire world; the latter contains 2 million features for US only.

The databases consist of several attributes for each of the geographical features:

- full name (including conventional, native and variant names)
- region
- latitude and longitude (in degrees)
- populated place classification (a graduated numerical scale denoting the relative importance of a populated place)
- feature name

The vast amount of geographical data was first filtered; only larger populated areas were retained, reducing the list to 27,290 features for the entire world.

When only partial location information is available for a document (e.g. city only), the most populated feature from the GIS database is used.

5 User interface

The categorized and located news articles are finally rendered in an SVG (Scalable Vector Graphics [2]) widget. The widget can be displayed in a web browser (using Adobe SVG Viewer) or in a standalone application, such as Apache Batik.

The SVG widget consists of:

- A satellite image of the world in 8096×4096 resolution (enabling three levels of zooming).
- Vector political boundaries of 203 countries.
- News pop-ups, shown when hovered; the pop-ups contain an image, title and the headline; by clicking it, a new browser window is opened, following the URL of the article.
- An ECMAScript library enabling client interaction: zoom-in and zoom-out, panning, news pop-ups, article linking, etc.

The result is an SVG widget displaying the map of the world (figure 3). News articles are marked as circles. The circle sizes denote news importance. The "bad" news is marked as red.

6 Conclusion

As expected, the result SVG widget marked the current crisis areas: the war in Iraq, the terrorist attack in Jakarta and violence in the US. Increased news location resolution would enable an even finer outlook of the world threat level. However, many news sources only cite the capitals or even news agency locations instead of the exact venue. This leads to news aggregation and a distorted threat location view.

The SVM classifier in coordination with the RCV1 corpus were successful in categorizing the news articles. They were especially fitting for the Reuters news source.

The SVG markup language proved to be suitable for geographical applications. The mixture of raster and vector graphics provides for a fast and appealing user interface. The `Newsloc` widget is modular and can be used for other GIS-related applications as well.

References

[1] W. Brenner, H. Wittig, and R. Zarnekow. *Intelligent Software Agents: Foundations and Applications*. Springer-Verlag, 1998.

[2] J. Ferraiolo, F. Jun, and D. Jackson. *Scalable Vector Graphics (SVG) 1.1 Specification*. W3C, 2003. `http://www.w3.org/TR/SVG`.

[3] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the European Conference on Machine Learning*, 1998.

[4] T. Joachims. Making large-scale SVM learning practical. *Advances in Kernel Methods - Support Vector Learning*, 1999.

[5] D. Lewis, Y. Yang, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[6] National Geospatial-Intelligence Agency (NGA). *GEOnet Names Server (GNS)*, 2004. `http://earth-info.nga.mil/gns/html`.

[7] U.S. Geological Survey. *Geographic Names Information System (GNIS)*, 2004. `http://geonames.usgs.gov/gnishome.html`.

[8] D. Winer. *RSS 2.0 Specification*, 2002. `http://blogs.law.harvard.edu/tech/rss`.

Figure 3: The `Newsloc` SVG widget

Visualization of News Articles

Marko Grobelnik and Dunja Mladenić
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{marko.grobelnik, dunja.mladenic}@ijs.si

Keywords: text visualization, context visualization, document clustering, large collection of news stories

Received: May 10, 2004

This paper presents a system for visualization of large amounts of new stories. In the first phase, the new stories are preprocessed for the purpose of name-entity extraction. Next, a graph of relationships between the extracted name entities is created, where each name entity represents one vertex in the graph and two name entities are connected if they appear in the same document. The graph of entities is presented as a local neighborhood enriched with additional contextual information in the form of characteristic keywords and related name entities connected to the entity in the focus. Operations for browsing a graph are implemented to be efficient enabling quick capturing of large amounts of information present in the original text.

Povzetek: članek opisuje postopek za vizualizacijo novic.

1 Introduction

Text visualization is an area having the main goal to present textual contents of one or many documents in a visual form. The intention of producing visualization of the textual contents is mainly to create graphical form of the content summary on different levels of abstraction.

In general, we can say that ideas used in text visualization algorithms come primarily from data analysis research areas (such as statistics, machine learning, data mining) [1, 2, 3, 4] where data visualization play important role as a key technique for showing the data and results of analytic methods. Textual data is in this respect just another type of data with its specific properties which need to be taken into account when visualizing it. Main characteristics relevant for text visualization are [5]:

- High data dimensionality when using typical bag-of-words representation, where each word and each phrase represents one dimension in the data space.
- High redundancy, meaning that many dimensions can be easily merged into one dimension without loosing much information. This is caused by the two properties of words, namely synonymy (different surface word forms having the same meaning – e.g. singer, vocalist) and hyponymy (one word denotes a subclass of an another – e.g. breakfast, is a subclass of a meal)
- Ambiguity between words in the cases where the same surface form of the word has different meanings (homonomy – e.g. the word 'bank' can mean 'river bank' or 'financial institution') or in the cases where the name form has related meaning (polysemy – e.g. 'bank' can mean 'blood bank' or 'financial institution')
- Frequency of words (and phrases) follows power distribution, meaning that we deal with small number of very frequent words and high number of infrequent words. Having this in mind, we need to use appropriate weighting schemas (e.g., most popular being TFIDF) to normalize importance of the words to be able to work with the standard data analytic techniques.

Furthermore, when talking about text visualization we also need to be aware of the type of text we are dealing with. Namely, different document types have different characteristics which need to be considered when designing an efficient text visualization mechanism. Some examples of such different types of textual data are: Web documents (being typically short, having linkage structure and additional formatting information), e-mails and news-group postings (short documents with specific internal structure, appearing in content threads and using specific language), customer reports, chat rooms discussions, literature, legal documents, technical text, news stories etc.

In this paper we are dealing with news stories. Specifically, we have designed and developed a system for preprocessing and visualizing large amounts of documents coming from a news wire. In general, news stories are special type of text having most often the following properties:

- short documents,
- written by professionals,
- low number of language mistakes,
- having good rhetorical structure,
- rich information about people, companies, places, etc.,
- a single news document containing pieces of larger stories usually spanning over several documents.

Our approach takes into account the above properties giving a special emphasis on the last two items namely, named objects (such as people, companies, and places) and the context they are appearing in.
In the following sections we present related work, sample news articles corpus, design and architecture of the system, name entity extraction, keyword extraction, browsing and visualization user interface and discussion at the end.

2 Related work

Wider area of the work presented in this paper is data visualization [3] and in particular text visualization [6]. This work also fits in the recent developments of semantic web in particular visualization of ontologies and other knowledge structures [1].

In this paper we are dealing with visualization and browsing of news stories which require special treatment. In the literature there are not many published works on this specific subtopic. Most prominent is the overview publication from MITRE team [7] giving good overview over the approaches for visualization of different document types, including news stories. Their goals are similar to the work presented here, but the actual approach is quite different. Their publication appeared also at [8] together with some other interesting approaches for document visualization.

Another approach for visualizing trends in news documents is the system ThemeRiver [9] developed at Pacific Northwest National Laboratory together with many other interesting approaches for information text visualization [10]. ThemeRiver in particular is specialized for analyzing and visualizing trends in news stories over time, enabling efficient detection of trends in the vocabulary used in the texts. Among others, we would also like to mention our previous work on visualization of large text corpora [11].

3 Sample news corpus

The functionality of our approach is presented here on a corpus of news articles from "ACM Technology News" service at http://www.acm.org/technews/archives.html. The corpus includes general news from the most areas of Information Technology (from December 1999 on). It includes over 11.000 article summaries of the length 200-400 words. Figure 1 shows a typical article summary from the corpus which is used in the subsequent procedure.

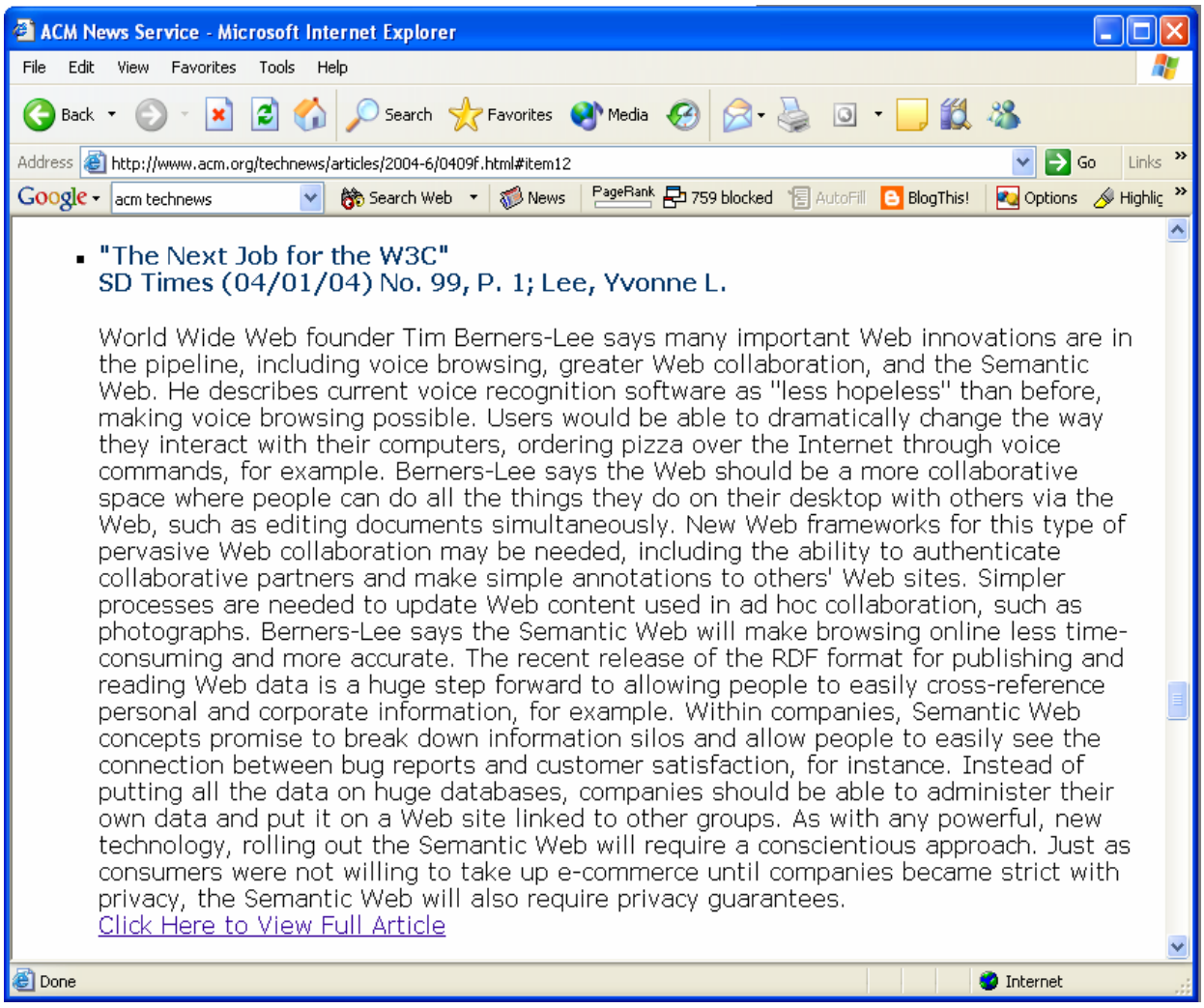


Figure 1. Example of a news article summary from ACM-TechNew

4 Design and architecture of the system

The main goal, when designing the system called "Contexter", was to help expert and semi-expert users (such as analysts, journalists, social scientists, experienced web surfers) to get an efficient and quick understanding of large corpus of general news stories providing different levels of abstraction. This is to be achieved by several means:

- by showing relationships between entities appearing within documents,
- by calculating and showing contexts within which the entities appear either individually or in combination with other entities,
- by using several types of visualization simultaneously,
- by efficient and responsive graphical user interface enabling easy moving from abstract to detailed information.

One of the fundamental design assumptions is that most of the relevant information is centered around the entities mentioned within the documents. In our context, entities can be names of people, names of companies and other institutions, geographical names and places, product names, etc. An additional property of entities is that they serve as connectors between different documents forming longer threads of stories which are not explicitly noted with typical news corpora (usually such information is not present in meta-data of news articles). Based on these observations, our basic representation of documents within the news corpus is three-fold: (1) plain text as originally provided, (2) bag-of-words representation of the text, (3) representation by a set of name-entities.

1. **Plain news text as written by the authors**.

This representation is used exclusively for showing the document content to the user, when the user comes to the point that s/he explicitly requests the full textual information. This representation offers lowest level of content abstraction.

2. **Bag-of-words representation using some kind of weighting schema** (in our case TFIDF).

In this case we still include most of the words appearing in the original text – we just delete the stop-words (non-informative functional words), perform stemming (unifying different surface forms for the same words), pre-calculate phrases (frequent and significant consecutive sequences of several words), and the most important, ignore the order of the words (for the purpose of more efficient computation). The goal of this representation is to efficiently calculate contexts in the form of keyword lists to allow for a higher abstraction of the contents compared to the plain text.

3. **Set of name-entities appearing within the article**.

In our case we use variant of relatively standard name-entity extraction algorithm based on word capitalization (primary candidates for the name-entities are the words starting with capital letter) with additional mechanism for name consolidation (detecting that e.g. 'Bill Clinton'=='President Clinton'=="Clinton'). This representation in our case offers the highest abstraction level for an individual document. Because of its structured nature (e.g. names are consolidated on the level of the whole news corpus) it serves as a connecting level between different documents.

On the input to the system we get a set of documents representing news articles. We have no special assumptions on the form, structure and meta-data within the documents – main element is textual part of the documents which is further processed. Next, the documents are preprocessed in two different ways. First, the text is cleaned and the bag-of-words representation is created, and next, the name-entities are extracted. All the documents are stored in the database in three different representations (as already described: plain text, bag-of-words and name-entities). The database is used by the client software using efficient graphical user interface described in the following sections. Figure 2 shows the architecture of the system.

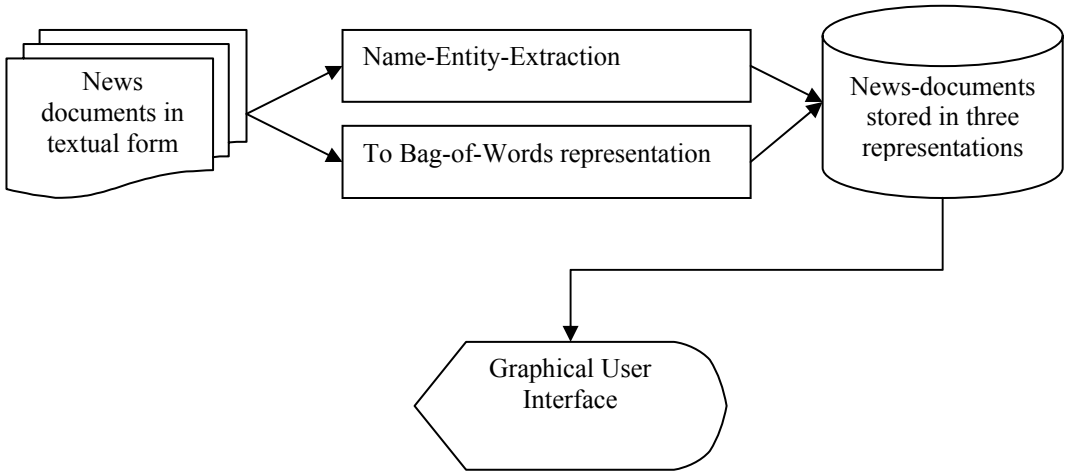


Figure 2. Architecture of the "Contexter" system

4.1 Named entity extraction

Information extraction and specifically name-entity-extraction [4] are one of the most popular areas of text mining. The main function is to convert parts of unstructured textual data into structured form which enables to use standard data analytic methods available in data mining and statistical packages (e.g. SAS and SPSS use this kind of approach). There are three main approaches when extracting useful pieces of information from text: (1) manual extraction rules, (2) automatically generated rules with machine learning methods, and (3) hybrid methods combining the two approaches. In everyday practice the approach with manual rules seems to be the most effective and frequently used. While machine learning methods give good results on datasets with lack of domain knowledge, the automatically generated rules usually need human corrections and additions to be practically useful. In general, for controlled corpora, initial investment needed to get good results with manually modified or even created rules seems to be the most price-performance effective.

In our case, name entity extraction algorithm is based on one of the most typical heuristic approaches – on word capitalization. This approach usually gives good results on high quality texts and introduces low overhead in terms of computational efficiency and additional tuning of parameters. Furthermore, it gives good results for most of the western languages without any special tuning (except for German which uses capital letters for all the nouns). Main characteristic of the method is that it provides very good recall (almost all of the real name-entities are proclaimed as name-entities), but slightly lower precision (some of the proclaimed name-entities are not name-entities) which is in practical setting enough – errors and exceptions are handled separately by the list of exceptions. Empirical evaluation of the method (based on 100 randomly selected news articles) showed precision value of 73% and recall value of 96% – recall and precision are standard Information Retrieval evaluation measures measuring the 'truth' and the 'whole truth' of the result set.

In addition to the name-entity extraction, we also use name consolidation mechanism which tries to unify different surface forms into one name-entity (e.g. 'Bill Clinton'=='President Clinton'=="Clinton'). For this purpose we use heuristic approach based on the phrase similarity.

4.2 Bag-of-Words representation and Keyword Extraction

Classical representation of documents in Information Retrieval is so called the bag-of-words (or word-vector) representation [2, 4, 5]. It enables efficient execution of several fundamental operations on the transformed text documents. The idea of bag-of-words representation is to represent each document as a vector of numeric variables, where each variable represents one word (or phrase) from the dictionary (union of all words from all the documents in the corpus). If a particular words appear within a document, then its vector includes non-zero value for the word-variable (usually number of appearances of the word within the document), otherwise, the value is zero. Since most of the values within the single document vector are zero, this calls for more efficient representation of the vector – typically vectors are represented with so called "sparse vector representation" which is an ordered set of pairs *(WordId, Weight)*, where *WordId* denotes word and *Weight* non-zero frequency of the word within the document (usually called term-frequency).

An important issue when dealing with bag-of-words representation is how to represent the word weights. Using plain term-frequency is usually not enough, because the power-distribution of the words (small number of very frequent words and high number of infrequent words) damages performance of most of the analytic methods. Therefore, we use one of the improved heuristic weighting schemas which correct the influence of the word distributions. The most popular weighting schema is TFIDF which calculates a weight for each word within each document using the following formula:

$$tfidf(w) = tf.\log(\frac{N}{df(w)})$$

In the above formula *tf* stands for the term-frequency (the number of word appearances within a document), *df* stands for the document frequency (the number of documents in which the words appears), and *N* is the number of all documents within the corpus.

Intuitively, we can say that words with higher TFIDF weight are more important. This intuition is also used for keyword extraction from one or more documents. When extracting keywords from a set of selected documents, we take their sparse vector representations (having TFIDF weights), we sum the vectors, and sort the words according to the TFIDF weight. The keywords are the words with the highest weight in the sorted list. This method is not perfect for selecting the best keywords (again, recall measure is usually higher then precision), but it gives reasonable results, is computationally very efficient and its results are easy interpretable. This method could be understood also as calculating an average document from a set of documents – this average document is also referred to as a centroid vector in the context of clustering (e.g. K-Means algorithm). This method of calculating most representative keywords from a set of documents is related to other eigenvector based methods (such as SVD, PCA, etc.) which are also used to calculate vectors of keywords but are in general computationally much more expensive and in general don't provide significantly better results. Figure 3 show an example of such a centroid vector for the documents from ACM TechNews corpus which mention the phrase "Semantic Web".

SEMANTIC (0.548)
SEMANTIC_WEB (0.524)
WEB (0.261)
BERNERS (0.119)
BERNERS_LEE (0.119)
ONTOLOGIES (0.100)
SEARCH (0.099)
LEE (0.099)
W3C (0.094)
RDF (0.089)
WORLD_WIDE_WEB (0.082)
METADATA (0.082)
WORLD_WIDE (0.081)
WIDE_WEB (0.081)
OWL (0.067)
WIDE_WEB_CONSORTIUM (0.065)
WEB_CONSORTIUM (0.065)
LANGUAGES (0.063)

Figure 3. Top 18 keywords with their TFIDF weight for the documents from ACM TechNews that contain phrase "Semantic Web".

5 Visul interface

In this section we present the client part of the "Contexter" system offering graphical user interface to the pre-calculated name-entities and bag-of-words representations of the news documents corpus which are stored together with the original textual representation within the database.

The core part of the system is the main graphical user interface form, which primarily offers two functionalities:

1. Browsing through the network of connected name-entities (two name-entities are connected if they appear in at least one common document).
2. Visualizing a context of a name-entity appearance within the corpus. The context of a name-entity is shown in three different ways:
   - by a set of keywords usually collocated with the selected name-entity,
   - by a set of other name-entities usually collocated with the selected name-entity,
   - by a set of keywords collocated with the simultaneous appearance of the selected and most frequent other name-entities.

Usage of "Contexter" consists from the following steps:

1. Preprocessing of the document corpus which generates name-entity and bag-of-word representations which are saved together with the original textual representation within a database. This step is preformed only once per database change. Since all the algorithms used in the preprocessing phase are computationally efficient, this step takes approx. 15 seconds for the whole ACM TechNews corpus (11.000 articles) on the 2.4GHz PC. We also experimented with other larger corpora (non English languages) and the experiments showed the system scales linearly according to the size of the data (which is expected according to the design of the system).
2. The user runs the client (see Figure 4) with the graphical user interface. First, the user connects to the database that contains the three document representations (see Section 4). This loads a part of the data into the system (list of all name-entities and cashed part of the bag-of-words sparse vectors).
3. As the user selects a name-entity in the left most window (eg., "Marc_Andreessen" in Figure 4), the system instantly shows the corresponding content in other three "context windows". First to the right is the window (1) with the graphical representation of the local context of the network around the selected name entity, (2) next, window to the right shows the context in the form of characteristic keywords from the documents where the selected name entity appears, and (3) the right most window shows the context in the form of the most frequent other name-entities collocated with the selected name-entity.

In the next steps, the user can select other name-entities (either from the complete list on the left, from the graphical interface in the middle or from the right most context list) which instantly adapts the screen according to the new selection. With additional local menu functions the user can view the actual context of the documents where the selected name entities appear.

6 Discussion

In the paper we presented design, architecture and implementation of the system "Contexter" used for analytical browsing of news articles. In the first stage documents from the corpus are preprocessed and transformed into two alternative representations – each document gets in addition to its original textual representation also name-entity and bag-of-words representations. As we are dealing with large amounts of text, for both transformations we decided to use simple and computationally efficient procedures which give satisfactory results in terms of quality. Quality could have been slightly increased with the selection of some other methods, but on the cost of computational efficiency which would further decrease usability of the interface.

There are several potential additions which are interesting for the future development of the system. In particular, with a more detailed analysis of the text in the preprocessing stage using some natural-language-processing tools, we would be able to identify finer grained contexts in which an individual name-entity is appearing; furthermore it would be possible to detect more explicit relationships between the name entities. Next, some more text visualization and text summarization techniques can be applied to extend levels of abstraction when observing the content. With an improved name-entity recognition and consolidation

(disambiguation), the usability of the system would increase especially in the cases where the cost of the preprocessing phase (in terms of time and human resources) is not very important.

Finally, the whole system would benefit a lot from a wider Human-Computer-Interaction study which would evaluate current system and suggest corrections to the user interface design and to the needs for various user profiles. In the current stage we designed system mainly for research journalists from some of the Slovenian daily newspapers which contributed suggestions through descriptions of their needs and what they perform in their everyday routine.

Acknowledgement

This work was partially supported by Slovenian Ministry of Education, Science and Sport and SEKT 6FP IP project on semantically enabled knowledge technologies.

References

[1] Geroimenko, V., Chen, C. (ed): *Visualizing the Semantic Web*. Springer Verlag, (2003).

[2] Manning, C., Schütze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press (1999).

[3] Fayyad, U., Grinstein, G., Wierse, A.: *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann (2001).

[4] Chakrabarti, S.: *Mining the Web: Analysis of Hypertext and Semi Structured Data*. Morgan Kaufman (2002).

[5] Jurafsky, D., Martin. J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall (2000).

[6] Chen, C.: Visualization of Knowledge Structures, In *Handbook of Software Engineering and Knowledge Engineering*, World Scientific Publishing (2002).

[7] Chase, P., D'Amore, R., Gershon, N., Holland, R., Hyland, R., Mani, I., Maybury, M., Merlino, A., Rayson J.:Semantic Visualization. *ACL-COLING Workshop on Content Visualization and Intermedia Representation*.

[8] Content Visualization and Intermedia Representations (CVIR'98), http://acl.ldc.upenn.edu/W/W98/

[9] Havre, S., Hetzler, E. , Whitney, P., Nowell, L.: ThemeRiver: Visualizing Thematic Changes in Large Document Collections. *IEEE Transactions on Visualization and Comp. Graphics*, V8, No.1, 2002.

[10] Pacific Northwest National Laboratory, Information Visualization, http://www.pnl.gov/infoviz/

[11] Grobelnik, M., Mladenic, D.: Efficient visualization of large text corpora. *7thTELRI, Info. in Corpora* (2002).

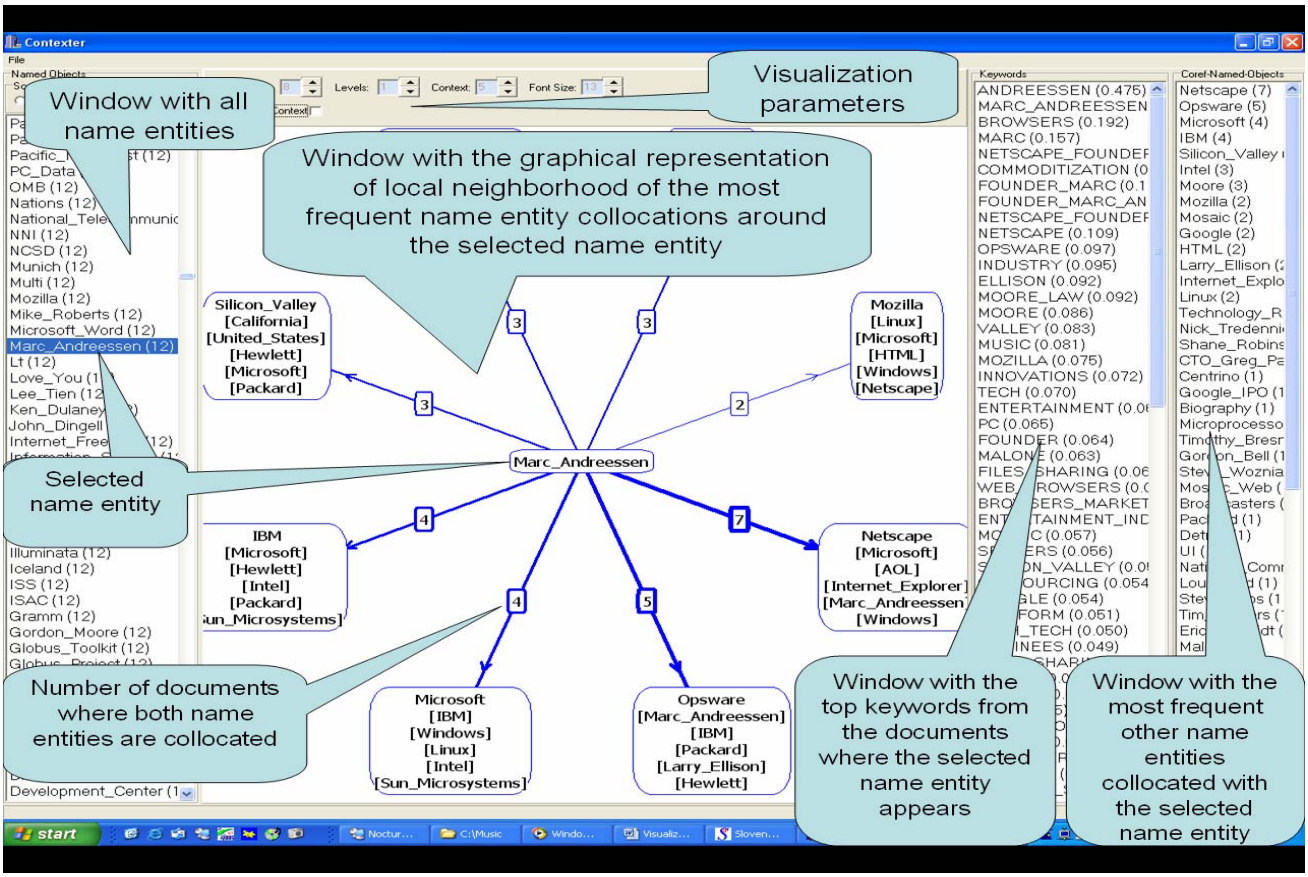


Figure 5. Graphical interface of "Contexter" for browsing/visualizing the name-entity network.

Shortest-Path Semantic Distance Measure in WordNet v2.0

Jure Ferlež and Matjaž Gams
Jožef Stefan Institute, Department of Intelligent Systems, Jamova 39, 1000 Ljubljana, Slovenia
Jure.Ferlez@ijs.si

Keywords: semantic relatedness, semantic distance, WordNet

Received: July 14, 2004

This paper analyses a measure of semantic relatedness between two words. Our measure is based on shortest path between synsets in WordNet v2.0. It uses all available links in WordNet v2.0 and is implemented by a bidirectional breadth-first algorithm. Experimental evaluation and comparison with a benchmark set of human similarity judgments demonstrates that the simple measure applied on WordNet v2.0 performs better than the more complicated approaches mostly combining an IS-A taxonomy with the notion of shared information content extracted from corpora. One explanation is that our pretty basic method efficiently exploits all the available links in WordNet v2.0, while other measures, although more complicated and advanced, do not make good use of the new derivational links added to WordNet in the latest version 2.0.

Povzetek: članek opisuje iskanje najkrajše pomenske razdalje v WordNet v2.0.

1 Introduction

"The need to determine the degree of semantic relatedness between lexically expressed concepts is a problem that pervades much of the computational linguistics" [1]. Recent research on the topic in computational linguistics has emphasized the perspective of semantic relatedness of two words in a lexical resource, or its inverse semantic distance. Today, artificial measures of semantic relatedness are used in a wide area of natural language processing (NLP) applications, such as word sense disambiguation, automatic correction, information extraction, retrieval and indexing, text summarization and construction of ontologies.

A natural way to compute the semantic relatedness of words given a semantic network is to evaluate the distance of nodes corresponding to words being compared – the shorter the shortest path from one node to another, the more related the words are. Thus the length of the shortest path in a semantic network is named semantic distance.

On the other hand this approach to semantic relatedness has a problem of assuming that links between the nodes in a network represent uniform distances. This problem is most apparent in the most studied form of semantic network used for calculation of semantic relatedness or its specialization semantic similarity – the form of taxonomy. Attempts trying to overcome the issue of variability in the distance covered by a single semantic link inside a taxonomy include depth relative scaling [2][3] and combining taxonomy with the notion of information content [4][5].

WordNet [6] is an electronic lexical database and a broad coverage semantic network. It has been widely explored and is used in many studies of NLP. WordNet v1.7.1 is constituted of 111.223 synsets (sets of synonym words) divided into nouns, verbs, adjectives and adverbs, and more than 295.000 links of different types between them.

Up to the version 2.0 the majority of the links between synsets was of hypernym or hyponym semantic type. This is why WordNet prior to version 2.0 was considered a lexical taxonomy by most of the researchers. However, the latest edition of WordNet version 2.0 introduced more than 40.000 additional derivational links between noun and verb synsets describing morphological relatedness e.g.: noun *cook* got connected with the verb *to cook*. These new links have interconnected otherwise more or less separate taxonomy trees.

Section 2 of this paper describes the logic and the implementation of the shortest-path semantic distance measure applied to WordNet database. Section 3 explains the evaluation methods used to measure performance of the inspected measure on standard test data sets. Section 4 lists the results of evaluation of the measure on different versions of WordNet, using different test data sets. Section 5 presents and explains the acquired results.

2 Shortest-Path Semantic Distance in WordNet v2.0 (SP)

Our decision to apply shortest-path algorithm to evaluate semantic relatedness is based on the assumption that the Shortest-path semantic distance approach produces better results when applied to a denser semantic network like version 2.0 of WordNet. Our algorithm of computing semantic distance by computing shortest path between nodes in WordNet via all the available edges is similar to [7] or [8].

WordNet itself can be described as a directed graph G(V, E), where vertices set V represents a set of synsets and edge set E represents a set of directed semantic links regardless of their type. Figure 1 shows a very small

subset of WordNet. Note that each node represents one synset, which consists of a set of words. Also, each synset has links to other synsets and this links are of different types, e.g. hypernym, hyponym, antonym, meronym, etc.

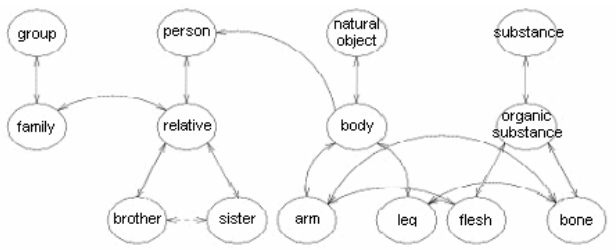


Figure 1: *WordNet nodes and edges.*

Most of the links in WordNet come in pairs like hyponym-hypernym, antonym-antonym and meronym-holonym. Newly added derivational links are also bidirectional, thus making WordNet in a large part an undirected graph.

In effect, unidirectional links allow a simple algorithm to effectively compute the shortest path between two synset sets in WordNet v2.0. Further more, a problem of computing semantic relatedness between two words can simply be translated to searching the shortest path in an undirected graph G(V, E) between the start set of vertices S (representing meanings - synsets of the first word) and target set of vertices T (representing meanings - synsets of the second word), where edge length is set to 1:

$$dist(S,T) = \min_{\substack{s\in S \\ t\in T}} path\_length(s,t) \qquad (1)$$

Our program computes the shortest path from vertex set S to vertex set T with a standard bidirectional breadth-first algorithm. Bidirectional approach is possible due to the mentioned generalization that WordNet is an undirected graph. Namely, the concept of a distance in a graph assumes undirected edges in a graph. Due to the bidirectional search technique, the breadth-first algorithm performs efficiently and produces correct results.

An example of the computed shortest path of length 4 between synset sets matching the meanings of the words *car* and *journey* is displayed in Figure 2. In the example, meanings of words *car* and *automobile* are connected through a derivational link. Meanings of words *automobile* and *driving* are connected through the category link. Our algorithm found the shortest path from meanings of word *car* to meanings of word *journey* of length 4 by examining all links to depth 2 from both directions.

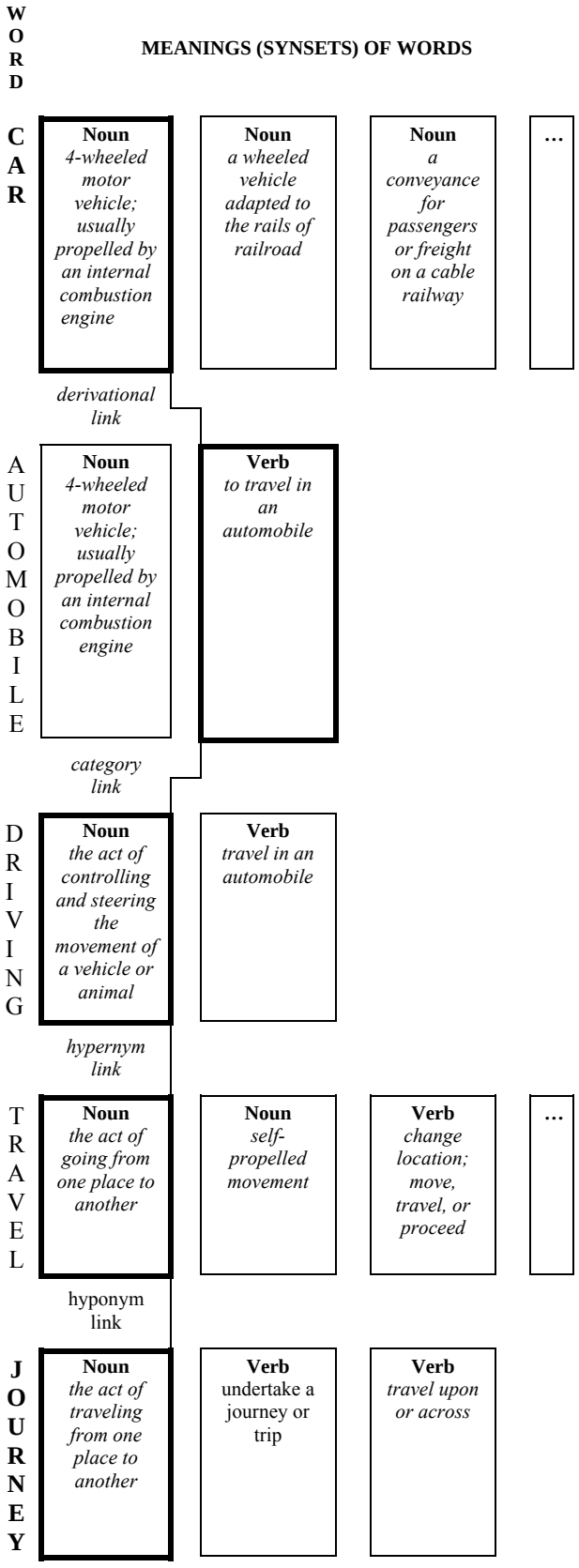


Figure 2: *Shortest path between meanings of words car and journey.*

3 Evaluation Method

We used the method of comparing the human judgment of word relatedness to computed estimates of relatedness in the evaluation of our measure.

This is one of the standard evaluation techniques [1] and arguably [4][1] yields the most general assessment of the "goodness" of a measure.

Resnik [9] has proposed this particular approach for evaluating the relatedness measure, stating that the "measure's worth is in its fidelity to human behavior, as measured by predictions of human performance on experimental tasks". The proposed evaluation method compares the computed relatedness scores with human ratings of relatedness. A series of word pairs and the average human score of relatedness between words in a pair represent human judgment of relatedness.

Scores must lie inside a predefined interval. Comparison of the relatedness grades between computed relatedness grades and those acquired by humans can be summarized by means of coefficient of correlation [9]. Resnik also argues [9] that an upper bound of the correlation coefficient between a computational measure grades and average human scores is represented by an average correlation between scores of a human individual in a repeated experiment and average human scores in an original experiment for a particular word pair set.

The problem of evaluation by comparing results against human judgment [1] is the difficulty of acquiring a larger set of human judgments of relatedness and consequently acquiring the proposed upper bound on correlation coefficient for a particular data set. Another problem of this evaluation approach according to Budanitsky [1] is that most of the applications use semantic relatedness to capture relatedness between meanings for which words are mere surrogates: "the human judgment that we need are of the relatedness of word senses not words".

On the other hand, the virtue of this evaluation approach is the ease of evaluation. Word pairs are simply submitted to evaluation of relatedness and the results are easily summarized by correlation coefficient between human and metric based results. This is why different relatedness measures are most often compared using this evaluation method. Surveys include those of Budanitsky [1] and McHale [10].

Some previous evaluations of the words relatedness measure also included reports [11] with results relative to the scope of the different embedding applications using the evaluated measure. Lin [5] proposed listing additional mathematical properties of a measure, e.g. if it presents a metric.

In this article our measure is evaluated in terms of similarity with human grades. Rated word pair data sets we used in evaluation include those of Rubenstein and Goodenough [12] (noted as 65 R&G in Tables 1,2,3), Miller and Charles [13] (noted as 30 M&C in Tables 1,2,3) and Finkelstein et al. [14] (noted as 353 F in Tables 1,2,3). The first set consists of 65 word-pair similarity grades acquired in an experiment involving 51 humans asked to rate the similarity of the word pairs on a scale of 0.0 (semantically unrelated) to 4.0 (highly synonymous). Resnik's computational measure's upper bound correlation coefficient for this data set is not known. Later, this data set was modified by Miller and Charles who extracted 30 pairs from the original 65; taking 10 pairs graded in the interval between 0 to 1, 10 pairs from 1 to 3 and 10 pairs from the grade interval 3 to 4. Resnik [4] acquired alternative human scores on the same word pair's series and argues that 0.8848 is the upper bound correlation coefficient a computational metric could achieve on this particular data set. Finkelstein et al. [14] published a greater word pair relatedness set. They acquired 353 relatedness graded word pairs, which include Miller and Charles's 30 pairs.

We used the described method of comparison against human judgment to compare our shortest path relatedness measure against other relatedness measures. Evaluation of the performance of the most of the following measures on 30 Miller and Charles's (30 M&C) word pairs and 65 Rubenstein and Goodenough's (65 R&G) word pair data sets was conducted by Budanitsky [1][11]. His results are relative to WordNet v1.5 and Brown Corpus [15], which was used as an additional knowledge source to some measures. Measures used for comparison are:

1. Hirst and St-Onge's semantic relatedness measure (HSO) [8]

The idea behind Hirst and St-Onge's measure of semantic relatedness is that two concepts are semantically close if a path, which is not too long, connects their WordNet synsets and it does not change direction too often:

$$rel_{HS}(C1, C2) = C - path\_length - K \times d \tag{2}$$

where d is the number of changes of direction in the path and C and K are constants. This measure is actually an extended shortest-path algorithm and is the only established relatedness measure, while others actually measure similarity. However, it was only evaluated on the previous version of WordNet (version 1.5).

2. Leacock and Chodorow's semantic similarity measure (LCH) [16]

Leacock and Chodorow's semantic similarity measure also uses shortest-path algorithm, however it only considers the IS-A links in the WordNet network. They modify the results by scaling them according to taxonomy depth D:

$$sim_{LC}(C1, C2) = -\log\left(\frac{length(C1, C2)}{2D}\right) \tag{3}$$

3. Resnik's similarity measure (RES) [4]

Resnik's similarity measure uses both taxonomy and corpus data. Resnik defined the similarity between two concepts lexalized in WordNet to be the information content of their most specific common subsumer – the first common predecessor in the taxonomy tree lso(C1,C2):

$$sim_{R}(C1, C2) = -\log p(lso(C1, C2)) \tag{4}$$

In equation (4), *p(c)* is the probability of encountering an instance of a concept *c* in a corpus. Finkelstein et al. [14] mentioned evaluation of the performance of the measure on the 353 word pairs.

4. Jiang and Conrath's semantic distance measure (JCN) [17]

Jiang and Conrath's semantic distance measure is an inverse of semantic similarity. The measure also combines WordNet taxonomy with corpus data. The distinction from Resnik's measure is that Jiang and Conrath's measure has the mathematical property of increasing with difference of the compared concepts.

$$dist_{JC}(C1, C2) = 2\log(p(lso(C1, C2))) - (\log p(C1) + \log p(C2)) \quad (5)$$

5. Lin's semantic similarity measure (LIN) [5]

Lin's semantic similarity measure uses the same elements and knowledge sources as does the Jiang and Conrath's semantic distance, but it is constituted according to Lin's general theory [5] of similarity between arbitrary objects:

$$sim_L(C1, C2) = \frac{2\log p(lso(C1, C2))}{\log p(C1) + \log p(C2)} \quad (6)$$

6. Roget's Taxonomy Shortest-path semantic distance (RTSP) [10]

Roget's Taxonomy Shortest-path semantic distance uses alternative taxonomy called Roget's thesaurus and is therefore not WordNet based. Roget's thesaurus is a wide shallow hierarchy densely populated with nearly 200,000 words and phrases. This method also relies on shortest path between concepts and does not use any alternative knowledge sources:

$$sim_{Roget}(C1, C2) = length_{Roget}(C1, C2) \quad (7)$$

Evaluation of the measure's performance on 28 Miller's word pairs was conducted by Mc Hale [10].

7. Latent Semantic Analysis (LSA) [18]

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. It can be used to calculate semantic relatedness based on corpus knowledge alone. The basic idea is that LSA represents the meaning of a word as a kind of average of the meaning of all the documents in which it appears, and the meaning of a document as a kind of average of the meaning of all the words it contains.

4 Empirical Measurements

We evaluated the shortest path relatedness measure by comparing the computed semantic distances against human relatedness judgment and against the results of other measures on the same data sets. Relatedness grades were computed by applying shortest path relatedness measure to all available word pair series. Correlation coefficient proposed by Resnik was used for summarizing the results and for comparison with other measures. We have performed experiments on our measure using two different versions of WordNet: WordNet v1.7.1 and WordNet v2.0.

For comparison we also used the WordNet::Similarity toolkit [19] to reevaluate Hirst and St-Onge's, Leacock and Chodorow's, Resnik's, Jiang and Conrath's and Lin's relatedness measures using WordNet versions 1.7.1 and 2.0. In this reevaluation the alternative knowledge source for extracting information content used by some measures was a much smaller SemCor [20] corpus, as the larger Brown data set could not be obtained. The SemCor, however, is a subset of the Brown Corpus used in original experiments performed by Budanitsky [1][11]. Due to this and possibly different settings of the two experiments the repeated experiment cannot be directly compared to the one performed by Budanitsky.

For evaluation of the LSA measure on 30 Miller's word pairs and 65 Rubenstein and Goodenough's word pairs we used the implementation of LSA available online at http://lsa.colorado.edu. Finkelstein et al. [14] published an evaluation on the 353 word pairs, which is presented in Table 2.

Table 1 shows the absolute correlation coefficients between the computed distance ratings of different measures and the mean ratings of human subjects per particular word-pair data set. This results were obtained using the WordNet::Similarity toolkit for each particular measure for different versions of WordNet and different test data sets. The first row of the table displays the correlation coefficients of our studied shortest path relatedness measure.

Table 2 summarizes experimental results obtained by Budanitsky in his studies of semantic relatedness [1][11] combined with results of Finkelstein et al. [14] and our research. Table 2 also lists the absolute correlation coefficients between the computed distance ratings of different measures and the mean ratings of human subjects. As described in section 3 values in Table 2 were obtained using different settings and more extensive knowledge sources as in the repeated experiment. The first two rows of the table display the correlation coefficients of our studied shortest path relatedness measure.

Table 3 lists the semantic distance ratings of our shortest path relatedness measure applied on WordNet version 2.0 per word pair for the Miller and Charles's word pair data set.

	30 M&C		65 R&G		353 F	
	WN v1.7.1	WN v2.0	WN v1.7	WN v2.0	WN v1.7	WN v2.0
SP	**0.80**	**0.86**	**0.80**	**0.88**	**0.32**	**0.47**
HSO	0.69	0.60	0.73	0.73	0.37	
LCH	0.82	0.82	0.85	0.86	0.36	0.36
RES	0.78	0.77	0.79	0.81	0.38	0.35
JCN	0.47	0.47	0.58	0.58	0.23	0.23
LIN	0.74	0.74	0.71	0.73	0.30	0.30

Table 1: *Correlation coefficients of measures compared to humans on WordNet v1.7.1 and v2.0*

Relatedness measure	30 M&C	65 R&G	353 F
SP WN v1.7.1	**0.80**	**0.80**	**0.32**
SP WN v2.0	**0.86**	**0.88**	**0.47**
HSO	0.74	0.79	
LCH	0.82	0.84	
RES	0.77	0.78	0.34
JCN	0.85	0.78	
LIN	0.83	0.82	
RTSP	0.89		
LSA	0.72	0.65	0.56
Human replication	0.88		

Table 2: *Correlation coefficients obtained by Budanitsky, Finkelstein et al. and our research.*

5 Conclusion

Results obtained in the repeated experiment of Budanitsky summed in Table 1 show that our shortest path measure, when applied to WordNet v2.0, always performs better than if applied to version 1.7.1. This can be explained by the presence of the additional derivational links in the latest version of WordNet, thus transforming it from taxonomy to a more complex semantic network.

The repeated experiment also showed that other tested measures, which are based on WordNet's taxonomy, are not significantly affected by the particular version of WordNet used. This can be explained by reviewing the differences of the taxonomy structure between versions of WordNet 1.7.1 and 2.0. According to documentation available [21] there are only insignificant differences in taxonomy part of WordNet between the two versions.

The experiments also showed that even the more complicated measure of Hirst and St. Onge, which uses the same principles of exploring all possible links in WordNet, does not compare favourably to our studied simple shortest path approach applied to the latest version of WordNet. A question arises whether the more complicated Hirst and St-Onge's measure would outperform the simpler shortest path approach if optimally applied to the latest version of WordNet. The answer remains unclear despite the results in Table 1, which show that the shortest-path algorithm produces better results. The improvement is perhaps due to suboptimal parameter settings of the more complicated measure. On the other hand the reason could lie in the fact that measures applied on prior versions of WordNet were in effect overspecialized in exploiting its prevailing taxonomic structure.

30 M&C Word pairs		Human average grades	Shortest Path on WordNet v2.0
car	automobile	3.92	0
gem	jewel	3.84	0
journey	voyage	3.84	1
boy	lad	3.76	1
coast	shore	3.7	1
asylum	madhouse	3.61	1
magician	wizard	3.5	0
midday	noon	3.42	0
furnace	stove	3.11	2
food	fruit	3.08	3
bird	cock	3.05	1
bird	crane	2.97	3
tool	implement	2.95	1
brother	monk	2.82	1
crane	implement	1.68	4
lad	brother	1.66	4
journey	car	1.16	4
monk	oracle	1.1	7
cemetery	woodland	0.95	5
food	rooster	0.89	6
coast	hill	0.87	3
forest	graveyard	0.84	5
shore	woodland	0.63	3
monk	slave	0.55	4
coast	forest	0.42	4
lad	wizard	0.42	4
chord	smile	0.13	7
glass	magician	0.11	6
noon	string	0.08	7
rooster	voyage	0.08	10

Table 3: *Semantic distance by item in the Miller's word pair data set.*

The comparison of results in Table 2 reveals that when applied to the latest version of WordNet, our studied shortest path measure performs at least as well as the other more complicated measures based mostly on taxonomy of WordNet version 1.5. This can only result from additional features in WordNet v2.0 compared to WordNet v1.5. The shortest path measure performs comparatively or better than the measures combining WordNet taxonomy with the notion of information content. This indicates that WordNet v2.0 has become a much more informative and dense semantic network than the previous versions.

Results in Table 2 also show that shortest-path semantic distance in WordNet v2.0 is comparable to other relatedness measures based on alternative knowledge

sources. The shortest-path semantic distance measure applied to Roget's thesaurus performs only slightly better than the studied one applied to WordNet v2.0, suggesting that the knowledge of both semantic networks exposed by their nodes and edges is approximately equal. The LSA method based solely on corpus data, on the other hand, shows strongest performance resilience to enlarging test data sets. This suggests further improvements to our measure are possible and should be studied in the already promising direction of combining WordNet with additional knowledge sources like corpora.

From the results one might conclude that the studied shortest path relatedness measure applied to WordNet v2.0 gives better results than other measures based only on WordNet taxonomy regardless of the version of WordNet and regardless of the possible use of alternative data sources. This follows from the assumption that even if Budanitsky repeated his experiment on the taxonomy of WordNet version 2.0 it could have only yielded approximately the same results as with the version 1.5. The latter assumption is supported by (1) the fact that the repeated experiment produced WordNet version independent results for the taxonomy based measures and by (2) the fact that according to documentation [21], there are no significant differences in the taxonomy part of WordNet between versions 1.5 and 2.0.

According to evaluation against human ratings of relatedness, pretty basic shortest-path semantic relatedness measure applied to WordNet version 2.0 can be used in research and development of the NLP systems instead of the more complicated alternatives. The shortest-path method will typically use fewer resources to achieve similar or better results.

References

[1] Budanitsky, A., and G. Hirst, *Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures*, Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Pittsburgh, PA, 2001.

[2] Sussna, M., *Text Retrieval using Inference in Semantic Matanetworks*, PhD Thesis, University of California, San Diego, 1997.

[3] Wu, Z., and Palmer, M., Verb *Semantics and Lexical Selection*, Proc. of the 32nd Annual Meeting of the Associations for Computational Linguistics, 1994.

[4] Resnik, P., *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language*, Journal of Artificial Intelligence Research, 11, 1999.

[5] Lin, D., *An Information-Theoretic Definition of Similarity*, Proceedings of 15th International Conf. on Machine Learning, 1998.

[6] Fellbaum, C. (Ed) *WordNet – An Electronic Lexical Database*, MIT Press, 1998.

[7] Rada, R., Mili, H., Bicknell, E., and Blettner, M., *Development and application of metric on semantic nets*, IEEE Transaction on Systems, Man, and Cybernetics, 1989, Vol. 19, no 1, pp 17-30.

[8] Hirst, G., and St-Onge, D*., Lexical Chains as representations of context for the detection and correction of malapropisms*, Fellbaum, 1998, pp 305-332.

[9] Resnik, P., *Using information content to evaluate semantic similarity*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, 1995, pp 448-453.

[10] McHale, M., L., A Comparison of WordNet and Roget's Taxonomy for Measuring Semantic Similarity, In *Proc. Usage of WordNet in Natural Language Processing Systems*, COLING-ACL '98 Workshop, August 16, 1998, University of Montreal, 1998.

[11] Budanitsky, A. *Lexical Semantic Relatedness and its Application in Natural Language Processing*, technical report CSRG-390, Department of Computer Science, University of Toronto, 1999. http://www.cs.toronto.edu/compling/Publications/Abstracts/Theses/Budanitsky-thabs.html

[12] Rubenstein, H., and Goodenough, J. B., *Contextual Correlates of Synonymy*, Computational Linguistics, 8, 1965, pp 627-633.

[13] Miller, G.A., and Charles, W.G. *Contextual correlates of semantic similarity*, Language *and Cognitive Processes*, Vol. 6, No. 1,1991, pp 1-28.

[14] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E., *Placing Search in Context: The Concept Revisited*, ACM Transactions on Information Systems, 2002, Vol. 20, no 1. pp 116-131.

[15] Francis, W.N. and Kučera, Henry, Frequency analysis of English usage. Lexicon and grammar. Houghton Mifflin, Boston, 1982.

[16] Leacock, C., and Chodorow, M., *Combining Local Context and WordNet Similarity for Word Sense Identification*, C. Fellbaum, editor, WordNet: An Electronic Lexical Database, MIT Press, 1998, pp 265-285.

[17] Jiang, J., J., and Conrath, D., W., *Semantic similarity based on corpus statistics and lexical taxonomy*, Proceedings on International Conference on Research in Computational Linguistics, Taiwan, 1997.

[18] Landauer, T.K., Foltz, P.W., and Laham, D. *Introduction to Latent Semantic Analysis*, Discourse Processes, Vol. 25, No. 2 & 3, 1998, pp 259-284.

[19] Pedersen, T., *WordNet::Similarity*, http://www.d.umn.edu/~tpederse/similarity.html

[20] Miller, G., Leacock, C., Tengi, R. and Bunker, T., *A Semantic Concordance*, Proc. of ARPA Workshop on Human Language Technology, 1993.

[21] *WordNet Documentation*, Cognitive Science Laboratory, Princeton University, ftp://ftp.cogsci.princeton.edu/pub/wordnet/.

Multi-Attribute Modelling of Economic and Ecological Impacts of Cropping Systems

Marko Bohanec, Sašo Džeroski and Martin Žnidaršič
Institut Jožef Stefan, Department of Knowledge Technologies, Jamova 39, SI-1000 Ljubljana, Slovenia
marko.bohanec@ijs.si

Antoine Messéan
CETIOM, Centre technique interprofessionnel des Oléagineux métropolitains and
INRA Eco-Innov, POB BP1, Centre de Grignon, FR-78850 Thiverval, Grignon, France

Sara Scatasta and Justus Wesseler
Wageningen University, Environmental Economics and Natural Resources Group,
De Leeuwenborch, Hollandseweg 1, NL-6706 KN Wageningen, The Netherlands

Keywords: qualitative multi-attribute modelling, genetically modified plants, cropping system, ecology, economy

Received: June 27, 2004

Modelling of economic and ecological impacts of genetically modified crops is a demanding task. We present some preliminary attempts made for the purpose of the ECOGEN project "Soil ecological and economic evaluation of genetically modified crops". One of the goals of the project is to develop a computer-based decision support system for the assessment of economic and ecological impacts of using genetically modified crops, with special emphasis on soil biology and ecology. The decision support system will be based on a rule-based model incorporating both economic and ecological criteria. In this paper we present some preliminary results of developing the integral model and describe four specific sub-models. The first two sub-models are concerned with ecology and assess the ecological impacts of various types of weed and pest control, respectively. The other two sub-models assess the economic impacts of cropping systems at the farm and regional level, respectively. All the models were developed using a qualitative multi-attribute modelling methodology, supported by the software tool DEXi.

Povzetek: članek opisuje modeliranje ekoloških problemov kmetijstva.

1 Introduction

The possible use of genetically modified (GM) plants in agriculture needs in-depth investigations of ecological and economic consequences [1,2]. The investigations are important for both the European Commision (EC), who needs specifications for GM-plant risk assessment, and to farmers and the public who are concerned about the possible ecological and economic implications. Crop production involves complex decision-making processes, which require and justify the application of decision support systems [3].

The ECOGEN project [2] (*Soil ecological and economic evaluation of genetically modified crops*) is an EC-funded project aimed at combining simple lab tests, studies of multi-species model ecosystems, and field studies to acquire mechanistic and realistic knowledge about economic and ecological impacts of GM crops on the soil. Economic trade-offs are assessed and related to ecological effects. The economic and ecological knowledge gained in ECOGEN will be combined into a rule based model for a decision support tool.

The goals of the ECOGEN project are to:

1. Provide ecological and economical assessment and comparison of integrated cropping systems using GM or conventional crops, respectively.
2. Provide an ecological risk assessment of a GM cropping system and a conventional cropping system for the soil ecosystem based on single species tests, multispecies tests and long-term field investigations.
3. Adapt existing ecotoxicity testing tools to GM plant material and validate their use.
4. Provide economic assessment of GM crops and conventional crops with respect to a quantification of the expected trade-offs between the two and the implications for the EU Agriculture Policy.

Finally, we wish to incorporate ecological knowledge from single species tests, multispecies tests, and field investigations, as well as economic information from farming practices into a *rule-based model* to be used for predictions of economic decision-making processes and ecosystem behaviour.

In this paper, we present our preliminary attempts at this kind of modelling. We describe four qualitative multi-attribute models. Two of these models assess the ecological impacts of using various cropping systems that differ in the applied weed-control and pest control mechanisms, respectively. The other two models assess the economic impacts of cropping systems; these are assessed at farm and regional level, respectively.

2 Methodology

The goal of the project is to build an integrated rule-based model for assessing the sustainability of farming (GM and non-GM) taking into account ecological and economic aspects. On the ecological side, this includes a model of the impact of GM crops and pesticides on non-target organism and soil functions. The model will be hierarchically structured, with submodels for different aspects, e.g., a submodel for economic (ECONOMY) and a submodel for ecological (ECOLOGY) aspects. In general, then, the approach will involve the following components (Figure 1):

1. *Cropping systems*: Input items assessed by the model. Each cropping system is described by a vector of values, such as: crop type (e.g., corn), soil preparation (e.g., type of tillage), weed control (e.g., use of herbicides), pest control (e.g., use of pesticides), type and quantity of fertilization, soil characteristics, climate characteristics, economic indicators (e.g., involved yields and variable costs).
2. *Multi-attribute model*: A model that aggregates the characteristics of cropping systems into overall ecological and economic evaluations.
3. *Outputs*: Two assessments are obtained for each cropping system: ecological and economic impacts.

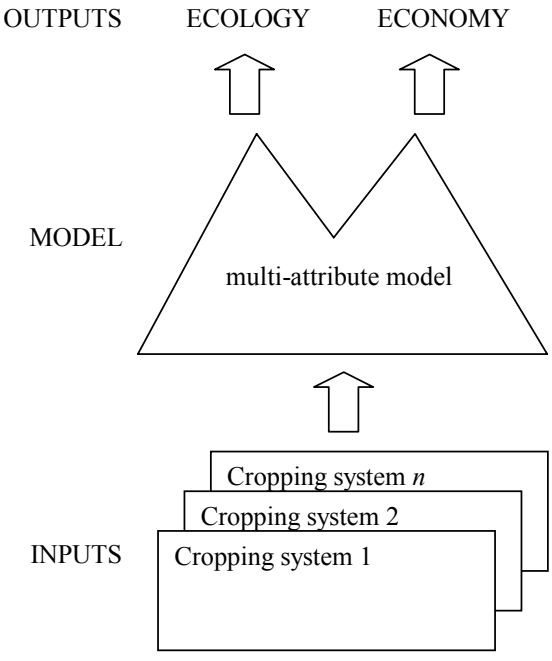


Figure 1: A general approach to multi-attribute assessment of cropping systems.

Using this schema, it will be possible to evaluate each cropping system and its impacts for several consecutive years, basically obtaining a chart as sketched in Figure 2. In addition, the model will facilitate all analyses and reports typically available in multi-attribute modeling [4,5]: what-if analysis, sensitivity analysis, simulation, selective explanation, and various visualizations.

The integral multi-attribute model will be mainly rule-based and will contain further submodels, which will be both qualitative (using rules) and quantitative (numerical/equations). They will be developed by the soil biology experts in the respective subareas, and in intensive interaction and collaboration with the decision support/data analysis experts. Decision support methods that rely on manual knowledge acquisition from domain experts will be used to elicit existing knowledge. Techniques from the area of multi-attribute decision-making and support will be used to support the construction of the overall model.

Where enough data are available, some submodels will be generated in a (semi)automated fashion by data analysis. In particular, machine learning techniques will be used to construct some submodels by analysing available data. Some sub-models of this type have already been developed in this way [6].

Reasoning with the rule-based model for decision support is crisp by default, but can be extended to fuzzy reasoning with moderate effort.

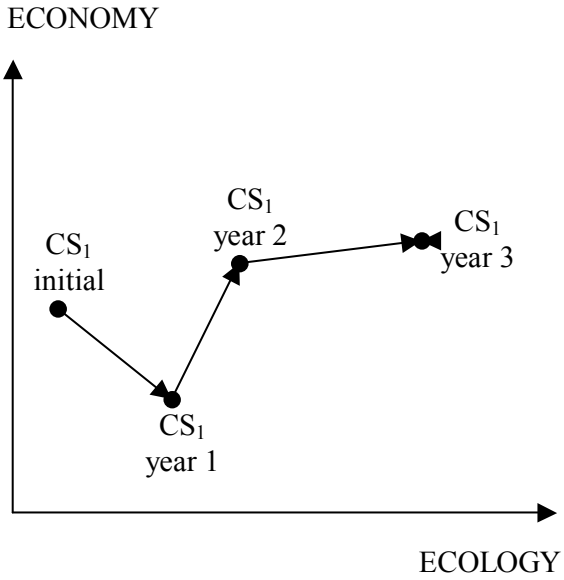


Figure 2: An example assessment of a cropping system CS1 through three consecutive years.

So far, we have developed two models for the ecological assessment of cropping systems, dealing with weed-control and pest-control mechanisms, respectively. These are described in section 3. In addition, we have developed two models for the assessment of economic impacts of cropping systems. The first model assesses the impacts at the farm level. The second model is an extension and adaptation of the

first one so as to assess the economic impacts at the regional level. These are presented later in section 4.
All these models are hierarchical, qualitative and multi-attribute. Thus, they are characterised by the following [4]:

- Each model consists of a number of hierarchically structured variables called *attributes*.
- Terminal nodes of the hierarchy *represent input attributes*; each cropping system is described by a vector of values of input attributes.
- Input attributes are aggregated through several levels of *aggregate attributes* into the overall assessment, which is represented by a single *root attribute*.
- All the attributes in the model are *qualitative*, meaning that they take symbolic values, described by words.
- The aggregation of values in the model is defined by *rules*.

The models were developed using the software tool DEXi [7]. DEXi facilitates the development of a tree of attributes, definition of aggregation rules (e.g., see Figure 4), evaluation of options (cropping systems in this case), what-if analysis and charting.

3 Ecological assessment

3.1 Weed-control model

With this model, cropping systems are assessed qualitatively using the five-value ordered scale: preferrable, acceptable, regular, poor, unacceptable. The model is hierarchical and has the structure of attributes as shown in Figure 3.

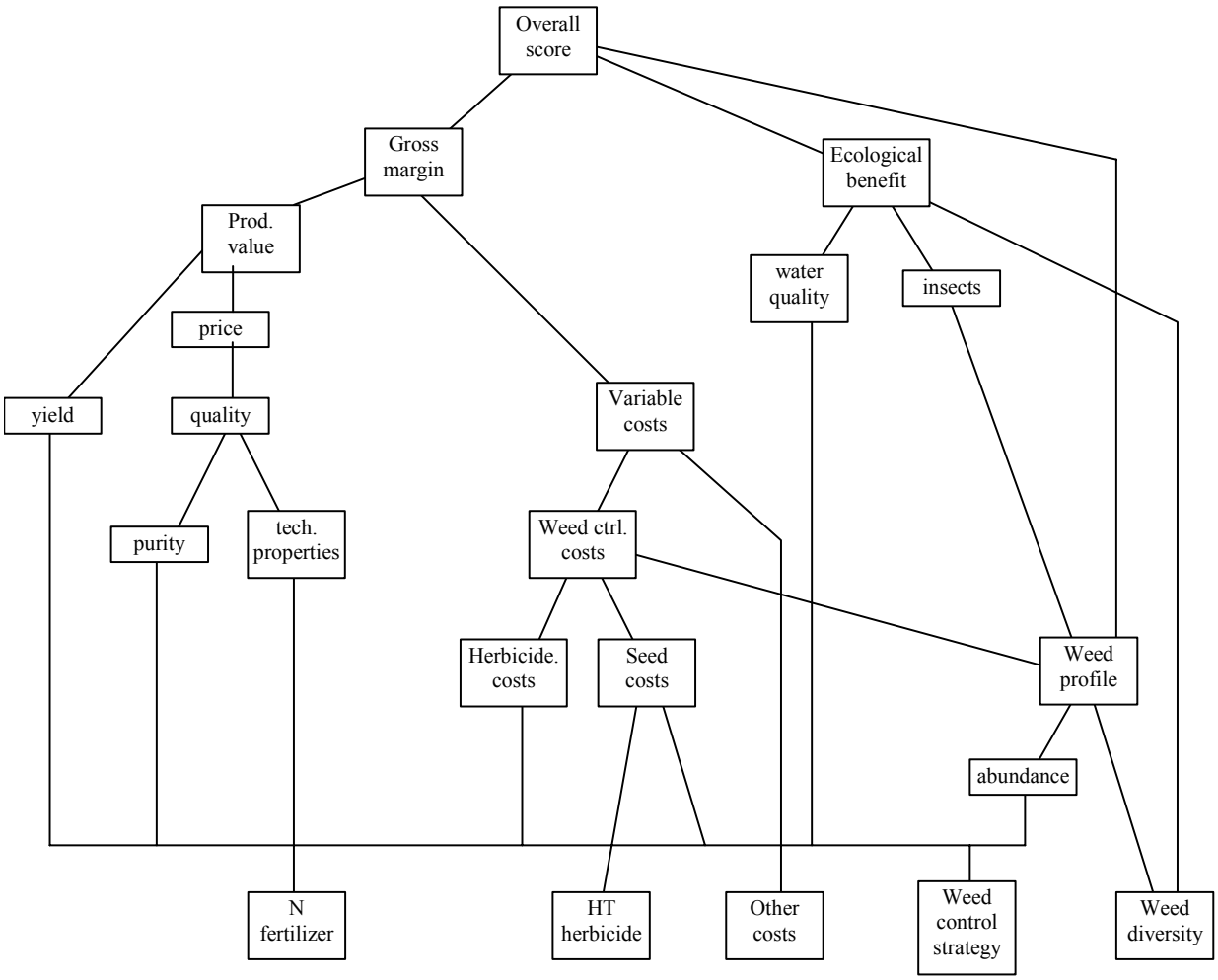


Figure 3: The hierarchical structure of the weed-control model.

Input attributes. The assessment of cropping systems is based on six input attributes:

1. *Weed_control_strategy*: the strategy of controlling weeds, either for conventional crops or for GM crops. The GM crops considered in this study are herbicide-tolerant (GMHT). There are six different strategies:
   a. Non-GM and simple pre-sowing herbicide application (pre_sowing trifluralin);
   b. GMHT with one fall application of herbicide;
   c. GMHT with fall + spring applications of herbicide;
   d. Non-GM with pre-sowing+pre-emergence applications of herbicide;
   e. Non-GM with pre-sowing+post-emergence applications of herbicide;
   f. Non-GM with post-emergence application of herbicide only.
2. *HT_herbicide*: the application of HT herbicide: glyphosate, none, glufosinate, or generic.
3. *N_fertilizer*: the application of nitrogen fertilizer: high (>200 kg/ha), medium (150–200 kg/ha), or low (<150 kg/ha).
4. *Weed_diversity*: the diversity of weed at the location studied: high, medium, or low.
5. *Other_costs*: relative estimation of marginal costs other than weed control costs: high, medium, or low.
6. *Fixed_costs*: relative estimation of the fixed cost of production: high, medium, or low.

Aggregate attributes. The aggregate (intermediate) attributes are grouped into three main subtrees:

1. *Weed profile* is an aggregate sub-model that affects several other parts of the model. Basically, it defines the weed profile according to the abundance and diversity of weeds.
2. *Gross margin* is estimated on the basis of Production value and Variable costs of production. Production value depends on yield and quality, which in turn depends on purity and technological properties of production. Variable costs are estimated on the basis of herbicide costs, seed costs and weed profile.
3. *Ecological benefit* is estimated according to water quality, insects and weed diversity. The effect on insects is assessed through weed profile.

All the aggregate attributes in the model are assessed according to rules defined by an expert. Figure 4 shows two such rulesets, conveniently presented in a tabular form. The bottom ruleset defines the mapping between the input attribute *Weed_control_strategy* to the aggregate attribute *Abundance*. The top ruleset defines the rules that combine *Abundance* and *Weed_diversity* into the aggregate attribute *Weed_profile*.

	Abundance	weed_diversity	Weed_Profile
	22%	78%	
1	***high***	<=medium	high_potential_problems
2	<=medium	***high***	high_potential_problems
3	medium	medium	regular_problems
4	**low**	***high***	regular_problems
5	**low**	>=medium	low-problematic
6	<=medium	**low**	specific_flora_problems

	weed_control_strategy	Abundance
	100%	
1	simple pre-sowing	***high***
2	HT one fall application	medium
3	>=pre-sowing+pre-emergence	medium
4	HT fall + spring applications	**low**

Figure 4: Two tables of aggregation rules: for Weed profile and Abundance of weed.

3.2 Pest-control model

In addition to the weed-control model, we have also developed a pest-control model. Its structure is similar to the weed-control model, with the following differences:

- The subtree *Weed control* is replaced by a subtree *Pest control*, having similar structure, but different attribute values and aggregation rules.
- Two new attributes are added to the node *Ecological benefit*: *Greenhouse gasses* and *Soil*. *Soil* is in turn composed of *Soil fauna* and *Soil quality*.
- The attribute *Purity* is replaced by *Damage*.

4 Economic assessment

Figure 5 shows the structure of the ECONOMY submodel for the assessment of economical indicators. This model assesses gross margins at the *level of a single farm* for *Bt-corn*. [Bt-corn has been genetically engineered to produce an insecticide known as Bt-toxin, produced by a naturally occuring soil organism Bacillus thuringiensis (Bt).] In addition to the ECONOMY submodel, some possible links for the ECOLOGY submodel are also shown in Figure 5.

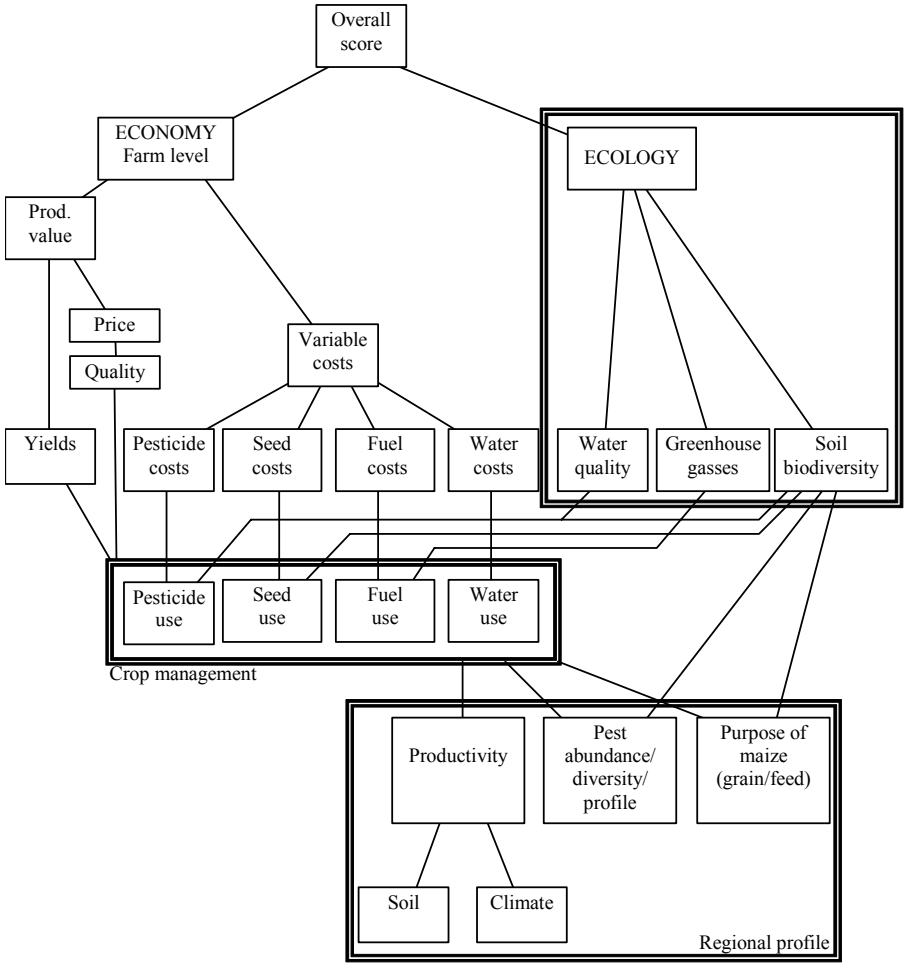


Figure 5: Hierarchical structure of the ECONOMY model (Bt-corn, farm level).

Basically, the farm level impacts in terms of economy depend on *Production value* and *Variable costs*. The former can be determined on the basis of *Price* and *Yields*, where *Price* depends on the *Quality* of production.

Variable costs incorporate the costs of *Pesticides*, *Seeds*, *Fuel*, and *Water*. Each of these costs directly depends on the *use* of the respective item: pesticides, seeds, fuel, and costs. The used quantities of each of these items, as well as *Yields* and *Quality*, directly depend on the cropping system employed at the farm level. Thus, these four variables form a group referred to as *Crop management*. Notice that some of these indicators affect the ECOLOGY part of the model, too. For instance, *Pesticide use* influences *Water quality* and *Soil biodiversity*. The latter is affected by *Seed use*, too. *Fuel use* influences *Greenhouse gasses*.

The lowest level of attributes form the group called *Regional profile*. These attributes describe the properties of a particular region in terms of: *Productivity*, *Pest abundance*, *Purpose of maize* (which can be used for grain or feed). *Productivity* depends on the characteristics of *Soil* and *Climate* in the region. Notice that all the three main attributes of *Regional characteristics* affect the *Crop management* group, and that two of them (*Pest abundance* and *Purpose*) additionally affect *Soil biodiversity*.

When we move from the farm level to the *regional level*, a new important factor comes into play: *Adoption rate*. Namely, when assessing a cropping system at the farm level, it is clear whether Bt-corn has been adopted in that system or not. It can only be adopted or not adopted, there are no intermediate choices. However, when assessing a cropping system at the regional level, it becomes important which proportion of the farms have adopted Bt-corn, because the adoption rate influences the *Regional yields* and *Regional costs* of the crop. Also, the adoption rate itself can be influenced by the *Price* in the market, which introduces a cycle into the model.

All this is reflected in the strucure of the ECONOMY model for Bt-corn at the regional level (Figure 6). The structure is very similar to the structure of the model at the farm level (Figure 5), except that there is an additional block appearing above the *Crop management* block. This new block assesses *Adoption rate* based on *Field trial yields* and *Field trial costs*. Both of these variables depend on the indicators of

Crop management, which in turn depend on the assessed cropping system. Once determined, *Adoption rate* directly influences *Regional yields* and *Regional costs*.

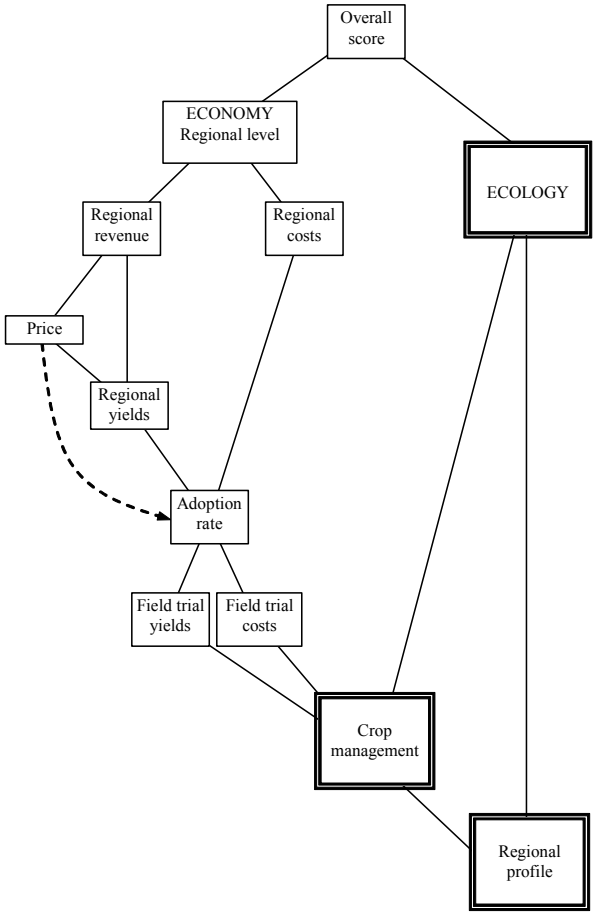


Figure 6: Hierarchical structure of the ECONOMY model (Bt-corn, regional level).

5 Conclusion

The modeling of economic and ecological impacts of genetically modified crops is inherently difficult. It requires knowledge from different fields and disciplines, which is scarce and largely unknown. It also requires complementary approaches, such as a combination of data mining and expert modeling, which has been attempted in ECOGEN. The benefits of modeling, however, are manifold, as it facilitates various computer-based assessments, evaluations, analyses and simulations. The results are eagerly awaited by European administration, politicians, ecologists, farmers and the interested public.

The models presented in this paper provide a preliminary step in this direction. They are in an early development stage and a lot of further work is expected. First, the models should be tested using real field data which is being collected. The models and their results should be evaluated by relevant experts. Second, the developed models are truly hierarchical (as opposed to traditional tree-like structure) and involve some very complex relationships between attributes, even cycles. These characteristics exceed the capabilities of currently available supporting software, which will have to be accordingly modified and extended. Last but not least, the developed models provide just a part of the final integral model, which will address additional cropping system control mechanisms and additional GM crops.

Acknowledgement

This work was supported by the ECOGEN project, funded by the Fifth European Community Framework Programme: *Quality of life and management of living resources*, under contract QLK5-CT-2002-01666.

References

[1] Birch, A.N.E., Krogh, P.H., Cortet, J., Tabone, E., Griffiths, B.S., Džeroski, S., Wesseler, J., Gomot de Vaufleury, A., Badot, P-M., Andersen, M.N., Messéan, A. (2003). ECOGEN: Soil ecological and economic evaluation of genetically modified crops. Poster at ”Biodiversity Implications of Genetically Modified Plants”, September 7-12, 2003 Monte Verità , Ascona, Switzerland Centro Stefano Franscini, Swiss Federal Institute of Technology Zürich.

[2] ECOGEN: Soil ecological and economic evaluation of genetically modified crops. http://www.ecogen.dk

[3] Recio, B., Rubio, F., Criado, J.A. (2002): A decision support system for farm planning using AgriSupport II. *Decision Support Systems* 36, 189–203.

[4] Bohanec, M. (2003): Decision support. In: Mladenić, D., Lavrač, N., Bohanec, M., Moyle, S. (eds.): *Data mining and decision support: Integration and collaboration*. Kluwer Academic Publishers.

[5] Triantaphyllou, E. (2000): *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers.

[6] Demšar, D., Džeroski, S., Larsen, T., Krogh, P.H. (2003): Identifying most important agricultural factors for soil community of microarthropods. In Proc. of the 12th International Electrotechnical and Computer Science Conference. 25.–26. September 2003, ERK'2003, Ljubljana.

[7] Jereb, E., Bohanec, M., Rajkovič, V. (2003): *DEXi: Računalniški program za večparametrsko odločanje [DEXi: Computer program for multi-attribute decision making*]. Moderna organizacija, Kranj, Slovenia.

Software Development with Grammatical Approach

Tomaž Kosar, Marjan Mernik and Viljem Žumer
University of Maribor, Faculty of Electrical Engineering and Computer Science, Slovenia
E-mail: {tomaz.kosar, marjan.mernik, zumer}@uni-mb.si

Pedro Rangel Henriques
University of Minho, Department of Computer Science, Portugal
E-mail: prh@di.uminho.pt

Maria João Varanda Pereira
Polytechnic Institute of Bragança, Portugal
E-mail: mjoao@ipb.pt

Keywords: software design and modelling, software development, context-free grammars, attribute grammars, rapid prototyping

Received: July 12, 2004

The paper presents a grammatical approach to software development. It supports formal software specification using attribute grammars, from which a rapid prototype can be generated, as well as the incremental software development. Domain concepts and relationships among them have to be identified from a problem statement and represented as a context-free grammar. The obtained context-free grammar describes the syntax of a domain-specific language whose semantics is the same as the functionality of the system under implementation. The semantics of this language is then described using attribute grammars from which a compiler is automatically generated. The execution of a particular program written in that domain-specific language corresponds to the execution of a prototype of the system on a particular use case.

Povzetek: članek opisuje razvoj programov na osnovi slovnice.

1 Introduction

One of the well known properties of software systems is that they are subject to frequent changes. A software developer needs to build a software system in such a manner that he can easily adapt it to the user's changeable requirements. Current object-oriented design techniques [7] [8] are well suited for such design supporting changes. However, any changes during the software life cycle are costly. Therefore, it is very important that the user is involved in the software development process from the very beginning and that the software system is delivered to the user before his requirements have the opportunity to change. Rapid prototyping enables the software developer to build executable prototypes and to involve the user in an iterative build-execute-modify loop until his requirements are validated. The prototype is then used to build the final version of the software system through the use of the architecture included in the prototype or it is simply thrown away [21]. In the latter case the prototype is used to clarify the user's needs until reaching a stable and convenient model for the given problem.

The proposed approach, i.e. *software development with grammatical approach*, rests on the success reached by attribute grammars in the specification of language semantics [12] [6] [16] and in the systematic implementation of language processing tools [9] [10].

In the paper the grammatical approach to problem solving supported by an attribute grammar developed and written in an object-oriented style (OOAG - object-oriented attribute grammar) is proposed. One of the benefits of the proposed approach is that it enables rapid prototyping and the validation of the user's requirements in a pragmatic way. The idea is to translate the OOAG obtained in the specification phase into the concrete syntax of a compiler generator in order to create a simulator for that problem. We can then write scenarios (in the domain-specific language [17] [22] [24] defined by that OOAG) describing different uses of the system, and use the generated simulator to process those scenarios computing the desired results.

The organization of the paper is as follows. In Section 2 related work is discussed. The software development with grammatical approach is presented in detail in Section 3 followed by an example in the Section 4. A synthesis and concluding remarks are presented in Section 5.

2 Related Work

The grammatical approach to problem solving (software development) can be seen as an extension (e.g. as in [15]) of object-oriented design methods [20] [7] [8] where a problem domain model is developed from use cases and class diagram. However, their main goal is to develop good software models. Our goal is to develop rapid prototypes and early validation of user's requirements.

Our work is closely related to the Grammar-Oriented Object Design (GOOD) [2] [14], where all valid object interaction sequences of the cluster of objects are identified. Then a meta-model is constructed and represented as a context-free grammar. Therefore, a context-free grammar represents the set of all possible interactions (collaborations) of objects in a particular cluster in order to fulfill the domain goals. When a grammar is interpreted at a run-time a cluster will dynamically bind the collaborators to the collaborations. Hence, GOOD facilitates the creation of dynamically configurable components, which encapsulates volatile business rules. The rationale behind this is that creating and representing a model of solutions is more extensible, simpler and more scalable than just creating the single solution. Possible solutions are modeled with a meta-model and represented as a context-free grammar. If this grammar is available to the "users" at run-time, then they are able to customize the system behavior. Since the interaction of objects is obtained from use case diagrams that describe the functionality of a system, the author called such a grammar a use case grammar. In other words, use cases are described with a domain-specific language. In the domain analysis the key abstractions are identified and classified as interactions among subsystems that may be realized as software components. The author in his work distinguishes two types of meta-models: the static (class diagram) and the dynamic (valid object interaction sequences) meta-model. The latter is described with a context-free grammar. Our approach differs from [2] [14] since they are using a context-free grammar to describe behavior of the objects (methods), while in our case the structure of a class (attributes) is described. An example of a production rule in [2] [14] using the EBNF is:

```
ShoppingCartOperation ::=
  {AddItem | DeleteItem |
   SaveShoppingCart} CheckOut
```

Our approach has different goals and advantages. However, it can be seen as complementary to the GOOD approach. Combining both approaches to describe the behavior and the structure with a domain-specific language, is under investigation.

The grammatical approach to software development is also related to the rapid prototyping research (e.g. [4]). In [4] Two-Level Grammars (TLG) were proposed as an object-oriented requirement specification language. Successive refinement steps starting with natural language lead to more detailed specifications that can be translated to VDM++, which in turn is translated to Java, yielding a rapid prototype of a system. With this approach it is possible to obtain the rapid prototype of a system from natural language specifications. Their Specification Development Environment (SDE) has natural language parsing capabilities and can classify words into nouns (objects/class) and verbs (operations) and their relationships. This initial analysis of requirement documents provides the basis for further refinement with an attempt to classify the domains (classes) to which functions (operations) belong. In more complex cases a rapid prototype is not completely automatically derived since a sufficient degree of interaction with a user is required to ensure a correct interpretation.

Resolving the semantical gap between use case diagram and class diagram is also presented in [19]. From the use case diagram agents state machines and values added invariants are derived. The term agent is used to represent an actor collaborating with the system through specific use case. Both techniques are collectively used in iterative converting algorithm, which builds the OCL specification and class diagram. The OCL specification (define a set of preconditions, postconditions and actor invariants) are further used to check the correctness of the model.

3 The Grammatical Approach

To achieve a good understanding of the user's world we need to understand the application domain. In other words, we need to identify concepts and their relationships in the problem domain. For this purpose object-oriented design (OOD) employs use case diagrams (UCDs) and conceptual class diagrams (CCDs) [7] which we will take as a starting point for our approach.

The use case diagram [5][1] describes the functionality of the system and its interaction with an environment. The use case diagrams form foundations for further modelling of developing system. They are also helpful for generating system test cases.

While use case diagrams are narrative descriptions of specific tasks, the conceptual class diagram captures concepts and relationships between them. Guidelines for developing the conceptual class diagram can be found in [20]. To develop the conceptual class diagram one can apply iteratively the following steps:

- identification of potential classes (look for nouns in the description of the problem),
- elimination of unnecessary (eg. redundant, irrelevant) classes,
- identification of potential associations (any dependency between two classes is an association),
- elimination of unnecessary associations,
- identification of attributes (attributes are properties of individual objects),

– elimination of unnecessary attributes,

– refining with inheritance.

From the use case diagram and from the conceptual class diagram a design model is obtained which should be robust with respect to changes of the user's requirements.

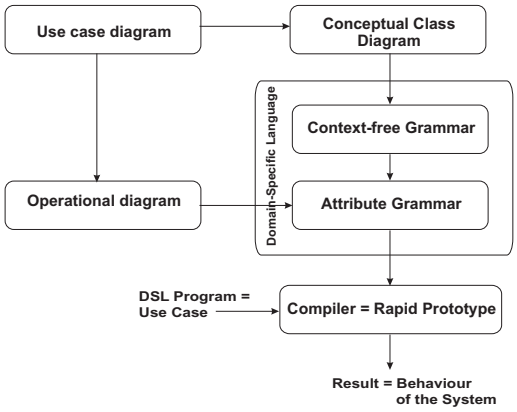


Figure 1: High-level view of the grammatical approach

To identify concepts and their relationships in the problem domain our grammatical approach is not limited to object-oriented design. Also other approaches, such as entity-relation diagrams and data-flow diagrams, which show the flow of work and the relationship between activities and deliverables, can be applied. However, OOD [3] [8] is now almost the-facto standard for software system design, and on account of that, it was also our choice.

Our approach (described in Fig. 1) is based on the following steps:

– describe the syntax of the problem (the structure of the classes that characterise problem domain), deriving the context-free grammar from the conceptual class diagram,

– describe the semantics of the problem (the meaning of the classes in problem domain), associating attributes to every concept derived from the use cases and operational diagrams,

– generate a rapid prototype of the system, using a compiler generator and the attribute grammar obtained in the two previous steps.

The steps above will be detailed in the next subsections.

3.1 Deriving a context-free grammar from a conceptual class diagram

The role of non-terminals in a context-free grammar is two fold. First, at higher abstraction level non-terminals are used to describe different concepts in the programming language (e.g. an expression or a declaration in a general-purpose programming language). On the other hand, at a more concrete level, non-terminals and terminals are used to describe the structure of a concept (e.g. an expression consists on two operands separated by an operator symbol, or a variable declaration consists of a variable type and a variable name). Therefore, both the concepts and relationships between them, belonging to the specific problem domain, are captured in a context-free grammar. But, this is also true for the conceptual class diagram which describes concepts in a problem domain and their relationships. It is clear that both formalisms can be used for the same purpose and that some rough transformation from a conceptual class diagram to a context-free grammar and vice versa should exist. The transformation from a conceptual class diagram to a context-free grammar is depicted in table 1 and table 2. In general, classes are mapped to non-terminal symbols and instance variables are mapped to terminal symbols.

Transformation table shows how to derive a context-free grammar from a conceptual class diagram. A class and a non-terminal are basic concepts in a conceptual class diagram and in a context-free grammar. The mapping here is self-evident. A conceptual class diagram contains instance variables, which define the state of a class instance. Instance variables are represented in a context-free grammar as terminal symbols. In general, a class diagram consists also of operations, which will be identified when the semantics of context-free grammar is going to be defined. Associations represent the interaction between classes and have to be included in a context-free grammar. The navigability association can be shown with the production $A \rightarrow B$, where the non-terminal A gets information about attributes of the non-terminal B. Association has multiplicity. Describing multiplicity with grammar productions is straightforward as shown in table 2. For generalization we propose the production $A \rightarrow B \mid C$. The non-terminal A can be implemented either with the non-terminal B or non-terminal C. The composition and aggregation are shown as the navigability association. In the composition the non-terminal B can appear in other productions. On the other hand, in the aggregation the non-terminal B is reachable only from the non-terminal A.
Classes can collaborate with more than just one class. For example, a class A associates with classes B, C and D. In our approach, this collaboration is described with context-free grammar production $A \rightarrow B\,C\,D$. The sequence of non-terminals on right side of the production should be in natural order and depends on collaboration of entities in a given problem domain.

3.2 Describing the semantics of each concept

To describe the semantics or the meaning of a concept an attribute grammar is used. Attribute grammars [12] [6] [16] are natural extensions of context-free grammars and as such very well support our approach which is based on context-free grammars. The syntax and semantics of each symbol is specified in a module; modularity is, on one hand, inherent to the class concept in OOD, and, on

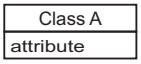
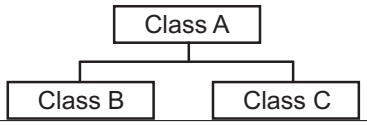
Description	Class diagram element	Grammar
Class		A (non-terminal) instance variable (terminal)
Association		A → B
Navigability		A → B
Generalization		A → B | C
Composition		A → B $(\neg\exists X \in N, X \Rightarrow B) \wedge X \neq A$
Aggregation		A → B

Table 1: From a conceptual class diagram to a context-free grammar

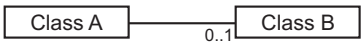
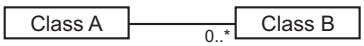
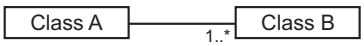
Cardinality	Class diagram element	Grammar
Multiplicity exactly one		A → B
Optional multiplicity		A → B | ϵ
Multiplicity [0..m]		A → MoreB MoreB → MoreB B| ϵ
Multiplicity many		A → MoreB MoreB → MoreB B | B

Table 2: Association multiplicity

the other hand, it is implicit to grammars (based on the locality associated with symbols and productions). The first part of a module is the declaration of its attributes, divided in two subsets, the inherited (context dependent) and the synthesized (computed locally). The functions to be used to evaluate each attribute are then defined, in the context of each production. Also the contextual conditions, if any, that express the data constraints are defined in the context of each production. This step is intellectually most demanding; therefore some additional supporting techniques based on the use cases (diagrams and scenarios) should be used; namely we suggest the use of the operational diagram that is infered from the referred scenarios.

The result of this step is a complete attribute grammar specification for a given problem.

3.3 Generating the rapid prototype of a system

To generate the rapid prototype of a system our compiler-generator LISA [18] has been used. The LISA system automatically generates a compiler or an interpreter and other language-based tools—such as language-knowledgeable editor, inspectors, and animators [10]—from an attribute grammar specification. One of LISA's most important feature is that it supports incremental development of specifications, which is especially important in particular tasks of the software development described in this paper.

4 An Example: Video Store

The Video Store (VS) case study is one of the basic examples of the refactoring [7][23]. The case study represents

a prototype program for customer charges at a video store. The program calculates the charge, which depends on how long a movie is rented and on the type of the movie. There are three kinds of movies: regular, children and new releases.

The problem specification: After the analysis of the problem stated above, the discovering of the main functionalities is to be done and present them as use case diagram.

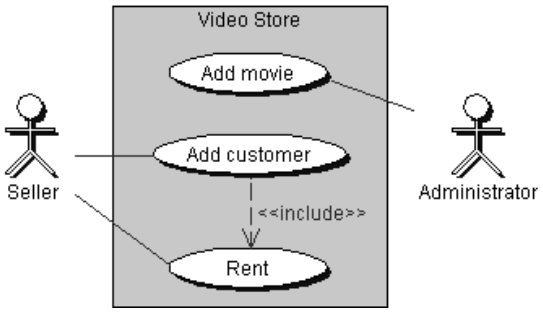


Figure 2: Use case diagram

For the case study of the Video Store we identify three main services represented with use cases *Add movie*, *Add customer* and *Rent* (Fig. 2). To specify their functionalities, the sequence of actions has to be defined. Therefore, scenarios for use cases are written (description follows below).

```
Scenario for Add movie use case:
1. Request for a movie title.
2. Request for a movie type.
3. Insert the movie in movie database.
   Use case end.
ALT 3a: Movie title already contained
   in movie database.
   Inserting skipped. Use case end.

Scenario for Add customer use case:
1. Request for a customer name.
2. Insert the customer in customer
   database. Use case end.
ALT 2a: Customer already contained in the
   customer database. Inserting skipped.
   Use case end.

Scenario for Rent use case:
1. Request the name of the customer.
2. Request the titles of rented movies.
3. Insert the list of rented movies in
   customer's database.
4. Calculate the charge for rental
   service. Use case end.
ALT 1a: Name not in the customer
   database. Insert new customer. Use
   Add customer.
ALT 2a: Movie title unknown.
   Go to step 2.
```

The Conceptual Class Diagram: The use case diagram (Fig. 2) is crucial to find the basic entities and to derive the conceptual class diagram. There are no specific rules to support this derivation, but you can find many guidelines in [11][13].

The structure of the problem domain can be defined in terms of classes and relationships as depicted in the conceptual class diagram in figure 3.

As shown on figure 3, the *VideoStore* is identified as the main concept. The two other important concepts in the management of the *VideoStore* are: *Customer*, and *Movie*. *Movie* associates with class *Price* which describes the type of a movie. Generalization class *Price* is further implemented with classes *New*, *Child*, and *Reg*. The data for each rental are kept in class *Rental*.

The Structure: Remember that, in our approach, a problem concept is denoted by a grammar symbol. The context-free grammar below formalizes the problem syntax in the sense that it specifies the structure of the problem domain, relating the concepts among them. The following context-free grammar is obtained using transformations described in Section 3. To be able to read context-free grammar see the transformation table 1 and table 2.

```
VIDEO_STORE ->  MOVIES CUSTOMERS
MOVIES      ->  MOVIES MOVIE
            |   &
MOVIE       ->  title PRICE
CUSTOMERS   ->  CUSTOMERS CUSTOMER
            |   &
CUSTOMER    ->  name RENTALS
RENTALS     ->  RENTALS RENTAL
            |   &
RENTAL      ->  daysRented MOVIE
PRICE       ->  new | child | reg
```

To follow the transformations from table 1 abstract class Price defines non-terminal PRICE and its subclasses (Fig. 3) define non-terminals in the production

```
PRICE       ->  NEW | CHILD | REG
```

Unfortunately, this subclasses have no terminals and represent the last classes in every traverse through the conceptual class diagram. Described classes are named final classes. Each non-terminal of final class can be replaced with terminal in productions (see the above context-free grammar).

It may happen, that deriving context-free grammar from a conceptual class diagram through transformations in table 1 and table 2 does not show an optimal grammar. Such grammar can have useless non-terminals, which can be reduced. Try to imagine the video store example as stated above, except the rental service changes a bit. Now, the

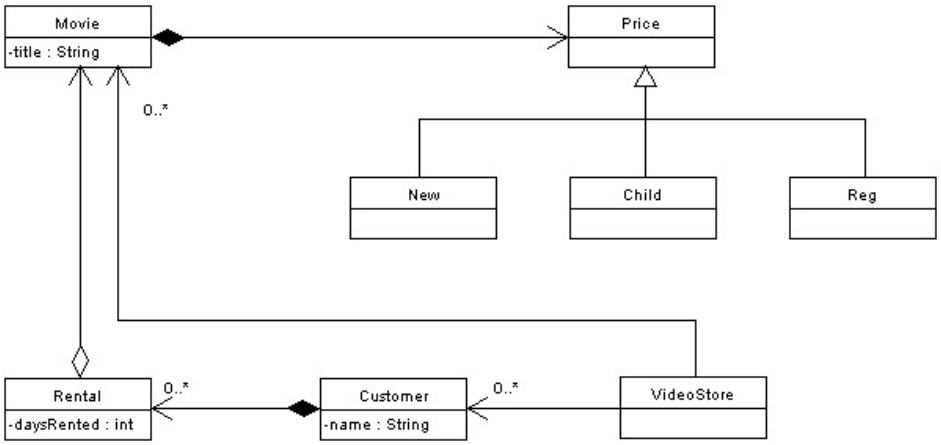


Figure 3: Conceptual Class Diagram for Video Store

rental length for all movies of one customer is the same (in our example the rental length is defined separately for each movie).

```
CUSTOMER    ->  name daysRented RENTALS
RENTALS     ->  RENTALS RENTAL
            |   &
RENTAL      ->  MOVIE
MOVIE       ->  title PRICE
...
```

In the partial context-free grammar we have useless production for non-terminal MOVIE. The production, that have just one non-terminal and no terminal on right side, can be rearranged or even removed (e.g. obtaining just the context-free grammar production `RENTAL -> title PRICE`).

Removing the non-terminal from the context-free grammar brings the question, if class is reasonable in the conceptual class diagram at all. We believe that, if there is no other association with this conceptual class, the class can be removed. Looking from this prospective, building context-free grammar can help in evolving the optimized conceptual class diagram.

Semantics (1. phase): Capturing semantics of the domain is the most demanding part of the approach, therefore an auxiliary (supporting) diagram is proposed.

The semantic constructs in attribute grammar are determinated in Section 3.2. The starting point for finding them the use case diagram is used. Use case diagram is further described with scenarios, which define the interaction between an actor and evolving system. Parsing the scenarios can bring most of the semantic information needed for writing attribute grammar. To support the derivation of semantic information from scenarios, the operational diagram (Fig. 4) has been used.

Each collaboration of an actor and use case diagram is introduced with operational diagram. In the diagram actor shows up twice. First appearance on the left represents an actor before using the system and on the right represents an actor after collaboration with the system. In the middle the name of influenced use case is noted.

Both actors are supported with semantical information, which we get with parsing scenarios of involved use case. Left actor possesses information that the actor needs to collaborate with the system. On the right we write information that actor synthesize in collaboration with the use case.

Information represent semantics of the system and will be further represented as inherited and synthesized attributes in attribute grammar. Still, the open question is to which non-terminals attributes are associated. Explanation follows later in the paper.

The operational diagram brought some important information about attributes and contextual conditions. The next task is to associate attributes from operational diagram to non-terminals in context-free grammar. The table 3 shows the partial attribute mapping to non-terminals. In the first column the attribute names that appeared in operational diagram are written. The next column represent the name of the non-terminal to which attribute should be associated. The column Side and column Terminal are crucial to determinate, whether attribute should be inherited or synthesized. The Side column represents the side where attribute in operational diagram appears. If attribute appears on both sides, attribute should be inherited, as well as synthesized. If it appears on left side of operational diagram and is represented as terminal in context-free grammar, the attribute should be defined as synthesized. If attributes appears on the left side and no terminal can be found in context-free grammar, the attribute should be inherited. The last case is, when an attribute appears only on the right side of the operational diagram. This attribute is synthesized.

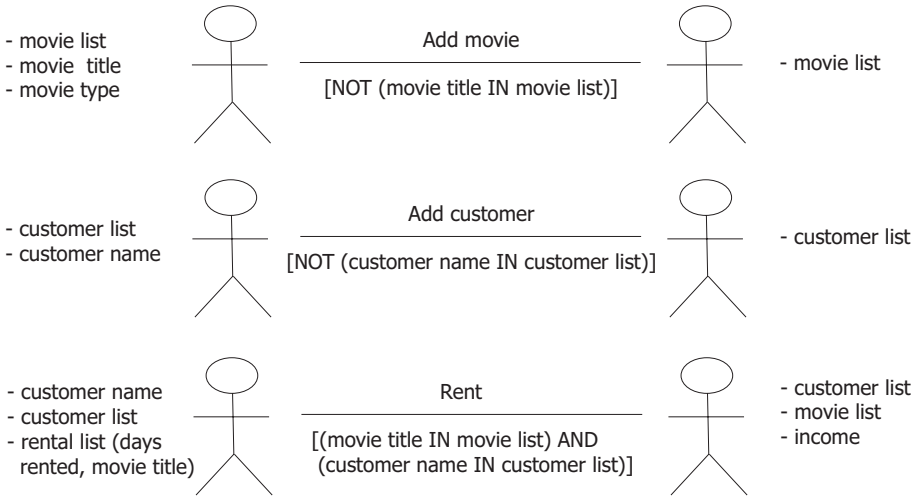


Figure 4: The operational diagram

Operational diagram	Non-terminal	Side	Terminal	I(x)	S(x)
movie list	Movies	left,right	no	inMS	outMS
movie title	Movie	left	yes		title
movie type	Price	left	yes		type
customer list	Customers	left,right	no	inCS	outCS
customer name	Customer	left	yes		name
days rented	Rental	left	yes		daysRented
income	Rental	right	no		income

Table 3: Attributes mapping to nonterminals

Attribute	Starting non-terminal
outCS	Video_Store
outMS	Video_Store
income	Video_Store

Table 4: Attributes in starting non-terminal Video_Store

Attribute	Other nonterminals
inMS	Customers, Customer, Rentals
name	Rentals
inCS	Rentals, Customer
outCS	Customer, Rentals
type	Movie
title	Rental
income	Rentals, Customers, Customer

Table 5: Attributes in other nonterminals

The right side attributes from operational diagrams are important to find information that should be present in starting non-terminal *VideoStore*. In the case study of Video Store, three distinct attributes are defined in operational diagram: *customer list*, *movie list* and *income*. Therefore all three attributes are synthesized in starting non-terminal (Table 4). The table 5 shows attribute carrying between non-terminals in attribute grammar. In the table attributes that must be carried to other non-terminals are showed. To construct this mapping table the domain must be understood well. Each attribute, synthesized or inherited must be considered separately. The main point is to define where should each attribute be carried and with what purpose.

The alternatives in use case scenarios are basics to find the contextual conditions. The contextual conditions are inserted between square brackets (see Fig. 4) where basic boolean operator can be used. Contextual conditions noted on operational diagrams must be also associated to productions of attribute grammar. Their appearance in productions is closely connected to the attributes which define the contextual conditions. Contextual conditions are further implemented with functions which evaluates attributes. The identification of functions are further described in the next semantical phase.

Semantics (2. phase): After detailed semantic description of the problem domain, we can write specifications in attribute grammar. The only semantic part left, is to define function for attribute evaluation. The specifications are broken into separate non-terminal descriptions.

The first production is VIDEO_STORE → MOVIES CUSTOMERS. The non-terminal defines element of entity MOVIES and CUSTOMERS. To keep video store information we define two attributes for each entity. Both attributes are of type TAB, which is a mapping function.

```
TABM = FF(string, (string, int))
TABC = FF(string, (string, int, TABR))

NonTerm VIDEO_STORE:
  Inh: {}
  Syn: {outMS: TABM, outCS: TABC,
        income: int}

  mkVideoStore(VIDEO_STORE ->
               MOVIES CUSTOMERS):
    VIDEO_STORE.outMS = MOVIES.outMS
    MOVIES.inMS = {}
    VIDEO_STORE.outCS = CUSTOMERS.outCS
    CUSTOMERS.inCS = {}
    CUSTOMERS.inMS = MOVIES.outMS
    VIDEO_STORE.income = CUSTOMERS.income
```

For collecting the elements of entity Movie, we use non-terminals MOVIES and MOVIE (see Section 3). The semantic of the non-terminal is described with attributes inMS and outMS, where first attribute inMS is inherited and outMS synthesized. The function insert() adds an element of pair (name, type) to movie table. If the movie is already in the collection, the element is not added in the collection of movies. This is represented with contextual condition (CC).

```
NonTerm MOVIES:
  Inh: {inMS: TABM}
  Syn: {outMS: TABM}

  mkMovies(MOVIES -> MOVIES MOVIE):
    MOVIES/1.inMS = MOVIES/0.inMS
    MOVIES/0.outMS =
      insert(MOVIES/1.outMS,
        new Movie(MOVIE.title,
                  MOVIE.type))
  CC: (NOT(MOVIE.title IN
       MOVIES/1.outMS))
  emptyMovies(MOVIES -> &):
    MOVIES.outMS = MOVIES.inMS
```

Semantic constructs of non-terminal MOVIE are shown below. The symbol MOVIE is semantically described with two attributes that represent basic data of the Movie entity.

```
NonTerm MOVIE:
  Inh: {}
  Syn: {title: String, type: Price}

  getMovie(MOVIE -> title PRICE):
    MOVIE.title = title.lexval
    MOVIE.type = PRICE.type
```

The entity Customer follows the same principle as shown at the non-terminal MOVIES. The multiplicity 0..m brings the use of the non-terminals CUSTOMERS and CUSTOMER.

```
NonTerm CUSTOMERS:
  Inh: {inCS: TABC, inMS: TABM}
  Syn: {outCS: TABC, income: int}

  mkCustomers(CUSTOMERS ->
              CUSTOMERS CUSTOMER):
    CUSTOMERS/1.inCS =
      CUSTOMERS/0.inCS
    CUSTOMERS/0.outCS =
      CUSTOMER.outCS
    CUSTOMER.inMS =
      CUSTOMERS/0.inMS
    CUSTOMERS/1.inMS =
      CUSTOMERS/0.inMS
    CUSTOMER.inCS =
      CUSTOMERS/1.outCS;
    CUSTOMERS/0.income =
      CUSTOMERS/1.income +
      CUSTOMER.income
  CC: (NOT(CUSTOMER.name IN
       CUSTOMERS/1.outCS))

  emptyCustomers(CUSTOMERS -> &):
    CUSTOMERS.outCS = CUSTOMERS.inCS;
    CUSTOMER.income = 0.0
```

Semantics constructs of non-terminal CUSTOMER consist of attributes name (String type), inCS (inherited enumeration of Customers), outCS (synthesized enumeration of Customers) and outMS (synthesized enumeration of Movies).

```
NonTerm CUSTOMER:
  Inh: {inCS: TABC, inMS: TABM}
  Syn: {name: String, outCS: TABC,
        income: int}
  getCustomer(CUSTOMER ->
              name RENTALS):
    CUSTOMER.name = name.lexval
    RENTALS.name = CUSTOMER.name
    CUSTOMER.outCS = RENTALS.outCS
    RENTALS.inCS = insert(
      CUSTOMER.inCS,
      new Customer(CUSTOMER.name))
    RENTALS.inMS = CUSTOMER.inMS
    CUSTOMER.income = RENTALS.income
```

To define rental items, the non-terminal RENTALS holds three distinct inherited attributes: inMS, inCS and name. To keep the final value after mapping rentals to specific customer, the synthesized attribute outCS is used. To support the rental charging service, a synthesized attribute income is applied.

```
TABR = FF(string, (MOVIE, int))

NonTerm RENTALS:
  Inh: {inMS: TABM, inCS: TABC,
        name: String}
  Syn: {outCS: TABC, income: int}

  mkRentals(RENTALS -> RENTALS RENTAL):
    RENTALS/1.inCS = RENTALS/0.inCS
    RENTALS/1.inMS = RENTALS/0.inMS
    RENTALS/1.name = RENTALS/0.name
    RENTALS/0.outCS =
      addRental(RENTALS/1.outCS,
      getCustomer(RENTALS/1.outCS,
        RENTALS/0.name),
      new Rental(getMovie(RENTALS/0.inMS,
        RENTAL.title),
        RENTAL.daysRented))
    RENTALS/0.income = RENTALS/1.income +
      getCharge(getMovie(RENTALS/0.inMS,
        RENTAL.title),
      RENTAL.daysRented)
  CC: ((RENTAL.title IN RENTALS/0.inMS)
       AND (RENTALS.name IN
       RENTALS/0.inCS))

  emptyRentals(RENTALS -> &):
    RENTALS.outCS = RENTALS.inCS
    RENTALS.income = 0.0
```

As shown above, for mapping the rental items to customer, the function addRentals is defined. The mapping process is prevented if rented movie is not present in inherited attribute inMS and also if customer is not present in inherited attribute inCS. This is shown above with contextual condition.

The semantic of non-terminal RENTAL is specified using the values returned by the scanner. Therefore, attributes title (inherited from non-terminal MOVIE) and daysRented are used.

```
NonTerm RENTAL:
  Inh: {}
  Syn: {title: String,
        daysRented: int}

  getRental(RENTAL -> daysRented MOVIE):
    RENTAL.title = MOVIE.title
    RENTAL.daysRented =
      atoi (daysRented.lexval)
```

The non-terminal PRICE represents class Price from the conceptual class diagram. This is an abstract class which defines three subclasses, classes Reg, Child and New (non-terminals REG, CHILD and NEW) in the conceptual class diagram. Because of the final class rule (see Section 4), non-terminals are replaced with terminals.

```
NonTerm PRICE:
  Inh: {}
  Syn: {type: Price}

  getPriceNew(PRICE -> new):
    PRICE.type = new New()

  getPriceReg(PRICE -> reg):
    PRICE.type =  new Reg()

  getPriceChild(PRICE -> child):
    PRICE.type =  new Child()
```

As described in above specifications, attribute evaluation is derived through semantic functions. Functions open the next question. Can this functions help to derive information to obtain methods in class diagram (fig. 5). In that case, the part of prototype could be reused in developing the complete system. This part of our approach is under investigation.

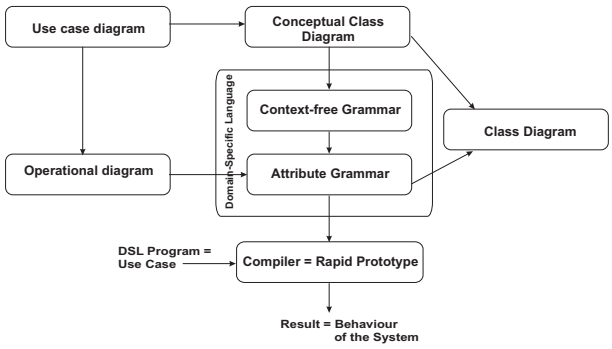


Figure 5: Developing class methods from functions

The rapid prototype: The attribute grammar specified in the previous step is then written using our compiler generator system LISA. The inherent modularity of attribute grammars enables iterative design of prototype. Therefore, more functionalities of a system can be implemented.

A part of these specifications is shown below. Note the straightforward translation from above specifications to LISA. Notice that, the contextual conditions are not shown below. They are implemented with functions which appear in the LISA method part.

```
language VIDEO_STORE {
  lexicon {
    daysRented [0-9]+
    reserved new | reg | child
    name [A-Z][A-Za-z0-9_]*
    title [a-z][a-z0-9_]*
    ignore [\ \0x0A\0x0D\0x09]+
  }
```

```
attributes
  Hashtable *.outMS,*.inMS;
  Hashtable *.outCS,*.inCS;
  Price *.type;
  String *.name;
  String *.title;
  int *.daysRented;
rule Store {
  VIDEO_STORE ::= MOVIES CUSTOMERS
  compute {
    VIDEO_STORE.outMS = MOVIES.outMS;
    MOVIES.inMS = new Hashtable();
    VIDEO_STORE.outCS =
      CUSTOMERS.outCS;
    CUSTOMERS.inCS = new Hashtable();
    CUSTOMERS.inMS = MOVIES.outMS;
    VIDEO_STORE.income =
      CUSTOMERS.income;
  };
}
rule Movies {
  MOVIES ::= MOVIES MOVIE compute {
    MOVIES[1].inMS = MOVIES[0].inMS;
    MOVIES[0].outMS = insert(
      MOVIES[1].outMS,
      new Movie(MOVIE.title,
      MOVIE.type));
  }
  | epsilon compute {
    MOVIES.outMS = MOVIES.inMS;
  };
}
rule Movie {
  MOVIE ::= #title PRICE compute {
    MOVIE.title = #title.value();
    MOVIE.type = PRICE.type;
   };
}
rule Customers {
  CUSTOMERS ::= CUSTOMERS CUSTOMER
  compute {
    CUSTOMERS[1].inCS =
      CUSTOMERS[0].inCS;
    CUSTOMERS[0].outCS =
      CUSTOMER.outCS;
    CUSTOMER.inMS =
      CUSTOMERS[0].inMS;
    CUSTOMERS[1].inMS =
      CUSTOMERS[0].inMS;
    CUSTOMER.inCS =
      CUSTOMERS[1].outCS;
    CUSTOMERS[0].income =
      CUSTOMERS[1].income +
      CUSTOMER.income;
  }
  | epsilon compute {
    CUSTOMERS.outCS = CUSTOMERS.inCS;
    CUSTOMERS.income = 0.0;
  };
}
rule Customer {
  CUSTOMER ::=  #name RENTALS compute {
    CUSTOMER.name = #name.value();
    RENTALS.name = CUSTOMER.name;
    CUSTOMER.outCS = RENTALS.outCS;
    RENTALS.inCS =
      insert(CUSTOMER.inCS,
        new Customer(CUSTOMER.name));
    RENTALS.inMS = CUSTOMER.inMS;
    CUSTOMER.income = RENTALS.income;
  };
}
rule Rentals {
  RENTALS ::= RENTALS  RENTAL compute {
    RENTALS[1].inCS = RENTALS[0].inCS;
    RENTALS[1].inMS = RENTALS[0].inMS;
    RENTALS[1].name = RENTALS[0].name;
    RENTALS[0].outCS = addRental(
      RENTALS[1].outCS, getCustomer(
        RENTALS[1].outCS,
        RENTALS[0].name),
      new Rental( getMovie(
        RENTALS[0].inMS, RENTAL.title),
      RENTAL.daysRented));
    RENTALS[0].income =
      RENTALS[1].income +
      getCharge ( getMovie(
        RENTALS[0].inMS, RENTAL.title),
        RENTAL.daysRented);
  }
  | epsilon compute {
    RENTALS.outCS = RENTALS.inCS;
    RENTALS.income = 0.0;
  };
}
rule Rental {
  RENTAL ::= #daysRented MOVIE compute {
    RENTAL.title = MOVIE.title;
    RENTAL.daysRented = Integer.
      valueOf( #daysRented.value())
      .intValue();
  };
}
rule Price {
  PRICE ::= new  compute {
    PRICE.type = new New();
  }
  | reg compute {
    PRICE.type = new Reg();
  }
  | child compute {
    PRICE.type = new Child();
  };
}
... // method part
} // Language
```

One of the possible scenarios is now described with the following program:

```
jurassic_park child
road_trip reg
the_ring new
Andy 3 jurassic_park child 2 road_trip reg
```

```
Mary 3 the_ring new
```

The meaning of the above program is the following movie table (attribute outMS), customer table (attribute outCS) and money income (attribute income).

```
outMS:{
  jurassic_park={Jurassic_park, child},
  road_trip={road_trip, reg},
  the_ring={the_ring, new}}
outCS:{Mary=(Mary,{the_ring=(
   (the_ring,new),3)},3.5),
   Andy=(Andy,{road_trip=(
   (road_trip,reg),3)},
   jurassic_park=(
   (jurassic_park,child),2)},
   4.5)}
income:8.0
```

Note that for the same scenario the following Java program has to be executed, which is much more verbose and less intuitive for the end-user:

```
public static void main(String[] args){
    double income = 0.0;
    Movie m1 = new Movie(
      "jurassic_park", Movie.CHILDRENS);
    Movie m2 = new Movie(
      "road_trip", Movie.REGULAR);
    Movie m3 = new Movie(
      "the_ring", Movie.NEW_RELEASE);
    Customer c1 = new Customer("Andy");
    Customer c2 = new Customer("Mary");
    Rental r1 = new Rental(m1, 3);
    Rental r2 = new Rental(m2, 2);
    Rental r3 = new Rental(m3, 3);
    c1.addRental(r1);
    c1.addRental(r2);
    c2.addRental(r3);
    income += c1.evaluateCharge();
    income += c2.evaluateCharge();
  }
```

5 Conclusion

In the paper our approach to developing a formal specification for a given problem using a complementary syntax/semantics approach is described. Not least, our approach can be also seen as a formal approach to program construction with all benefits of formal approaches. The proposed approach can be also applied if the user's requirements are not well defined; more symbols or attributes (attribute rules or constraints) can be easily added in a later phase (when the user comes up with new requirements/functionalities), and a new prototype will be immediately generated. The essence of our approach is the development of a domain-specific language that describes the user interaction with a system or the functionality of a system. While executing programs written in a specified domain-specific language the functionality of a system and user's requirements can be validated. The starting point of our approach is the identification of concepts in the problem domain. Here, well known techniques from object-oriented design, such as use case diagrams and conceptual class diagrams, are used. However, our approach can be used also with data-flow diagrams and entity-relation diagrams. In that case just new transformation rules have to be defined, similar to those presented in table 1 and table 2.

In our future work we would like to investigate the possibility to obtain a domain-specific language only from a use case diagram which describes the functionality of a system. It is well known that use case diagrams and class diagrams represent different views on a given problem and that there is no direct transformation between those two techniques. Has such context-free grammar some valuable information for constructing a conceptual class diagram? Is it possible that a context-free grammar of a domain-specific language, derived from use case diagram, describes the class diagram for a given problem? Such findings might have some impact on current object-oriented design. Hence, our future work is to explore this connection.

References

[1] S. Adolph. *Patterns for Effective Use Cases*. Addison-Wesley, 2002.

[2] A. Arsanjani. Grammar-oriented object design: Creating adaptive collaborations and dynamic configurations with self-describing components and services. In *Proceedings of TOOLS 2001*, volume 65. IEEE Computer Society Press, 2001.

[3] G. Booch. *Object-Oriented Analysis and Design with Applications*. Benjamin/Cummings, 1994.

[4] B. Bryant and B. Lee. Two-level grammar as an object-oriented requirements specification language. In *IEEE CD ROM Proceedings of 35th Hawaii International Conference on System Sciences*, 2002.

[5] A. Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2001.

[6] P. Deransart, M. Jourdan, and B. Lorho. *Attribute Grammars: Definitions, Systems and Bibliography*, volume 323. Lecture Notes in Computer Science, Springer-Verlag, 1988.

[7] M. Fowler. *UML Distilled. Applying the Standard Object Modeling Language*. Addison-Wesley Longman, 1997.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[9] J. Heering and P. Klint. Semantics of programming languages: A tool-oriented approach. *ACM Sigplan Notices*, 35(3):39–48, March 2000.

[10] P. Henriques, M. V. Pereira, M. Mernik, M. Lenič, E. Avdičaušević, and V. Žumer. Automatic generation of language-based tools. In Mark van den Brand and Ralf Laemmel, editors, *Electronic Notes in Theoretical Computer Science*, volume 65. Elsevier Science Publishers, 2002.

[11] I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software DevelopmentProcess*. Addison-Wesley, 1999.

[12] D. Knuth. Semantics of context-free languages. *Math. Syst. Theory*, 2(2):127–145, 1968.

[13] C. Larman. *Applying UML and Patterns-an Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice-Hall, 2nd edition, 2002.

[14] K. Levi and A. Arsanjani. A goal-driven approach to enterprise component identification and specification. *Communications of the ACM*, 45(10):45–52, October 2002.

[15] K.J. Lieberherr. *Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns*. PWS Publishing Company, Boston, 1996.

[16] M. Mernik and D. Parigot (Eds.). Attribute grammars and their applications. *Informatica*, 24(3), September 2000.

[17] M. Mernik, J. Heering, and T. Sloane. When and how to develop domain-specific languages. Technical Report SEN-E0309, University of Maribor, CWI Amsterdam, and Macquarie University, 2003.

[18] M. Mernik, M. Lenič, E. Avdičaušević, and V. Žumer. LISA: An Interactive Environment for Programming Language Development. In Nigel Horspool, editor, *11th International Conference on Compiler Construction*, volume 2304, pages 1–4. Lecture Notes in Computer Science, Springer-Verlag, 2002.

[19] B. Roussev. Generating ocl specifications and class diagrams from use cases: A newtonian approach. In *IEEE CD ROM Proceedings of 36th Hawaii International Conference on System Sciences*, 2003.

[20] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

[21] I. Sommerville. *Software Engineering*. Addison-Wesley, 5th edition, 1996.

[22] A. van Deursen, P. Klint, and J. Visser. Domain-Specific Languages: An Annotated Bibliography. *ACM Sigplan Notices*, 35(6):26–36, 2000.

[23] A. van Deursen and L. Moonen. The video store revisited thoughts on refactoring and testing. In *Proceedings of the 3rd International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2002), University of Cagliary, 2002*, pages 71–76, 2002.

[24] D. Wile. Supporting the DSL Spectrum. *Journal of Computing and Information Technology, Special Issue on Domain-Specific Languages, R. Laemmel and M. Mernik, eds.*, 9(4):263–287, Dec 2001.

Information Systems Integration Process Model

Matjaž B. Jurič, Marko Tekavc and Marjan Heričko
University of Maribor, Faculty of Electrical Engineering and Computer Science
Institute of Informatics, Smetanova 17, SI-2000 Maribor, Slovenia
matjaz.juric@uni-mb.si, http://lisa.uni-mb.si/~juric/

Keywords: integration, process, EAI, B2B, information systems

Received: June 23, 2004

Integration of information systems is a complex field where major challenges are semantic, process and technology related. Integration must be performed using methods, disciplines and activities that enable it to be effective in terms of costs and time – thus it should be supported by a well defined integration process. This article presents an information systems integration process model proposal with the goal to guarantee the quality of the integrated solution. The article focuses particularly on the integration specific disciplines: analysis of existing applications and integration design.

Povzetek: članek opisuje integracijo kompleksnih informacijskih sistemov.

1 Introduction

The growing need for the easy accessibility of information presents new challenges for information system development. This need is unlikely to be fulfilled by the separate "stand-alone" applications. Applications need to be integrated to make the information they contain available and accessible [17].

Integration is not an easy task; indeed it has become one of the most difficult problems facing enterprise application development in the last few years. The major challenges are semantic, process and technology related [16]. Information system integration or Enterprise Application Integration (EAI) as seen from the business perspective, is the competitive advantage an enterprise gets when all applications are integrated into a unified information system, capable of sharing information and supporting business workflows. From the technical perspective, EAI refers to the process of integrating different applications and data, to enable sharing of data and integration of business processes among applications without having to modify these existing applications. EAI must be performed using methods, disciplines and activities that enable it to be effective in terms of costs and time. EAI should be project oriented and should be supported by a well defined integration process.

The review of related work shows that not much has been done in the field of integration processes. In [1] the authors address the problems of EAI with ERP systems. In [2] the author addresses the problem of using middleware in integration projects. In [3] authors introduce agentified enterprise components to improve integration and cooperation. In [4] EAI is addressed from the workflow perspective. In [5] authors explain the integration of heterogeneous e-commerce applications and focus on technology questions. In [6] the use of web services for integration is discussed. In [7] the authors present a notation for modeling EAI architectures. In [8] the authors give an overview of architectures and technologies used for EAI. In [9] the component approach to EAI is presented. In [10] an XML based framework for integration is presented and in [11] a web based infrastructure is presented. None of these articles addresses the integration process. Some directives related to agile approach to integration can be found in [12], [13], [14], and [15]. They do not present the whole process however.

In this article we present the integration process model proposal which is based on the EMRIS methodology [18]. The integration process as presented in this article defines the sequence of activities to be done in a disciplined manner in order to successfully develop an integrated information system. The goal of the integration process is to guarantee the quality of the integrated solution that will satisfy the customer, will be completed on schedule, and will be within the allocated financial resources. The integration process is tightly connected to the software development process, with which it shares several disciplines. It is based on real-world experience and has been successfully used in several large-scale integration projects.

The article is organized as follows: section 2 gives an overview of the integration process, section 3 describes the analysis of existing applications, section 4 describes the integration design and section 5 gives the concluding remarks.

2 Integration Process Outline

The presented integration process is based on the following integration practices: iterative development, incremental development, prototyping, reuse, design simplification, test automation, and customer involvement.

Integration process consists of disciplines which are performed in several iterations. We focus on technical

disciplines only: Requirements gathering, Analysis of existing applications, Selection of the integration infrastructure, Problem domain analysis, Integration design, Implementation, Testing, and Deployment. Figure 1 presents the integration process outline.

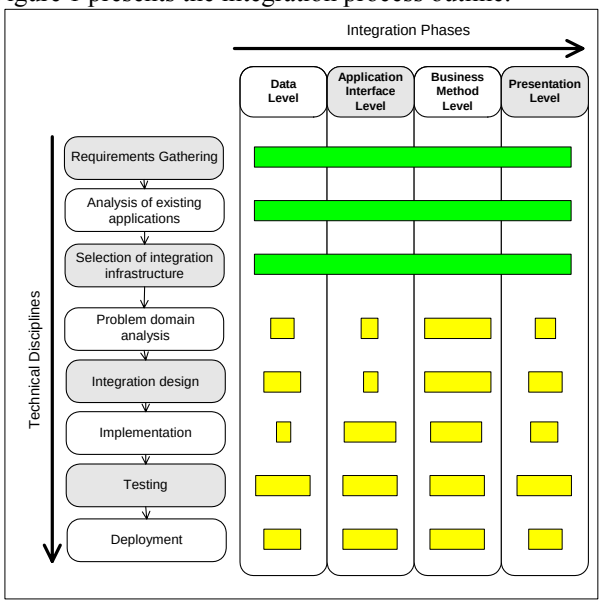


Figure 1: Integration process outline

The disciplines that are common to all phases are shown with a single box. The other disciplines are shown with separate boxes. The size of the boxes represents the approximate duration of each discipline in a certain integration phase. For example, problem domain analysis and the integration design disciplines require the most effort in business-method-level integration phase, where we have to define the global design model of the integrated information system. The least implementation effort is usually in data-level integration phase because it rarely requires changes to existing applications.

Integration is usually achieved in four phases:

- Data-level integration phase
- Application interface level integration phase
- Business-method-level integration phase
- Presentation integration phase

Each integration phase requires a lot of effort and time. Therefore, it has to be considered as a sub-project. To support iterative incremental development, each integration phase is usually broken into several iterations. Iterations enable a finer-grained control over the integration phase. Usually there are at least four iterations for each integration phase. These main iterations can however have further sub-iterations, depending on the project size and the schedule. The four main iterations for each integration phase are: inception, elaboration, construction, and transition.

Inception defines the business perspective of integration and estimates its size. We have to specify the requirements, identify all entities our system will cooperate with, and define how it will cooperate. We also have to define the milestones and the criteria for assessing the success of the integration, analyze the risks, and select the resources.

In elaboration we analyze the existing applications and get a clear understanding of what applications we have to deal with. We also analyze the problem domain, define the project plan, the basic architecture, and solve the most hazardous parts of the integration project. We also specify the requirements for the integrated information system. As we have to make architectural choices, it is very useful for us to build architectural prototypes to validate the chosen architecture. At the end of elaboration we evaluate the goals, the size of the project, and the architecture decisions, and we should once again assess the risks.

The goal of the construction is to actually implement the integration that will result in completing a certain integration phase. This part is the most time-intensive and will have the largest number of iterations. When constructing the integrated system, we obtain a clear understanding of the integrated information system that we are building. We also need to know how the existing applications map to the newly defined integration architecture and which functionality we will be able to reuse. Then we build the design model, write the implementation code, and perform testing and verification. At the end of the construction we verify whether the developed integration satisfies the requirements.

In transition we deploy the developed integration components into the production environment. Upon deployment there are often additional problems and complications that arise, which we have to solve. The transition usually begins when we have a beta version of the integration components ready. Transition finishes when we are satisfied with the functionality of a certain integration phase. After transition we usually proceed to the next integration phase (from data-level to application interface level for example).

The integration process differs from the usual software development process in that it has to take existing applications into account. Analysis of existing applications has to be made and the integration design discipline has to be adapted. In this article we will focus on both mentioned disciplines:

- Analysis of Existing Applications
- Integration Design

Selection of integration infrastructure has been addressed in [17], the integration assessment in [19]. Other disciplines, such as requirements gathering, problem domain analysis, implementation, testing, and deployment do not differ considerably from general software development disciplines, as described in [18].

3 Analysis of Existing Applications

Before we start analyzing existing applications we have to select the applications to be integrated. This should include all the major primary "backbone" applications. But we should also not forget subsidiary applications, often self-made or locally developed solutions that users use on a daily basis.

In the analysis of existing applications, we identify and specify the functionality of each application that will

be included in the integrated information system. We identify the data models, perform the functional analysis, identify the architecture of existing applications, and identify ways to access this functionality.

We also need to identify redundancy and other semantic problems, where the functionality of several applications may be overlapping. Usually in this discipline we will look at the applications in two ways. First, we'll study the data that is stored in applications. Second, we'll identify the functionality that is provided and the ways in which to access it – we will extract the business rules that are embedded in the existing applications. The outcome is the data- and functionality-level analysis models.

We perform the analysis of existing applications in a controlled and disciplined manner and follow the following activities in order to analyze each existing application selected for integration: functional analysis, technical analysis, analysis of functional overlapping, analysis of existing integration. Figure 2 shows the main activities and their refinements.

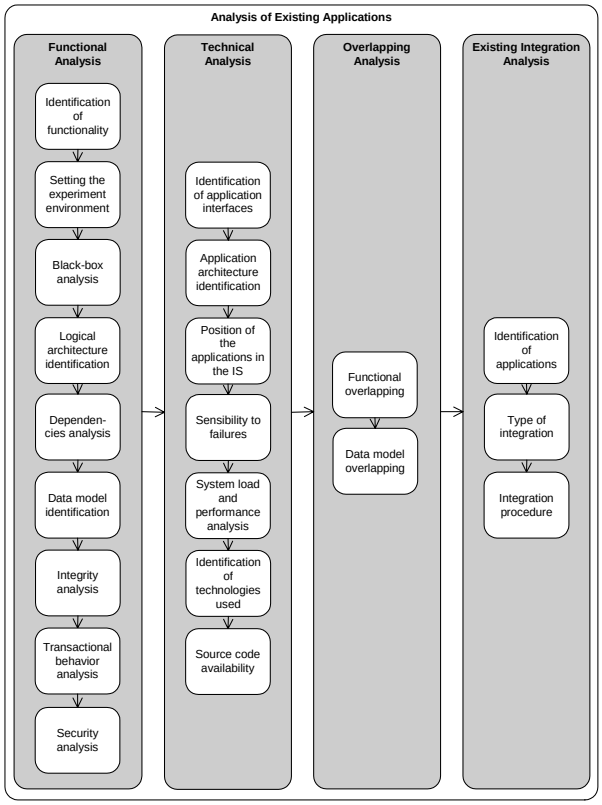


Figure 2: Analysis of existing applications discipline

3.1 Functional Analysis

3.1.1 Identification of Functionality

In order to reuse as much functionality as possible, it is important to identify all the functions the existing applications possess. In addition, we also have to identify how often a function is used. This is important because the existing application could even have some functions that have never been used. There may be no guarantee that these functions actually work correctly. To avoid unpleasant surprises, it is recommended that we consider only the functions that are actually used and which we know work correctly.

The documentation that will be interesting for the identification of functionality includes requirements specifications, analysis and design documentation, testing documentation, and user documentation. For commercial applications, we'll probably have up-to-date documentation, or at least the user documentation (user manual) that will explain how to use the application. For applications developed in-house we probably won't have up-to-date documentation, but we may be able to find the requirements specification. The requirements specification is often the basis for getting a software development project approved. This can be a good start, but we still have to check how each function is implemented. If the application is not too old, we may be in luck and the original developers may still be around. They will have the best understanding of the application and it is well worth talking to them about the functions that their application implements.

If we cannot talk with the original developers we have to talk with system administrators and users. System administrators will have an overview of how often the application has been used and where the problems have been. Users will be familiar with the functions. Although this is not the time to start developing code, we may take this opportunity to check whether the source code actually exists, and if so is it in-synch with the executable versions? Many existing applications do not have adequate documentation. For some, even the source code does not exist. Even if the source code does exist we have to check if it is in-synch with the executable versions.

For outsourced applications we are faced with similar problems as with in-house developments. If the outsourced projects have been managed efficiently then there should be documentation available that will be comprehensive and up-to-date. However, many projects have not been well managed and we will not have the documentation. On the plus side, for almost every outsourced project the requirements specification should exist. It usually forms the basis for the contract and for assessing the value of the software development project. Software development companies are also more aware of the importance of documentation.

However, for an outsourced application it might be even more difficult to get in touch with the original developers. They are probably not employed by the same company. Even if they are, they are not likely to want to talk with us for free. This problem is exacerbated when we find out that the consulting company does not exist anymore and it has delivered the executable application without source code.

3.1.2 Setting Experimental Environment

After we have prepared the list of functions with their frequency of use, we have to check each function to get an idea how it works. We do this in an experimental environment that we have to set up. This will basically be

a copy of the production environment, which will enable us to experiment with existing applications without disturbing their everyday operation. Setting the experiment environment is important not only for the analysis phase, but is very useful later when we apply modifications to the existing applications. Without an experimental environment it would be absolutely impossible to safely test and validate the integration solutions.

Setting it up can vary in complexity. It is easy in cases where we have the necessary hardware, and where we can simply copy the applications, with or without the persistent data. The more complicated the application architecture, the more work we have to set up the environment. Becoming comfortable in the environment of the existing applications is crucial to achieving integration.

This will be the most difficult for legacy applications. For them, there will be the problem of obtaining the necessary hardware, and we probably won't be familiar with the environment and the tools, which may present the biggest obstacle. A big problem can be setting up experimental databases. Again it depends on the architecture of the application: if it uses some standard way to access the database it will be easier.

Only with commercial applications we expect to have some form of installation procedure. However, we have to be sure that the actual product is identical to the application that is used in production. Otherwise it is a better idea to use the production version.

For applications where performance workload is not limited, we are able to use the same hardware for the production and experimental configurations. If we make this decision, we have to be very careful not to interfere with the production data. This approach will not be applicable if the application has a high workload and/or is mission critical. In this case we have to set up a fully isolated experimental environment.

If we are unable to set up an experimental environment for an application we want to integrate, we have to be very careful with the tests that we do. We have to consider what time to perform the tests, for example, when the application is not in use (during nights, weekends, or holidays). This will influence our flexibility considerably.

3.1.3 Black-box Analysis

We then have to check each function that we have listed in the functional specification. Note that we're not only talking about the functions accessible from the user interface, we have to include all functions, even those that the application provides through APIs.

We call this activity black-box analysis because we don't care about how the function is performed by the existing application. We are interested solely in the output that we get and what input parameters we have to provide to get the desired output.

When specifying the input and output behavior we should pay particular attention to the boundary conditions. This means we should consider the allowed intervals for input parameters. We should specify this in the form of preconditions for the input parameters. This will become important later when we reuse the functionality.

To describe the functions of existing application we can use a textual form, where we produce a table and description of the functions. The proposed table should include the following columns:

- Function – name of the function that the application provides.
- Description – description of the functionality.
- Access via user interface or via API – we should identify how we can access the functionality.
- Frequency of use – we should identify if the function is used at all and, if so, how frequently it is used. If possible we should use an objective metric, for example number of times per week.
- Required inputs – we should clearly identify the input parameters and their allowed ranges.
- Outputs – we should identify the outputs that we get.

3.1.4 Logical Architecture Identification and Dependencies Analysis

After we have identified the functionality of the existing applications, we have to recognize its internal structure. Here we first have to identify if the application is monolithic, client/server, or multi-tier. Then we try to categorize how it is constituted – if there are several modules or components, where the business logic is, etc.

After we have identified the logical architecture we have to classify the dependencies between the applications. Here we should identify all the dependencies. Two applications can have logical dependencies that can be implemented either automatically or manually.

If implemented automatically then there is a sort of interoperability between the applications – these applications share data or functionality. Often, particularly with legacy systems, such connections are implemented through data exchange, very often via shared files or tables. This will be important later when we come to identify the existing integration between applications. Then we will consider how the integration is implemented from a technical perspective.

More frequently, we will see dependencies that are carried out manually. This means that the users will have to re-enter the same data, leading to possible inconsistencies. An application can provide a summary of some data it processes that the users then enter into some other application. There is obviously a dependency between them that we should identify and show on a diagram. If possible, we can also document these dependencies. This information will be useful later in the analysis.

3.1.5 Data Model Identification

Another very important activity in functional analysis is the identification of the data models used by each application. This is important because we have to

understand how data is stored. We have to analyze the persistence storage of each application. We will be faced with one or more of the following types of databases: relational, object-oriented, universal, multidimensional, hierarchical and network, other formats, such as flat files.

We have to construct the database model for each existing application. This will be the basis for data-level integration. Often it is possible to generate database models automatically with the tools provided by the database. The majority of relational databases, for example, have tools to generate entity-relational (ER) schemas out of existing databases. This is usually better that depending on possibly out-of date documentation.

3.1.6 Integrity Analysis

Here we identify how the integrity of databases is achieved and which party is responsible for it. Most likely each application will be responsible for assuring the integrity of their own databases. In this activity we should identify the integrity rules for each database that the system uses. Identifying the integrity rules will be particularly important for data-level integration when we exchange data between applications based on direct database transfers. Since we will most likely omit the business rules at this stage, we have to be aware what the integrity rules are.

The integrity rules are sometimes described in the documentation. Sometimes they are incorporated within the database, if the database allows this. More often these rules are coded within existing applications. Database administrators can be very helpful with the identification of integrity rules.

The problem with the identification of these rules is that it is very difficult to be sure that we have identified all of them. Not identifying them on the other hand can lead to breaking the integrity of databases. Identifying this problem is a difficult task, and tracking down failures to database integrity problems is very time consuming.

3.1.7 Transactional Behavior Analysis

Transactions play an important role in all non-trivial applications. Their management is known as transactional processing. Transaction monitors can be a DBMS or some dedicated middleware. Transactions can work with a single resource – these are the simplest and most commonly used. However, in large systems the transaction might need to be invoked over several systems. This is when distributed transactions come into play. A distributed transaction spans more than one resource. Their context can be propagated or shared by more than one component; they require the cooperation of several different transaction monitors.

Our goal will be to identify the transactional model (flat, nested, chained or saga) and become familiar with how it works together with the existing application. We have to familiarize ourselves with transactional properties of the existing application, identify how the existing application uses transactions, and how critical the failures are.

3.1.8 Security Analysis

In security analysis we have to examine the way that security is utilized by the existing applications. Generally we need to answer: Does the application implement security? If yes, how is the security implemented? If no, should we add security now? There are four important security mechanisms found in existing applications. Authentication is the process of verifying that a client is who they claim to be. It can be performed on the client before it interacts with the server. It can also be performed on the server.

Authorization checks whether the client application is allowed to perform a certain operation. Authorization can be defined programmatically or declaratively, depending on the implementation. Typically it is defined in terms of security roles and Access Control Lists (ACLs). Extracting info on how authorization is performed from existing applications can be complicated because the logic may be in the application code.

Communication channel security – newer applications will typically use Secure Socket Layer (SSL) and Transport Layer Security (TLS), but this can differ significantly with older legacy systems.

Auditing let us see an exact history of operations performed on the system and is useful for analysis of past events.

The fact is that a lot of existing applications do not have much security implemented. Therefore attention will have to be paid to how to introduce security to existing applications.

3.2 Technical Analysis

3.2.1 Identification of Interfaces

In this activity we focus on the application interfaces. Our goal will be to specify the interfaces that an existing application provides to other applications. First of all, we have to identify how many interfaces there are and which operations they provide. Then we have to identify which technology is used to access them.

To identify the number of application interfaces we will have to go through the documentation, talk with the developers, and even analyze the source code. We might also consider using tools for analyzing existing applications. Such tools sometimes can identify application interfaces even if no source code is available. We should mention that we could consider every form of communication between two applications as an application interface. For now it's not important if those interfaces are implemented in a proprietary technology, if they are procedural or functional, even on protocol level.

We specify the interfaces on the UML component diagrams using the interface stereotype. We also identify the operations of each interface and show their signatures. This means that we have to identify the names and the syntax of operations, the necessary parameters and the return value.

Sometimes we have a situation in which the applications are tightly coupled, so there will have to be

some preconditions fulfilled before an operation can be called or invoked. We need to identify these preconditions (and maybe post-conditions). We also try to identify if there are some restrictions in the order in which the operations have to be invoked. Another important thing is to recognize the way that the application signals errors or other exceptional conditions.

Identifying the interfaces is very important, particularly for application-to-application integration, but sometimes also for data-level integration. Accessing data through operations is better than going directly to the database because we avoid circumventing the business logic. This makes it easier for us to maintain database integrity.

3.2.2 Architecture Identification

Having identified the logical architecture and the interfaces, we should now consider the physical architecture. We need to become familiar with the environment in which the production application is deployed, so we identify the computers on which the application parts are deployed and the type of connection between them. This step should be done for each application separately, although applications will frequently share resources.

To represent the architecture we can use UML deployment diagrams. They show the runtime configuration of hardware devices and the software components that execute on them. Nodes contain component instances, which show that the instances execute on a certain node. Typically there will be several component instances on a single node; however this depends on the granularity of the application. Monolithic and client/server applications will be typically represented by a few components only.

It is also very useful to show the dependencies between the component instances using a dependency relationship. If the components provide interfaces that their communication relies on, then we should show the dependencies using the interfaces that we have already identified. For example, an existing application can provide a custom API for communication with clients, and clients can use a remote procedure call or message-oriented middleware to call the procedures and functions in the API. This can be seen as an interface although it is not an interface in the sense of component/OO-based development. If there are no interfaces that we can identify, then we should just show the dependencies between the components. Sometimes we can specify the communication protocol for each dependency too.

After we have identified the architecture of each application separately we should build the diagram of the whole existing information system. This basically means that we gather together the deployment diagrams that we drew in the previous step. We also need to identify which resources the application share and denote the dependencies (already identified previously) on this diagram.

We should extend this diagram with the other existing applications that are present in the current information system, but have not been selected for integration. We should mark them clearly with the <<external>> stereotype, and note whether there are some dependencies between the external applications (those not selected for integration) and the applications that we are integrating.

Sensibility to failures analysis is the next step in the technical analysis of existing applications. Here we have to identify how critical each application is for the company. We have to see if the company has alternative scenarios regarding what to do if an application fails. If it does not (and most do not have such scenarios), we must develop them. Note that when altering an existing application we will considerably increase the risk of failing, so we have to take every measure possible to minimize the risk. This includes efficient backup systems, which include application data as well as the executable application files.

3.2.3 Performance Analysis

Here we should clarify what the performance considerations of applications are. In the requirements gathering phase we should have already identified the performance expectations for the integrated system. Here we have to see how the existing applications perform. When integrating applications, one of the goals is to provide instant access to information. The technical implications are that after integration there will be a larger number of clients that will simultaneously use the application. Sometimes, for example when making applications accessible online, this number can be considerably higher.

It would be wrong not to consider the performance limitations now. We will look at the system from two perspectives: the client load, that is, the number of concurrent clients, and the data load, that is, the quantity of persistent data.

To identify the client load we should look at the predicted average and maximum number of concurrent clients; the response time by average and maximum number of clients; the highest acceptable response time; how much we increase the number of clients to fulfill the response time limit in the current configuration; the possibilities there are to increase scalability (hardware and software solutions).

If we identify that the application currently offers acceptable response times (and it should, because this is a production application, although in real-world it often does not), we will try to identify how much potential there exists in the application for raising the number of simultaneous clients. From this, we will try to infer the highest possible number of concurrent clients that the existing system can support in its current configuration.

To identify the data load we will first assess the current persistent data size. Then we try to identify if the integration will increase the data size. The reasons can be different. For example, it is possible that pre-integration the data between applications is transferred only once per month and only summary values are recorded. Upon integration we may require this transfer several times per

day or even instantly. This will also mean that the integrated system will record each transaction separately, thus increasing the persistent data dramatically.

3.2.4 Identification of Technologies

In this step we have to familiarize ourselves with the technology used in each existing application. If this is a commercial application we have to check the exact version that is being used. If it is a custom-developed application we have the following points to check: programming language, compiler, IDE, linker, operating system version, DBMS versions, middleware, and all other related software. We also have to look if those versions of software still exist, and if not, how we can obtain them. This will be important for making decisions on rebuilding the system using the source code.

After we have defined the technology we have to see if the source code is available for the existing application. There are a large number of systems (particularly legacy) where source code is not available. Source code will also probably not be available for commercial applications.

For custom-built applications, we will most likely have access to the source code, unless they are old or the source code has been lost, be it accidentally or intentionally. But even if we find the source code we have to check that we have all the necessary tools to rebuild the application and the source code version corresponds to the actual version used in production. Often it happens that a single missing library or configuration file prevents us from rebuilding the application.

To check whether the production version is identical to the source code we can use a simple procedure. We build the application from the source code and compare it to the production version using a file compare utility. We have to be sure that we compare the executable files only, without any data. If this simple procedure does not work then we will have to compare applications, which can be very difficult for small changes.

3.3 Overlapping Analysis

After we have analyzed the existing applications from functional and technical perspectives, we are familiar enough with them to perform an overlapping analysis. The objective here is to identify which parts of the applications overlap – which functionality and data is redundant. We also select which application is responsible for which overlapping functionality. Overlapping analysis consists of two steps: functional overlapping, and data model overlapping.

3.3.1 Functional Overlapping

As existing applications are not usually integrated, an application can often contain certain functionality that has already been implemented by some other application. This is essentially due to a lack of architecting.

So, we are often faced with two or more applications that implement the same functionality. Often one application implements it in the detail, while another implements only the parts that they need. Typically these applications will introduce a slightly modified view of the functionality, which will complicate the situation even more.

We would like to identify which functionality is overlapping in the applications that we have selected for integration. Now we identify which functions of which existing application we will use later, when we reuse some existing functionality for the integrated information system.

To identify the overlapping functions it is a good start to have a look at the dependency analysis that we performed as part of the functional analysis. We should look at the dependencies; particularly those that are implemented manually are suspicious. Implementing a dependency manually means that the user has to re-enter some data that has been processed from one application into another. This may mean that the applications had to overlap a part of their functionality. We also have to check the dependencies that are implemented automatically. If there is only data exchange between applications it can still mean that the functionality is overlapping.

To describe the functionalities that are overlapping we first have to identify the function, then all the applications where the function is implemented and finally select the application that will be responsible for that function (the application that we will use when reusing this function for integration).

3.3.2 Data Model Overlapping

After we have analyzed the functions, we also have to identify the data that might be overlapping in the databases of different applications. Dependency and functional overlapping analysis can be useful here. Functional overlapping almost always means that there is data overlapping under it. But note that there might be data overlapping somewhere else, too.

To identify it, we should again focus on identified dependencies between applications and evaluate first those implemented manually and then those implemented automatically. For data model overlapping analysis, it is very helpful if we have the schemas of all the databases. Then we can identify the data that is overlapping. Similarly, as in functional overlapping, we should select the databases that will be responsible for certain data. These databases will then be used in the integrated information system.

If we are lucky we will only have to deal with one database model, probably relational. Then we have to identify the entities that are overlapping. If we build the data dictionary is very useful to supplement the information that each entity name represents. This point it is also a good opportunity to resolve the name conflicts and to explain the cryptic names for entities and attributes. The most difficult task will however be to resolve semantic issues.

3.4 Existing Integration Analysis

The last activity is to identify any existing integration solutions. It is very likely that we will be faced with some form of already implemented integration. The most common ways are data exchange using shared databases or flat files, or the use of message-oriented middleware to enable point-to-point communication between applications. We have to be aware of existing solutions when planning our integration, although it is often simpler if we don't have any integration at all and can start from scratch.

First we identify all the applications that each application is integrated with. As we have already done the dependency analysis this will not be very difficult. We pay attention to all automatic dependencies, and focus on some specific details that we need to identify: type of integration, exact procedure of how integration is implemented and performed.

The type of integration identification is the second step. We will identify what integration level the existing integrated applications use.

4 Integration Design

In this discipline we focus on the global architectural design model, where we represent the integrated information system as a set of components (identified in the problem domain analysis discipline) that have well defined interfaces through which they communicate. Instead of focusing on how to implement each component from scratch, we focus on how to reuse existing applications to provide implementations for the components.

We approach architectural design from a high-level perspective. Due to the size and complexity of the problem domain, it is practically impossible to design the integration architecture down to the finest detail. This would also be unreasonable because a lot of functionality is implemented by existing applications. Accordingly, we approach the architectural design in a more high-level way, where we define the global architecture in the sense of components and their interfaces. This is somewhat analogous to the planning of a city's architecture compared to designing a house.

Several key activities of this discipline characterize the architectural integration design process. Firstly, we cope with the global situation, and then we focus on information system-specific-functions. Here we start to solve the use cases that influence the architectural decisions and, as a result, we produce a set of subsystems. Each of the subsystems realizes a use case. After iterating though the subsystems we start building the global architecture step-by-step and finally define a stable architecture.

The main activities of the integration design discipline can be organized into three groups as shown in Figure 3.

The integration design discipline is a highly important discipline. Getting the integration design wrong will result in the failure of the whole integration project. Of course the quality of the results in the design discipline is dependent on the quality of the inputs from previous disciplines. Still we should be aware of the importance of this discipline. The risk of mistakes can be greatly reduced with iterative and incremental development.

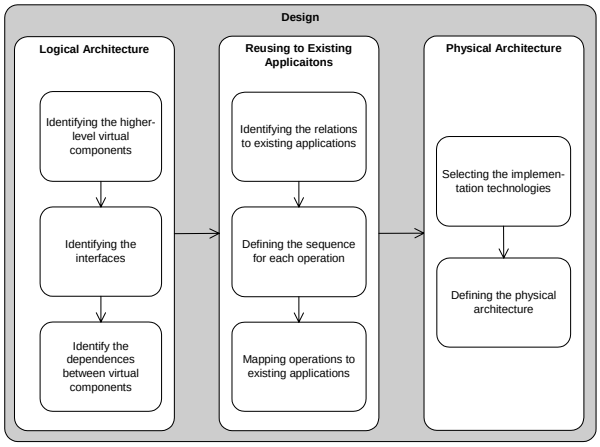


Figure 3: Integration design discipline

4.1 Logical Architecture

Identifying the higher-level virtual components [20] is the first activity in the integration design discipline. We need to identify the higher-level virtual components that constitute the integrated system. The problem is that although this task sounds easy, in reality it is not.

Selecting the correct higher-level virtual components will have a long-lasting influence on the information system as a whole. The selection also determines how suitable the integration architecture is to re-engineering existing applications and replacing them with newly developed solutions.

To identify the high-level virtual components, we focus on the analysis model class diagram. The analysis-level entity and control components that we identified will map to virtual components on the business logic tier, so we will focus on them. The analysis-level boundary components represent user interface constructs. These will be realized in the client and web component tiers.

To identify the virtual components we go through the control and entity components from the problem domain analysis discipline. We try to group them into virtual components based on their functionality. Components encapsulate their internal implementation and represent their functionality through the interface. To identify the higher-level virtual components we can follow these guidelines:

- Start with the analysis class diagram.
- Gather the analysis components that are logically connected because they implement a part of a larger functionality.
- Try to make the virtual components as independent of other components as possible.
- Often we will have to add other specific components that will implement non-functional requirements, for example, or model some implementation-related concepts.

After we have identified the higher-level virtual components, we define the interfaces through which we access the functionality of these components. We should ensure that the interfaces are high-level and that they focus on business processes and not on implementation details. The interfaces act as the contracts between the components. The interfaces represent a part of the integration architecture that we should not change – each change will influence all dependent components.

Keep in mind, however, that we can still add operations to existing interfaces without creating problems on related components. Therefore we will often introduce modified methods as new methods with a slightly different signature. This protects us from having to change all related components. However, doing this too many times will make the interfaces very hard to use because we will have to cope with the redundancy of methods – we will not know exactly which to use and when. So we have to be very cautious with the interfaces that we define.

Identifying the dependencies is important because they show how the changes to one part of the system will influence other parts. Dependencies between parts of the system can be direct, in which case a change in one part will require a modification to another part. For example, if part A is directly dependent on part B, this means that if we change something in B we also need to update A.

Dependencies can also go through interfaces, which will decouple the direct connection between the two parts of a system. This will obviously be the preferred way and we will model the integration architecture through interfaces, as we have already stressed several times over. Making components dependent only on component interfaces simplifies their management considerably. As long as we do not modify the interfaces we can change the implementation of the component.

Still, we have to be aware which dependencies exist between virtual components, so we will identify them and show them on the diagram. This enables us to efficiently track and measure the complexity. As we apply changes to the architecture, we should also update these diagrams, otherwise they are effectively useless.

The degree of coupling between components can be used to identify and describe the dependencies. Weak coupling shows that the groups are relatively independent, and fewer dependencies between components show that we have gathered the classes correspondingly and that the system will be relatively easy to understand, maintain, and extend.

Strong coupling, on the other hand, indicates that there are many dependencies between components. This suggests that changes to one part of the system (to an interface, for example) will require modifications in many other parts. It also makes the structure of the system less easy to understand. Sometimes strong coupling is a consequence of incorrectly gathered classes and poorly identified components, and in such cases, it might be a good idea to rethink the architecture. Indeed, such re-evaluations can be a normal part of the whole process.

4.2 Reusing Existing Applications

Identifying the relations to existing applications is the first step in this activity. It is recommended to show the relations for each component, because this will make it easier to follow later steps. This stage is dependent on the existing applications that we have. To be able to identify the relations to existing applications we have to be familiar with their functionality, and to achieve this we have to do the analysis of existing applications.

When we have identified the existing applications that the higher-level virtual component has to interact with, we identify the exact sequence of operations that the higher-level virtual component has to invoke in order to get the desired result. To identify the operations and the sequence that needs to be invoked we study the interfaces of existing applications lower-level virtual components and map the desired functionality in the best possible way.

In real-world examples we will frequently be overwhelmed with the complexity of the interfaces that existing applications provide. We will often also be confused about which operations to actually use, because often there will be more than one way to achieve the same result. To model the sequence of operations that have to be invoked we can use UML sequence diagrams. It is very important that we model all possible sequences of operations, including the normal flow of events and any alternative flows in which something could go wrong. In this way we can define how to handle all exceptional situations, how and to whom we should propagate the exceptions, and we will ultimately make our components highly robust.

The sequence of operations sometimes is not enough and the component has to do some calculation, and perform other operations to get the desired result. As such, in this step we must identify what exactly has to be done. The goal is to identify the interaction with the existing application to such a level that we will be able to write code directly from the specification.

It will vary from operation to operation how complex a mapping we will have to use. With a highly complex mapping we might consider representing the whole procedure with an activity diagram too; sometimes we could even use "pseudo code". We have to map each operation of the newly defined higher-level virtual component to lower-level virtual components that represent existing applications. Sometimes we will not be able to find the corresponding methods in the existing applications. This means that the functionality we require is not supported by existing applications, in which case we have to implement it from scratch. Or we might be able to reuse only a part of the whole functionality. Following the proposed integration process we will be able to add the missing functionality in a relatively painless manner.

4.3 Physical Architecture

In this activity we have to select the implementation technologies and physical architecture. The selection of implementation technologies will depend of the used

software platform. We have to take into account the requirements regarding performance and reliability. This will then influence the deployment scenarios that we select.

To achieve acceptable performance we consider locating tightly-coupled components inside a single container and use local access to components to optimize the method invocation performance [21]. To achieve higher reliability we might consider clustering or replication.

To identify the most suitable physical architecture we select a few different candidate architectures first. Then we build prototypes that help us to validate these candidate architectures by the criteria that we have to meet. Only then will we select the final appropriate architecture and do the implementation.

5 Conclusion

In this article we have presented the process model proposal for information systems integration that specifies a disciplined approach to top-down integration. The integration process introduces sound practices, like iterative and incremental development, prototyping and reuse. It specifies the phases, disciplines, and activities. The four integration phases are: data-level, application interface level, business-method-level, and presentation-level phase.

For each phase the integration process defines several disciplines that have to be performed in order to obtain results. Some of these disciplines are equal for all phases, some depend on the phases. We have focused on the technical disciplines only.

Analysis of existing applications and integration design are highly important disciplines for integration projects. We have to get a clear understanding of the existing situation in order to be able to later map the functionality to the newly integrated system. We also need to adapt the design phase to involve existing applications. This is why in this article we have focused on those two disciplines and presented detailed activities which should be carried out as a part of each discipline.

One of the important features of the presented integration process model is its ability to be adapted to specific need of each company, which will be addressed in our future work.

References

[1] J. Lee, K. Siau, S. Hong (2003) Enterprise integration with ERP and EAI, *Communications of the ACM,* ACM, Vol. 46, Iss. 2, pp. 54 – 60.

[2] M. Stonebraker (2002) Too much middleware, *ACM SIGMOD Record,* ACM, Vol. 31, Iss.1, pp. 97 – 106.

[3] J. Sutherland, W. J. van den Heuvel (2002) Enterprise application integration and complex adaptive systems, *Communications of the ACM,* ACM, Vol. 45, Iss. 10, pp. 59 – 64.

[4] Z. Wu, S. Deng, Y. Li (2004) Introducing EAI and Service Components into Process Management, *Proceedings of the Services Computing,* IEEE, Shanghai, pp. 271 – 276.

[5] A. Eyal, T. Milo (2001) Integrating and customizing heterogeneous e-commerce applications, *The VLDB Journal,* Springer, Vol. 10, Iss. 1, pp. 16 – 38.

[6] S. Baker (2002) The three steps to web service integration, *IONA,* www.iona.com.

[7] F. Losavio, D. Ortega, M. Pérez (2002) Modeling EAI, *XII Int. Conference of the Chilean Computer Science Society,* IEEE, Chile, pp. 195 – 204.

[8] I. Gorton, A. Liu (2004) Architectures and Technologies for Enterprise Application Integration, *26th Int. Conference on Software Engineering,* IEEE, Edinburgh, pp. 726 – 727.

[9] P. Maheshwari (2003) Enterprise Application Integration using a Component-based Architecture, *27th Annual Int. Computer Software and Applications Conference,* IEEE, Dallas, pp. 557 – 560.

[10] V. S. Pendyala, S.Y. Shim, J. Z. Gao (2003) An XML Based Framework for Enterprise Application Integration, *Int. Conference on E-Commerce,* IEEE, California, pp. 128 – 133.

[11] D. Gawlick (2001) Infrastructure for Web-based Application Integration, *17th Int. Conference on Data Engineering,* IEEE, Heidelberg, pp. 473 – 477.

[12] B. Hunter, M. Fowler, G. Hohpe (2002) *Agile EAI Methods: Minimizing Risk, Maximizing ROI,* ThoughtWorks Inc.

[13] S. Chatterhee (2004) *Managing EAI Projects in Agile way,* Cap Gemini Ernst & Young Consulting.

[14] M. Fowler, G. Hohpe (2002) *Agile EAI,* ThoughtWorks Inc.

[15] G. Hohpe, W. Istvanick (2002) *Test-Driven Development in Enterprise Integration Projects,* ThoughWorks Inc.

[16] D. S. Linthicum (1999) *Enterprise Application Integration,* Addison Wesley.

[17] M. B. Juric et al. (2001) *Professional J2EE EAI,* Wrox Press Ltd.

[18] M. Silic et al. (2000) *EMRIS - Enotna metodologija razvoja informacijskih sistemov, Zv. 4, Objektni razvoj,* Center Vlade RS za informatiko.

[19] M. Pusnik, B. Sumak, M. B. Juric, M. Hericko (2004) Ocenjevanje pripravljenosti podjetij na proces integracije s pomočjo indeksa integrabilnosti, *7th Internation Multiconference Information Society IS 2004,* Inštitut Jožef Stefan, Ljubljana, pp. 53 – 56.

[20] M. B. Juric et al. (2003) Application integration patterns, *Technology supporting business solutions, Advances in computation: theory and practice,* Nova Science Publishers, New York, pp. 115-138.

[21] M. B. Juric et al. (2002) *J2EE Design Patterns Applied,* Wrox Press Ltd.

Providing Cross-Lingual Information Access with Knowledge-Poor Methods

Ralf Steinberger, Bruno Pouliquen and Camelia Ignat
European Commission – Joint Research Centre (JRC)
Via E. Fermi, T.P. 267, 21020 Ispra (VA), Italy
Firstname.Lastname@jrc.it, http://www.jrc.it/langtech

Keywords: multilingual text analysis; cross-lingual information access; information extraction; document similarity; categorisation; clustering; nomenclatures; thesauri; gazetteers; freely available resources.

Received: July 16, 2004

We are proposing a simple, but efficient approach for a number of multilingual and cross-lingual language technology applications that are not limited to the usual two or three languages, but that can be applied with relatively little effort to larger sets of languages. The approach consists of using existing multilingual linguistic resources such as thesauri, nomenclatures and gazetteers, as well as exploiting the existence of additional more or less language-independent text items such as dates, currency expressions, numbers, names and cognates. Mapping texts onto the multilingual resources and identifying word token links between texts in different languages are basic ingredients for applications such as cross-lingual document similarity calculation, multilingual clustering and categorisation, cross-lingual document retrieval, and tools to provide cross-lingual information access.

Povzetek: članek predstavlja metodo med-jezikovne uporabe sistemov.

1 Background and Motivation

The European Union (EU) currently has 20 official languages, plus a few non-official ones. Most existing text analysis software tools have been developed for a few major languages, while very few resources and tools are available for the less widely spoken languages. There clearly is a need for more tools that can help the European citizens to access textual information written in the other languages.

The 20 official EU languages add up to 190 language pair combinations. Almost all *cross-lingual* text analysis applications, including Machine Translation (MT), Cross-Lingual Information Retrieval (CLIR) and Cross-Lingual News Topic Tracking (CLNTT), make use of *bilingual* equivalences and rules. The few approaches to CLNTT, for instance, are either based on bilingual dictionaries (Wactlar 1999) or use MT (Leek et al. 1999). In the EU setting, interlingua approaches and approaches towards unified multilingual resources, such as EuroWordNet and MULTEXT, clearly gain in attraction. However, there are many more unexploited resources that may not have been developed for machine use, but that can be exploited for multilingual Information Extraction (IE) and to provide cross-lingual information access.

The Language Technology team of the *Joint Research Centre* (JRC) has the aim to produce a number of text analysis applications for ideally all official EU languages (and more) that help users to navigate in large multilingual document collections and that provide them with cross-lingual information access. Due to a lack of manpower and due to the limited availability of machine-usable linguistic resources, we developed the following preferences:

(a) limiting language-specific text processing to a minimum, by using heuristics and other shallow methods;
(b) preference of statistics and Machine Learning (ML) methods over hand-crafted linguistic rules;
(c) making use of various available multilingual lexical resources, even if they were not initially developed for machine use.

While it is clear that more thorough knowledge-driven methods would produce better results in many cases, the JRC's work has shown that a shallow and mostly language-independent approach can yield a number of useful and new text analysis applications while keeping the language-specific effort to between one and three person months of effort per language.

In Section 2, we describe efforts to map documents onto multilingual gazetteers of place names (2.1), onto the product nomenclature TARIC (2.2) and onto the thesaurus Eurovoc (2.3). Section 3 shows how additional text features such as names, dates and cognates can be exploited for the same purpose. Section 4 explains how to deal with some language-specific issues and Section 5 lists a few language-independent methods and tools that can be used together with the resources mentioned in the previous sections. Section 6 shows some useful applications built with the procedures described in this article. In Section 7, we draw some conclusions.

2 Mapping texts onto thesauri, nomenclatures and gazetteers

When mapping a given text onto a knowledge structure such as a thesaurus, we create a vector representation of this text consisting of a choice of thesaurus nodes, and possibly also of the relative importance of various nodes for the text representation. One, but not the only way of carrying out this mapping process is by verifying the

lexical overlap between the document's vocabulary and the terms of the thesaurus. Two documents can be assumed to be similar if they have a similar representation according to the mapping onto this thesaurus.

In a multilingual thesaurus, nodes in the various language versions are linked via language-independent (typically numerical) node identifiers. While the conceptual world of a given language or of a specific thesaurus is, of course, not completely language-independent, the numerical thesaurus links between various language versions are good enough for an interlingua approximation. Two documents written in different languages can thus be assumed to be similar if they have a similar text representation according to this multilingual thesaurus.

Additionally to thesauri, gazetteers and nomenclatures can fulfil the same function. *Gazetteers* are geographical dictionaries, i.e. lists of place names. According to Norvilienė (forthcoming), the term *nomenclature* is used to describe ordered systems of words (e.g. product names) used in a particular discipline (e.g. business or customs), containing a description of entities from a particular domain and their, typically mono-hierarchical, relationship. *Thesauri* are poly-hierarchically ordered systems of concepts and their natural language names that are mainly used for documentation purposes such as indexing and retrieval.

The aim of this section is to show how texts can be mapped onto one or more thesauri to create a multi-faceted language-independent document representation. The more thesauri can be used, the more information will be available for the document representation and the better documents can be compared with each other. The following sub-sections sketch our current mapping process onto various such lexical knowledge sources.

2.1 Gazetteers of place names

The process of mapping documents onto gazetteers have been described in detail in Pouliquen et al. (2004a) so that this selection will only summarise major issues. Further improvements are reported in Kimler (2004).

Unlike people's names and other named entities, place names cannot be recognised by searching for patterns in text because there are as good as no contextual clues (Gey 2000). Instead, geographical place name recognition has to rely on gazetteers and can only be carried out via a lookup of text words in the gazetteer. As places are spelled with a first uppercase letter in EU languages, only uppercase words need to be looked up. The lookup process sounds simple, but there are five major difficulties:

(a) Place names can also be words in one or more languages, such as 'And' (Iran) and 'Split' (Croatia);
(b) Some place names are homonymic with people's names, such as 'Victoria' (capital of the Seychelles, and others) and 'Annan' (UK);
(c) Many major places have varying names in different languages (exonyms; Venezia vs. Venice, etc.) or even in the same language ('Saint Petersburg', 'Saint Pétersbourg', 'Санкт-Петербург' [Sankt-Peterbúrg], 'Leningrad', 'Petrograd', etc.).
(d) Multiple places share the same name, such as the fourteen cities and villages in the world called 'Paris';
(e) Simple lookup procedures will fail when place names are inflected.

While place name recognition in general is a well-understood named entity recognition task, disambiguation between various homographic place names (issue d) has only recently been tackled (Pouliquen et al. 2004a). Issue (b) can be solved by first identifying person names. Additional wrong hits (issue a) can be limited by using language-specific geo-stop word lists containing all those place names that can also be normal words in that language. This reduces recall slightly, but improves precision considerably. Exonym recognition (issue c) has to rely on an exhaustive multilingual database. For a list of freely available gazetteers, see Steinberger et al. (2004). Problems related to inflection (issue e) will be discussed in Section 4.

The result of the mapping process is thus a vector of place names where each place name is a dimension and the frequency with which it has been mentioned in the text is the length of the vector. For some applications, it may be useful to restrict the recognition resolution to the country level, i.e. each mention of a place in the country adds to the country score. The occurrence frequency and the country score can also be normalised, using TF.IDF or similar, to down-weight the importance of places like *Washington* that are highly frequent in some text types such as world news.

2.2 Nomenclatures of products, etc.

Other views of the same document can be produced by listing all document terms from various other fields, such as products and product groups, professions, medical or electro-technical terms, etc. Various nomenclatures can be downloaded from the internet (see Norvilienė 2004), and many of them are available on the EC's classification server *Ramon*[1]. For instance, there is the electro-technical nomenclature ETIM[2], the *Statistical Classification of Products by Activity in the European Economic Community* CPA, the *Statistical Classification of Economic Activities in the European Community* NACE, and many more.

To date, we have only worked with the *Integrated Tariff of the European Communities* TARIC[3], which is the hierarchical product list used by the Customs Offices in the EU to declare the movement of goods across borders. TARIC is a more detailed version of the so-called *Combined Nomenclature* CN, which is again more detailed than the *Harmonised System* HS used by the World Customs Organisation. TARIC distinguishes about 28,000 headings and subdivisions.

We chose TARIC because it exists in twenty languages (including Slovene) and because it is a rather complete list of tangible items that can be imported or exported. It

[1] Available at http://europa.eu.int/comm/eurostat/ramon/.
[2] See http://www.etim.de/html/download.html
[3] See http://europa.eu.int/comm/taxation_customs/databases/taric_en.htm

TARIC CODE	PRODUCT DESCRIPTION
0702	Tomatoes, fresh or chilled
0702 00 00 07	Cherry tomatoes
0702 00 00 99	Other
0703	Onions, shallots, garlic, leeks and other alliaceous vegetables, fresh or chilled
0703 10	Onions and shallots
0703 20	Garlic
0703 90	Leeks and other alliaceous vegetables
0703 90 00 10	Leeks
0703 90 00 90	Other

Table 1. English product descriptions in TARIC chapter: *Edible Vegetables and Certain Roots and Tubers*.

is in the nature of TARIC that illegal products such as *bombs* and many drugs are not included (although *heroin* and *cocaine* are part of TARIC). It includes live animals, food, chemicals, pharmaceuticals, textiles, precious stones, metals, machinery, vehicles, optical material, works of art, and much more. Table 1 shows some of the product descriptions that are organised hierarchically into up to 5 levels (two digits per level).

Knowing which of the products and product groups are referred to in a text can be very useful to generate a product-related document representation, i.e. a vector of products and their relative importance in the text. We can furthermore use the numerical TARIC codes as an interlingua to represent the product aspect of document written in the twenty languages in which the product nomenclature exists. However, before being able to use the product lists of this resource in a lookup process, we needed to overcome several difficulties:

(a) As the entire TARIC product description (e.g. "*Leeks and other alliaceous vegetables*" in code 070390) will not be found verbatim in the text, the product terminology first needs to be extracted from the description (e.g. *leeks* and *alliaceous vegetables* in Table 1).

(b) Usually, the plural forms are used in TARIC so that the singular or other inflected forms need to be added for the lookup process to be successful. For further issues concerning inflection of words and suffixes, see Section 4.

(c) Syntactic co-ordination constructions such as in code 0703 need to be resolved and expanded out to produce lists such as *fresh onions*, *chilled onions*, *fresh shallots*, *chilled alliaceous vegetables*, etc.

(d) This process typically results in product lists such as *fresh onions* and *chilled onions*, while the most usual underspecified term *onions* is not part of the list. This needs to be added.

(e) While multi-word terms are usually monosemous, many single-word terms such as *onion* or *juice* can be part of many different TARIC classes as there are many different types of juices and onions (*wild onions*, *pearl onions*, *dried onions*, etc.). As we did not want to miss frequently used products such as *onions* or *juice*, and we did not want one term to trigger many different TARIC classes, we decided to add about 350 super-groups such as *vegetables* and *milk products* and to place the under-specified term directly under the super-group.

They ate young river salmon with cream and potatoes.

(Milk product)-EN 0401000000 80	cream, milk	leite, nata
(vegetable)-EN 0710100000 80	potatoe, potatoes	batatas
(fish)-EN 0301991910 80	young river salmon	alívius de salmío

Figure 2. Automatic recognition of products in English text. Display of the results in English and Portuguese.

These steps were carried out, mostly by the *Centre for Information and Language Processing* CIS[4] at the University of Munich in Germany, in the context of a collaborative agreement, for the languages English, German, French, Spanish, Italian and Portuguese. In the semi-automatic process, heuristics were used and results were checked manually. Inflection forms were added by making use of extensive morphological dictionaries available at CIS. The English and Italian dictionary resources created by CIS were then checked thoroughly for correctness at the JRC.

The resulting dictionaries are thus of the form SUPER-GROUP | CODE | TERM where several terms are allowed for the same code if written one term per line, and several codes are obviously allowed for each super-group. The super-group column furthermore allows us to do a more coarse-grained classification of texts so that documents triggering the class *vegetables* several times are identified as similar even if they do not mention the same vegetables. To date, the dictionaries have been developed for the languages English, Italian, German, French, Spanish and Portuguese.

Regarding the recognition of the derived product terminology in the text, the same lookup procedure can be used as for geographical place names. However, in most European languages, products are not spelled with a first uppercase letter so that all words need to be checked against the terms in the product list.

The difficulties involved in the lookup process are again linked to polysemous words like *bush*, *joint*, *bus*, etc. Some of these terms belong to very different TARIC classes (e.g. *joint*). Others are simply homographic with words not related to products (e.g. *Bush*). For testing, we applied the system to various text types and, more importantly, to the 10,000 top frequent words derived from reference corpora. This gave us a good idea of the most frequent missing products, which were then added to the dictionaries. Furthermore, this helped us to identify those high-frequency words that are homographic with products and that could thus potentially generate wrong hits. Depending on the type of problem, we used one of two solutions. (a) For words triggering different TARIC product classes, we usually amended the dictionary by adding some additional specification (e.g. *joint* was changed to *rubber joint*) that helps in the disambiguation. The disadvantage is that the single word *joint* will no longer be

[4] See http://www.cis.uni-muenchen.de/

recognised. (b) For words that are homographic with non-product vocabulary of the language (e.g. *Bush*), we produced a language-dependent *product stop word list* containing all those words that the system should not recognise. This helps to avoid that the US president triggers the product class *live plants*. We thus decided to sacrifice recall for precision.

The effort to prepare and tune the product dictionaries for each language ranges between two and six months per language, but we foresaw that the advantage of mapping texts onto the TARIC nomenclature with its encompassing coverage would be worth the effort. The result of the product recognition procedure is thus a product information extraction tool that allows us also to provide users with product-specific cross-lingual information access and to produce a product-specific feature vector for each document, which can be used for monolingual and cross-lingual document similarity calculation.

The TARIC nomenclature is seemingly distributed for free, but the dictionaries derived from it cannot currently be made available due to the agreement with CIS.

2.3 Thesauri and classification systems

Libraries and documentation centres of most large organisations use hierarchically organised thesauri or flat lists of subject domain descriptions as classification systems to store and retrieve their documents. Documents are often multiply classified, meaning that each document is marked as belonging to several classes (*multi-label categorisation*). Such a classification of a document leads to yet another vector space representation of documents, using the descriptors as dimensions and, if the descriptors are ordered or weighted, the weight as vector length.

The European Parliament (EP) and the European Commission (EC) have jointly developed a thesaurus called EUROVOC (Eurovoc 1995) that is used by them and about twenty regional and national European parliaments to index (i.e. classify) their texts. Though other classification systems exist, EUROVOC is adapted by a growing number of national organisations so that it has now become sort of a standard. To obtain a licence, it is necessary to contact the EC's Publications Office OPOCE.

EUROVOC is a wide-coverage thesaurus that organises its over 6,000 descriptors (classes) from 21 different *fields* (e.g. *politics*, *finance*, *science*, *social questions*, *organisations*, *foodstuff*, etc.) hierarchically into a maximum of 8 levels. EUROVOC exists in currently 22 languages where each numerical descriptor code has exactly one terminological correspondence per language.

As EUROVOC is a wide-coverage thesaurus with only 6000 classes, its descriptors are mostly rather high-level, conceptual terms. Examples are PROTECTION OF MINORITIES, FISHERY MANAGEMENT and CONSTRUCTION AND TOWN PLANNING.[5] Unlike the concrete low-level terms from TARIC and many other nomenclatures, EUROVOC descriptors cannot normally be *extracted* from texts, i.e. they can only rarely be found via a lookup procedure. Instead, EUROVOC classification is a keyword *assignment* task, i.e. the most pertinent descriptors from an independent reference list (the thesaurus) are assigned to a text even if these terms do not occur verbatim in the text.

In the various European parliaments, this assignment is done manually by professional librarians, but the JRC has developed a system that learns from manually classified documents to assign a ranked list of EUROVOC descriptors to any given text. This work is described in detail in Pouliquen et al. (2003a) so that we only summarise the procedure here: The system maps documents onto EUROVOC by carrying out category-ranking classification using Machine Learning methods. In an inductive process, it builds a profile-based classifier by observing the manual classification on a training set of documents with only positive examples.

Before feeding the training texts to the ML algorithm, some linguistic pre-processing was carried out to lemmatise words and to mark up multi-word terms such as *power_plant* and *New_York* as one token and a large stop word list of words with low semantic content was used. However, tests have shown that lemmatisation and multi-word mark-up had only little impact on the performance for Spanish and English. Assignment results for the highly inflected Finnish language were very comparable, showing that the statistical method can be applied without using linguistic tools, if necessary.

The outcome of the mapping process for a given text is a ranked list of the EUROVOC classes that are most pertinent for this text.

Due to the multilingual nature of EUROVOC, this representation is independent of the text language so that it is very suitable for cross-lingual document similarity calculation. The system has currently been trained for thirteen languages so that documents written in any of these languages can be represented with the same language-independent EUROVOC descriptor vector. Unlike the applications described in sections 2.1 and 2.2, this Machine Learning method to map documents onto thesauri requires training material, i.e. documents that have been manually classified. For other approaches to map documents onto thesauri and for a list of freely available thesauri, see Steinberger (2004).

3 Language-independent text features

The mapping processes described in section 2 yield several vector space document representations, one for each thesaurus, nomenclature, gazetteer or word list used. Further multilingual representations can be generated by extracting named entities to create lists of text features such as (a) date or (b) currency expressions, (c) numbers and (d) names, as these can be represented in a normalised, language-independent format. For an introduction to the state of the art of the field of Named Entity Recognition (NER), see Daille & Morin (2000). Names of people or organisations are not strictly language-independent because names may be written differently depending on the language (and sometimes even within the same language), but at least among European languages many

[5]We write all EUROVOC descriptors in small caps.

names are spelled the same. Due to the historical relatedness of many European languages, there are even (e) a few general language words that are similar or the same. These are usually referred to as *cognates*. The English and German words 'finger', 'arm', 'demonstration', 'computer', etc. are some examples. In this section, we describe how these five additional text features can be recognised and exploited to contribute to linking related documents both monolingually and across languages.

3.1 Date and currency expressions

Within the same language, there are usually different ways of writing a certain date or currency expression (e.g. English *13 October 2004, 13/10/2004, 13.10.2004, thirteenth of October of the year two thousand and four, etc.*). Some of these date expressions may be the same as in other languages (e.g. 13.10.2004), but others are not. As the underlying concept is the same, namely a reference to a specific date in the same time reference system, the concept can be expressed in a standard way (see, for instance, ISO standard ISO-8601) so that it is the same across languages. For dates, we currently use 'DD'YYYYMMDD. Expressions such as 13.10.2004 are thus normalised to DD20041013.

Ignat et al. (2003) present a tool that recognises and normalises date expressions so that we will not describe this application here. It is a language-independent software tool that uses language-specific parameter files, one per language.

For some document types such as news articles, a list of the normalised date expressions can be a meaningful signature of the text. Together with further signatures for names, etc., documents can be described rather accurately. Following recognition, date expressions can be highlighted in text for faster retrieval. Another advantage of the application is that, once the recognised dates are normalised and stored in a database, users can search for all articles mentioning a date in a certain period, by using a simple SQL query.

3.2 Proper names

According to Gey (2000), 30% of content-bearing words in journalistic text are proper names such as names of people and of organisations. Friburger & Maurel (2002) showed that names recognised in text are very valuable for document similarity calculation, but say that the usage of names alone is not sufficient for this purpose. It is obvious, though, that a list of proper names can be a highly significant signature for at least journalistic text. If combined with further signatures, as proposed in this article, name lists can be very powerful.

Proper name recognition is a subject area that is very well understood and a number of named entity recognition (NER) tools are available either commercially or for research. At the JRC, we are currently using two alternative approaches to recognise people's names: (a) a PERL tool with regular expressions that identifies sequences of uppercase words as names if they are introduced or followed by cue words such as *President*, *Professor*, *teacher*, etc.; (b) the part-of-speech output of the readily trained *Tree Tagger*[6], combined with some minimalist local grammar rules. Until now, we have exploited the Tree Tagger tool only for English text, although trained Tree Tagger versions are also available for French, German and Italian. The less sophisticated PERL tool misses names that are not surrounded by cue words, but it has the advantage that it is just a question of a few hours to extend it to new languages, so that we are now able to recognise names in English, French, German, Spanish, Italian, Slovene, Estonian and Bulgarian.

Spelling	Language(s)
Vladimir Putin	DA, EN, ES, IT, NO, SV
Vladimir Poetin	NL
Vladimir Poutine	FR
Vladimir V Putin	EN
Vladmir Putin	EN
Vladímir Putin	ES
Wladimir Putin	DE
Władimir Putin	PL

Table 4. Variations of the name of the Russian President found in news texts in various languages.

Even within the same language and the same text, authors often use different versions of the same name. This is not only true for foreign names such as *Al Qaida* (*Al Qaeda*, *Al Kaida*, etc.), but also for known names such as *George Bush* (*George W. Bush, George Bush Jr.*, *George Walker Bush*, etc.). After having examined a number of approximative matching techniques, we decided to implement a simple letter trigram measure that allows us to recognise many monolingual and cross-lingual name variations found, as shown in Table 4. The most frequent variation is now taken as the prototypical one that is stored in the database, and all others are stored in an alias list of variations. Via an automatic lookup of the Wikipedia online encyclopaedia in various languages[7], further name variations such as Japanese □□□□□□ □□□□, Chinese □□ and Russian Владимир Путин can be found automatically.

By using the PERL regular expressions continuously over time, a database of frequently mentioned person's names can be built up so that names can then be found in new text by using simple lookup procedures, without the need for cue words.

The result of the proper name recognition is thus a list of people's names mentioned in a given text, together with possible name variants and with information on how often the name was mentioned, both in the given text and in other texts over time. This latter frequency can be used to weight the relevance of names in a given text, using TF.IDF or a related measure, in order to down-weight frequently mentioned names such as *George Bush* and to highlight new or rarely used person names.

[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[7] See various language versions at http://en.wikipedia.org, http://de.wikipedia.org, etc.

3.3 Cognates and numbers

When comparing the tokens of texts written in different languages with each other, one can frequently find some overlap. This overlap usually consists of (a) numbers in numerical form (e.g. *596*), (b) names or (c) other words that are coincidentally the same across languages (*cognates*). Cognates are normally due to common historical roots (e.g. English *finger* and *arm* vs. German *Finger* and *Arm*) or because they adapted the same loanwords (e.g. German *Computer* and Italian *computer*). These three types of identical text tokens can be exploited to contribute constructively to cross-lingual document similarity calculation. Two news articles about the same event written in English and Spanish, for instance, are likely to have a number of tokens in common, while two articles about different events are likely to have less tokens in common.

Obviously, several limitations are linked to this approach:

(a) Number formats can differ from one language to the other, for instance due to the different usage of number separators (e.g. English 1,000.00 vs. German 1.000,00), but more often than not there is no difference (1000 is used in both languages).
(b) Names of people and places often differ from one language to the other because of different pronunciation rules (e.g. English *Al Qaeda* vs. German *Al Kaida*), or for historical reasons (e.g. English *Venice* vs. German *Venedig* vs. French *Venise*, etc.). Languages with different writing systems are much less likely to have word tokens in common, even if the pronunciation of the words is identical (e.g. Italian *Venezia* vs. Greek *Βενετία*).
(c) So-called *false friends* (words that are the same without sharing the same meaning, such as English *manifestation* and French *manifestation* or English *war* and German *war*) would cause false hits.

In spite of these limitations, we are already exploiting identical cognates, numbers and other identical text tokens across languages in a system for multilingual news topic tracking, as described in Pouliquen et al. (2004b).

4 Language-specific issues

From a linguistic point of view, the procedures described in the previous sections are relatively simplistic. They mainly rely on tokenisation, case information, dictionary lookup procedures, stop word lists, simple local patterns, heuristics, and statistics and Machine Learning methods operating on 'words' without part-of-speech information. Many of these procedures will work well with English texts as English has a rather poor morphology. However, this approach will be much less successful for more highly inflected languages like Hungarian or those of the Slavic language family.

It should be possible to overcome most of these phenomena with the help of good morphology tools, but these are not available to us for the large range of languages we are interested in (all twenty official EU languages and more!). As the manpower available in the JRC's Language Technology group is rather limited, as well, we had to resort, yet again, to some simple heuristics that would allow us to benefit as much as possible from the available multilingual resources and the language-independent text features while limiting the effort to a few weeks per language. With the existing applications already being set up, adding the language-specific resources for a new language takes between two and twelve weeks. Extracting the relevant terminology from the TARIC product description and preparing it for the application described in section 2.2 is rather labour-intensive so that it takes an additional estimated 12 weeks. It is clear that not all linguistic phenomena and not all languages can be dealt with, but for a large number of European languages this is sufficient to produce good and very useful text analysis applications, as described in section 6.

For the statistical EUROVOC thesaurus text classification task, experiments with Spanish have shown that, surprisingly, performance gains only approximately 2% when operating on lemmas rather than on inflected words. Furthermore, multilingual performance tests for EUROVOC descriptor assignment on eleven different languages from different language families, including German, Spanish, Finnish and Lithuanian, have shown that performance is rather uniform across the languages. Details about these experiments can be found in Pouliquen et al. (2003a).

Simple dictionary lookup procedures such as for geocoding and product recognition are, however, more sensitive to word form variations because inflected word forms such as *New Yorker* will not be found in text if the gazetteer only contains the base form *New York*. We solve this problem partially by providing language-specific regular expressions that strip potential suffixes off those uppercase words that were found in a text, but not in the place name gazetteer. For instance, if words like *Londonit*, *Frankfurdis* or *New Yorgile* are found in Estonian text, regular expressions will strip *-it* to produce *London* and will replace *dis* to *t* and *gile* to *k* in order to produce *Frankfurt* and *New York*. Together with Finnish, Estonian is known for its extremely sophisticated morphology. However, place names occur with a limited number of case endings (*in/to/from/... London*) so that 37 regular expressions cover most cases. For most languages, a much smaller number of regular expressions is needed. A small evaluation on Estonian news headlines showed that 63 out of 72 place names were recognised correctly (Recall = 87.5%). The remaining nine places were not found because either the place name was not in the database or because the suffix stripping rule was missing (about equal parts). No wrong hits occurred in the test set (precision = 100%).

It should be possible to apply the same suffix-stripping procedure to other kinds of vocabulary lists such as products, professions, etc. However, as these lists are likely to be larger, and we cannot limit our search to upper case words, the lookup process should be slower and it is possible that it will produce more wrong hits.

It is not certain that for an agglutinative language like Hungarian, which can add many different types of suf-

fixes one after the other, suffix stripping is feasible. It would be an interesting experiment to apply cascades of suffix-stripping regular expressions to see whether this helps to find place names, but the danger to get false hits due to over-stripping is big.

Further tokenisation issues arise when dealing with languages such as Chinese which do not mark word borders by a space, and compounding languages like German where (mostly) nouns can be combined to form long words. While, at least in German, expressions like *Berliner actor* (an actor from Berlin) are not compounded (*Berliner Schauspieler*), nouns referring to products are: *Sauerstoffflaschenventilverschluss* (oxygen bottle valve closure).

For most European languages, the uppercase / lowercase distinction can be exploited when looking for the names of people or places. The same is not true for languages like Japanese, Hindi and Arabic. Furthermore, case rules even differ to some extent between languages such as English and French (e.g. *the English* vs. *les anglais*) so that rules either have to be adapted specifically to each language or lower recall has to be accepted when looking only at uppercase words.

5 Language-independent procedures and applications

In the highly multilingual setting of the set of applications discussed in this article, language-independent text analysis procedures are very useful. We currently use the following applications:

(a) An automatic language guessing tool using letter bigram and trigram statistics, that has currently been trained for 25 languages.

(b) A keyword extraction tool that identifies the statistically most salient words and their relative importance (their *keyness*) by comparing the word frequency in the text with an average word frequency as found in large reference corpora. While we use the log-likelihood formula to extract and rank the words, other formulae like TF.IDF or chi-square are possible alternatives. A language-specific list of stop words can be used to stop some words from being identified as keywords that are low in semantic content or that are meaningless when being out of context. A ranked list of keywords for a document is a good vector space representation of this document.

(c) A tool to measure the similarity between two documents by calculating the cosine or another similarity measure between the vector space representations of two documents. Monolingually, the list of extracted keywords and their keyness can be used as input. For cross-lingual similarity calculation, the features discussed in Sections 2 and 3 can be fed to the system.

(d) This document similarity measure can be used for a number of applications, including hierarchical unsupervised document clustering, classification and query-by-example document retrieval.

Further applications that can be based on language-independent methods are automatic document summarisation by extracting the most relevant sentences (e.g. those containing most keywords), and the generation of document maps. Document maps such as Kohonen maps are two-dimensional representations of the multi-dimensional document space that can be useful to get a first overview of the main contents of a large document collection or to navigate in the document collection.

6 Applications

At the JRC, we combine applications based on the language-independent algorithms listed in section 5 with the information extracted according to the procedures described in sections 2 and 3. In spite of the relatively shallow linguistic processing, we were able to produce applications that are being used as regular in-house services and for the ad-hoc analysis of document collections given to us by various users.

Once entities such as dates, names or products have been identified, they can be highlighted in text in different colours to allow users to find them quickly. For foreign language text, the entity can be displayed in another language to give users information about a text that they may not otherwise understand (*cross-lingual information access*). The various information aspects (products, places, keywords, etc.) extracted from unrestricted and unstructured text can also be displayed together to provide users with sort of a condensed document profile. Those information aspects that are linked to multilingual nomenclatures, gazetteers and thesauri can furthermore be displayed in languages other than the document language.

The structured meta-information is stored in a database to enable users to search document collections by using this meta-data as features. This makes it possible, for instance, to search for all documents mentioning tobacco products, making reference to Turkey and mentioning a date in the range 1.01.2003 and 31.03.2003.

When the reference of geographical place names has been identified unambiguously, i.e. when we have identified latitude and longitude of the places, it is easy to create a map showing the geographical coverage of a document, of a cluster of documents or of a whole document collection.

Similar articles of a large document collection can be clustered automatically into groups of related texts. In the news analysis system presented in Pouliquen et al. (2004b), major news clusters of one day are then compared to the major news clusters of previous days (*historical topic tracking*). The JRC's *cross-lingual news tracking* system represents each cluster by three different vectors, based on Eurovoc descriptors, place names and cognates. When comparing this document representation with those of clusters in other languages, each of the three vectors contributes with a different weight to the overall similarity between the clusters of documents written in different languages, as described in Pouliquen (2004b).

Another usage of the cross-lingual document similarity calculation is the automatic compilation of collections of parallel (or comparable) texts to train and test information extraction or Machine Translation software. Tests described in (Pouliquen et al. 2003b) show that the map-

ping of documents onto Eurovoc alone allows a rather accurate cross-lingual document similarity calculation: In over 96% of cases, the correct Spanish translation of an English text was automatically selected out of 820 possible candidates. This result shows that processes to map documents onto a multilingual thesaurus can lead to extremely powerful applications. Cross-lingual document similarity calculation is also an essential ingredient for cross-lingual document plagiarism detection, an application for which, to our knowledge, no solutions have been proposed to date.

7 Conclusion

The intention of this article was to describe how multilingual knowledge sources such as gazetteers, vocabulary lists, nomenclatures and thesauri, as well as language-independent text features such as dates, can be exploited for information extraction tasks, to provide cross-lingual information access and to calculate cross-lingual document similarity, which itself is a basic ingredient for many more text analysis applications. We furthermore wanted to show how relatively naïve text analysis tools can be helpful to develop powerful text analysis applications for many different languages with rather little effort, once the methodology has been decided on and the tools have been set up. At the JRC, we have already developed the language-specific resources for a number of European languages and we are currently making an effort to extend this tool set to all twenty official languages of the European Union. While we have no doubt that it is possible to produce better results with more thorough linguistic methods, such labour-intensive language-specific work is not an option for our small team whose aim it is to work on 20 or more languages. Instead, we exploit existing multilingual lexical resources (even if they had not initially been developed for machine use) and language-independent text features, and we make use of Machine Learning techniques, statistical methods and heuristics. We believe to have shown that this approach can lead to good results and that it is even possible to produce working versions of novel applications such as cross-lingual news topic tracking using an interlingua document representation. The effort required to develop the language-specific resources for a new language ranges between one week and three months for the applications we are currently using. Extracting and developing TARIC product nomenclature terms is a comparatively labour-intensive task that requires an additional estimated two to three months.

Individual applications out of the set presented in this paper have been tested and proven, including date and place name recognition, EUROVOC thesaurus descriptor assignment, monolingual news clustering and news topic tracking, and cross-lingual news topic tracking. A number of other applications presented here still need to be evaluated formally. Furthermore, it would be useful to carry out a thorough one-by-one evaluation of the effectiveness of each of the text features presented here, and of their relative impact for cross-lingual document similarity calculation.

Acknowledgements

Many people have contributed to developing the tool set described in this paper and to developing and evaluating the language-specific resources for various languages. We would particularly like to thank Laima Norvilienė (born Cekyte) and Irina Temnikova for their help with the product recognition tool, Victoria Fernandez Mera, Elisabet Lindkvist Michailaki and Arturo Montejo-Ráez for their help regarding the EUROVOC thesaurus indexing application, Marco Kimler for his refinement of the geocoding tool, and Emilia Käsper, Ippolita Valerio, Tom de Groeve, Victoria Fernandez Mera, Tomaž Erjavec, Christian Gold and Irina Temnikova for their help in creating language-specific resources for Estonian, Italian, Dutch, Spanish, Slovene, German, Bulgarian and Russian. We would also like to thank the JRC's Web Technology team for providing us with the multilingual news collection to develop and test many of the applications described here.

References

Daille Béatrice & Emmanuel Morin (2000). *Reconnaissance automatique des noms propres de la langue écrite : les récentes réalisations*. In: D. Maurel & F. Guenthner : Traitement automatique des langues vol. 41, No. 3. Traitement des noms propres, pp. 601-623. Hermes, Paris.

Eurovoc (1995). *Thesaurus EUROVOC - Volume 2: Subject-Oriented Version*. Ed. 3/English Language. Annex to the index of the Official Journal of the EC. Luxembourg, Office for Official Publications of the European Communities. http://europa.eu.int/celex/eurovoc.

Friburger N. & D. Maurel (2002). *Textual Similarity Based on Proper Names*. Proceedings of the workshop 'Mathematical/Formal Methods in Information Retrieval' (MFIR'2002) at the 25th ACM SIGIR Conference, pp. 155-167. Tampere, Finland.

Gey Frederic (2000). *Research to Improve Cross-Language Retrieval – Position Paper for CLEF*. In C. Peters (ed.): *Cross-Language Information Retrieval and Evaluation*, Workshop of Cross-Language Evaluation Forum (CLEF'2000), Lisbon, Portugal. Lecture Notes in Computer Science 2069, Springer.

Ignat Camelia, Bruno Pouliquen, António Ribeiro & Ralf Steinberger (2003). *Extending an Information Extraction Tool Set to Central and Eastern European Languages*. In: Proceedings of the Workshop 'Information Extraction for Slavonic and other Central and Eastern European Languages' (IESL'2003), held at RANLP'2003, pp. 33-39. Borovets, Bulgaria.

Kimler Marco (2004). *Geo-coding: Recognition of geographical references in unstructured text, and their visualization*. Unpublished diploma thesis, Fachhochschule Hof, Germany. http://www.jrc.it/langtech

Leek Tim, Hubert Jin, Sreenivasa Sista & Richard Schwartz (1999). *The BBN Crosslingual Topic Detection and Tracking System*. In 1999 TDT Evaluation System Summary Papers. http://www.nist.gov/speech/tests/tdt/tdt99/papers

Norvilienė Laima (forthcoming). *Computerlinguistische Analyse von Produktthesauri*. Unpublished Master's Thesis. Ludwig-Maximilians University Munich, Cen-

tre for Information and Language Processing.

Pouliquen Bruno, Ralf Steinberger & Camelia Ignat (2003a). *Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus*. In: Proceedings of the Workshop 'Ontologies and Information Extraction' at the EUROLAN'2003 Summer School. Bucharest, Romania.

Pouliquen Bruno, Ralf Steinberger & Camelia Ignat (2003b). *Automatic Identification of Document Translations in Large Multilingual Document Collections*. Proceedings of RANLP'2003. Borovets, Bulgaria.

Pouliquen Bruno, Ralf Steinberger, Camelia Ignat & Tom de Groeve (2004a). *Geographical Information Recognition and Visualisation in Texts Written in Various Languages*. In: Proceedings of the 19th Annual ACM Symposium on Applied Computing (SAC'2004), Information Access and Retrieval (SAC-IAR), vol. 2, pp. 1051-1058. Nicosia, Cyprus.

Pouliquen Bruno, Ralf Steinberger, Camelia Ignat, Emilia Käsper & Irina Temnikova (2004b). *Multilingual and Cross-lingual News Topic Tracking*. In: Proceedings of CoLing'2004). Geneva, Switzerland.

Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (2004). *Exploiting multilingual nomenclatures and language-independent text features as an interlingua for cross-lingual text analysis applications*. In Proceedings of IS'2004. Fourth Language Technologies Conference, pp. 2-12. Ljubljana, Slovenia.

Wactlar H.D. (1999). *New Directions in Video Information Extraction and Summarization*. In Proceedings of the 10th DELOS Workshop, Sanorini, Greece.

Conducting the Wizard-of-Oz Experiment

Melita Hajdinjak and France Mihelič
University of Ljubljana, Faculty of Electrical Engineering, Slovenia
{melita.hajdinjak,france.mihelic}@fe.uni-lj.si

Keywords: natural-language dialogue systems, Wizard-of-Oz experiment, dialogue-manager evaluation, PARADISE evaluation framework

Received: June 11, 2004

Human-human and human-computer dialogues differ in such an important way that the data from human interaction becomes an unreliable source of information for some important aspects of designing natural-language dialogue systems. Therefore, we began the process of developing a natural-language, weather-information-providing dialogue system by conducting the Wizard-of-Oz *(WOZ) experiment. In WOZ experiments subjects are told to interact with a computer system, though in fact they are not since the system is partly simulated by a human, the wizard. During the development of the weather-information-providing dialogue system this experiment was used twice. While the aim of the first WOZ experiment was, first of all, to gather human-computer data, the aim of the second WOZ experiment was to evaluate the newly-implemented dialogue-manager component. The evaluation was carried out using the PARADISE evaluation framework, which maintains that the system's primary objective is to maximize user satisfaction, and it derives a combined performance metric for a dialogue system as a weighted linear combination of task-success measures and dialogue costs.*

Povzetek: članek opisuje WOZ poskus, tj. testiranje komunikacije človek–računalnik.

1 Introduction

In a nutshell, a dialogue system or a voice interface enables users to interact with some application using spoken language. The application in question, for example, can be a piece of hardware (*command & control systems*) or a kind of database (*interactive voice response*, *information-providing dialogue systems*, *problem-solving dialogue systems*). A detailed overview is given by Krahmer [1]. In this article, we will focus on information-providing, natural-language dialogue systems, which have already been developed for different domains, for instance, restaurant information [2], theatre information [3], train travel information [4, 5], air travel information [6, 7], and weather information [8].

It is generally acknowledged that developing a successful computational model of natural-language dialogues requires extensive analysis of sample dialogues, but the question that arises is whether these sample dialogues should be human dialogues. On the one hand, it has often been argued that human dialogues should be regarded as a guidance and a norm for the design of natural-language dialogue systems, i.e., that a natural dialogue between a person and a computer should resemble a dialogue between humans as much as possible. On the other hand, a computer is not a person. Consequently, human-human and human-computer dialogues differ in such an important way that the data from human interaction becomes an unreliable source of information for some important aspects of designing natural-language dialogue systems, in particular the style and complexity of interaction [9, 10]. This is because the users of natural-language dialogue systems are influenced by the system's language [11], i.e., they often adapt their behaviour to the expected language abilities of the counterpart. Therefore, instead of gathering human-human data, we started the process of designing the Slovenian and Croatian spoken, weather-information-providing dialogue system [12] by conducting the Wizard-of-Oz (WOZ) experiment [10, 13], which is a more accurate predictor of actual human-computer interaction [9]. This is because in WOZ studies subjects are told to interact with a computer system, though in fact they are not. The system is at least partly simulated by a human, the wizard, with the consequence that the subjects can be given more freedom of expression or be constrained in more systematic ways than this is the case in already existing dialogue systems.

During the development of the weather-information-providing dialogue system the WOZ experiment was used twice. While the aim of the first WOZ experiment (section 2) was, first of all, to gather human-computer data, the aim of the second WOZ experiment (section 3) was to evaluate the newly-implemented dialogue-manager component [14]. Consequently, while in the first WOZ experiment dialogue management was still one of the tasks of the wizard, in the second WOZ experiment it was performed by the newly-implemented dialogue-manager component. The differences in the data from both WOZ experiments therefore reflect the dialogue manager's performance. However, this data was evaluated with the PARADISE evaluation framework [15], i.e., a potential gen-

eral methodology for evaluating and comparing the performance of spoken dialogue agents, which maintains that the system's primary objective is to maximize user satisfaction, and it derives a combined performance metric for a dialogue system as a weighted linear combination of task-success measures and dialogue costs.

2 First WOZ Experiment

The aim of the first WOZ experiment [13] was to gather data that would serve as the basis for the construction of the dialogue manager and the speech-understanding component within the developing Slovenian and Croatian spoken dialogue system for weather-information retrieval [12]. However, the first WOZ system consisted of the following components:

- ISDN telephony platform,
- weather-information database,
- wizard's graphical interface [13], designed as an internet application, which included facilities for the playback of predefined spoken responses as well as forms, image fields and handle of some keyboard shortcuts,
- natural-language generation module,
- Slovenian text-to-speech synthesis [16].

Hence, the task of the wizard in the first WOZ experiment was to simulate Slovenian speech understanding (speech recognition and natural-language understanding) and dialogue management. Croatian speech understanding was not performed since only Slovene users were being involved into the experiment. During the experiment, the wizard was sitting behind the graphical interface, listened to users' queries and tried to mediate an appropriate response, which was being successively followed by the natural-language-generation process and the text-to-speech process.

However, a total of 76 Slovene users (38 female, 38 male) were chosen to take part in the first WOZ experiment. The statistical distributions of the users' ages, educations, dialects, the telephone units and the background environments from where the telephone calls were made were chosen to simulate the actual scenarios. The users were given verbal instructions about the general functionality of the system and a sheet of paper containing a description of the tasks they were supposed to complete. They had two scenarios to enact. The first task was to obtain a particular piece of weather-forecast information, such as the temperature in Ljubljana or the weather forecast for Slovenia tomorrow, and the second task was a given situation, such as "You are planning a trip to... What are you interested in?", the aim of which was to stimulate the user to ask context-specific questions. After these two scenarios, users were given the freedom to ask additional questions.

In order to evaluate user satisfaction, users were given the user-satisfaction survey [17] used within the PARADISE framework (section 4), which asks to specify the degree to which one agrees with several questions about the behaviour or the performance of the system (**TTS Performance**, **ASR Performance**, **Task Ease**, **Interaction Pace**, **User Expertise**, **System Response**, **Expected Behaviour**, **Future Use**). The answers to the questions were based on a five-class ranking scale from 1, indicating strong disagreement, to 5, indicating strong agreement. All the mean values are given in table 1. A comprehensive **User Satisfaction** was then computed by summing each question's score, and thus ranged in value from a low of 8 to a high of 40. In the first WOZ experiment, the mean **User Satisfaction** value was 34.08, with a standard deviation of 5.07.

	WOZ1	WOZ2
TTS Performance	4.42	4.29
ASR Performance	4.51	4.29
Task Ease	4.27	3.74
Interaction Pace	3.94	3.76
User Expertise	4.40	4.28
System Response	4.23	3.76
Expected Behaviour	4.31	4.04
Future Use	3.99	3.78
User Satisfaction	34.08	31.96

Table 1: The mean user values in the first (*WOZ1*) and the second (*WOZ2*) WOZ experiment.

The spontaneous speech data, named *Slovenian Spontaneous Speech Queries (SSSQ)*, that was collected during the first WOZ experiment, was transcribed with the Transcriber tool [18]. The transcription was labelled for turns and utterances, and special labels for dialectal words and non-speech sounds were added. An example dialogue is given in table 2.

3 Second WOZ Experiment

The second WOZ experiment was carried out in order to evaluate the performance of the newly-implemented dialogue manager [14], build on the basis of the data collected during the first WOZ experiment. Therefore, all the other components of the system remained the same. Hence, in comparison with the first WOZ experiment, the task of the wizard in the second WOZ experiment was only to simulate Slovenian speech understanding. The wizard was sitting behind the dialogue manager's interface and entered the meaning representation of what the user said.

A total of 68 Slovene users (29 female, 39 male) were chosen to take part in the second WOZ experiment. They were given the same instructions and the same user-satisfaction survey as the users in the first experiment. All the mean user values, which were slightly worse than the

S:	*Hello. The dialogue system for weather-forecast information speaking. Can I help you?*
U:	Yes, I would like the visibility in Štajerska.
S:	*Wait a moment, please ...* [the wizard is choosing her answer] *In Štajerska today - the visibility will be more than 10 km. Is there something else?*
U:	Will there be any snowfalls?
S:	*For which location are you asking for?*
U:	Štajerska.
S:	*Wait a moment, please ...* [the wizard is choosing her answer] *No, in Štajerska today – the sky will clear up. Is there something else?*
U:	What about the weather in Poland in the next few days?
S:	*I have information for today.*
U:	So, how will it be in Poland today?
S:	*Wait a moment, please ...* [the wizard is choosing her answer] *In Varšava, Poland - it is cloudy, the air temperature is -6 degrees Celsius. Is there something else?*
U:	Is there some snow in Poland?
S:	*I do not offer this information. Do you have any other question?*
U:	No, thank you. Goodbye.
S:	*Thank you for your cooperation. Goodbye.*

Table 2: The Slovene-English translation of an example dialogue between a user (U) and the WOZ system (S), recorded during the first WOZ experiment.

values from the first WOZ experiment, are given in table 1. The mean **User Satisfaction** value was this time 31.96, with a standard deviation of 4.99. Note, the difference between the mean **User Satisfaction** values in both experiments is expected since the wizard with her human-level intelligence should had been able to manage the dialogue better than the implemented dialogue-manager component.

The Slovenian spontaneous speech data collected during the second WOZ experiment was named *Slovenian Spontaneous Speech Queries 2 (SSSQ2)*.

In agreement with previous studies [9, 10, 11], we observed that in both experiments the users adapted their behaviour to the expected language abilities of the natural-language-spoken WOZ system. In several dialogues the first question was much longer than the following ones and, in case of repetitions, requested by the system, the speech mode became more articulated, slower and/or louder. Moreover, while the wizard was mediating her response some users made fun of the system, they made comments like "What a voice - terribly", "It is thinking", "It is searching in the computer", and they laugh. But such side remarks certainly would be rather strange in a natural information-providing task because, in both experiments, subjects were basically role playing. They were not real users with real information requirements or real time constraints and telephone bills.

4 Dialogue-Manager Evaluation

The dialogue-manager component [14] was evaluated using the PARADISE framework [15], which maintains that the system's primary objective is to maximize user satisfaction, and it derives a combined performance metric for a dialogue system as a weighted linear combination of *task-success measures* and *dialogue costs* (i.e., *dialogue-efficiency costs* and *dialogue-quality costs*). The PARADISE model of performance posits that a performance function can then be derived by applying multivariate linear regression (MLR) with user satisfaction as the dependent variable and task-success measures, dialogue-efficiency costs, and dialogue-quality costs as the independent variables. Here, user satisfaction, which has been frequently used in the literature as an external indicator of the usability of a dialogue system, is calculated with the survey [17], used in our WOZ experiments.

In order to model the performance of both WOZ systems, we selected 17 regression parameters. First, we computed the task-success measure **Kappa coefficient** (κ) [19], reflecting the wizard's typing errors, and the dialogue-efficiency costs **Mean Elapsed Time** (MET), i.e., the mean elapsed time for the completion of the tasks that occurred within the interaction, and **Number of User Turns** (NUT). Second, the following dialogue-quality costs were selected: **Task Completion** (Comp), i.e., the user's perception of completing the given task; **Mean Words per Turn** (MWT), i.e., the mean number of words per user's turns; **Mean Response Time** (MRT), i.e., the mean system-response time; **Max Response Time** (MaxRT), i.e., the maximum system-response time; **Rejection Ratio** (RR), i.e., the ratio of system moves asking for a repetition of the last utterance; **Help-Message Ratio** (HMR), i.e., the ratio of system help moves; **Check Ratio** (CR) and **Number of Check moves** (NC), i.e., the ratio and the number of system

moves checking some information regarding past dialogue events; **Non-Provided Information Ratio** (NPR), i.e., the ratio of user-initiating moves that do not result in relevant information being provided; **No-Data Ratio** (NDR) and **Number of No-Data Responses** (NNR), i.e., the ratio and the number of system moves stating that the requested information is not available; **Relevant-Data Ratio** (RDR), i.e., the ratio of system moves directing the user to select relevant, available data; **Unsuitable-Initiative Ratio** (UIR), i.e., the ratio of user-initiating moves that are out of context; **Non-Initiating Ratio** (NIR), i.e., the ratio of non-initiating user moves.

When applying PARADISE to the data from the first WOZ experiment to derive a performance equation, we found that **Help-Message Ratio**, **Non-Provided-Information Ratio**, **Task Completion**, **Mean System Response Time**, and **Rejection Ratio** were the parameters that significantly contributed to user satisfaction. On the other hand, the most significant parameters in the second WOZ experiment were **Check Ratio**, **Kappa**, **Mean Elapsed Time**, **Non-Provided-Information Ratio**, and **Task Completion**.

Walker et al. [17] found in their experiments that **Task Completion**, rather than **Kappa**, was a significant factor in predicting user satisfaction, and argued that this was because the user's perceptions of task completion sometimes varied from **Kappa**. In our experiments, **Kappa** only referred to the wizard and **Task Completion** was related only with the first task, which could be the reasons why we did not come to the same conclusion. On the one hand, in these experiments, **Kappa** and **Task Completion** were uncorrelated, but on the other hand, in the second WOZ experiment, **Kappa** was an even more significant predictor of user satisfaction.

However, significant predictors of user satisfaction that did not refer to the wizard were **Help-Message Ratio** and **Non-Provided-Information Ratio** in the first experiment, and **Check Ratio** and **Non-Provided-Information Ratio** in the second experiment. The size of the **Help-Message Ratio** is a consequence of the user's behaviour during the conversation, which is, on the other hand, influenced by the system's level of user-friendliness and cooperation. A user-friendly and cooperative dialogue system should not only play an active role in directing the dialogue flow toward a successful conclusion for the user, it should also be able to take the initiative and to instruct the user if he/she asks for help. However, because some novice users of a dialogue system who are not able to adapt quickly are likely to need instructions provided by the system, **Help-Message Ratio** is expected to reflect user satisfaction. Furthermore, because **Check Ratio** is in a way related to the speech-understanding process, which is usually the most problematic part of a dialogue-system's performance, it is inappropriate to try to decrease it at any price. Consequently, user satisfaction can be remarkably improved only by decreasing **Non-Provided-Information Ratio**. This can be done by preventing the dialogue manager from giving no information before first checking that there is no other available data that might be relevant to the user's request, i.e., the dialogue manager should be as flexible as possible in directing the user to select relevant, available data.

5 Conclusion

In this study we have presented the conducted WOZ experiments, aim of which was to gather human-computer data and to evaluate the dialogue-manager component of the developing, Slovenian and Croatian spoken dialogue system for weather-information retrieval.

The results of applying PARADISE to the data from both WOZ experiments have been given. These have shown that user satisfaction is significantly correlated with the percentage of those user initiatives that did not result in relevant information being provided. We concluded that the ability to direct the user to select relevant, available data is of great importance, and, consequently, that a dialogue system should give no information only if there is no other available data that might be relevant to the user's request.

References

[1] Krahmer, E.J. (2001) *The Science and Art of Voice Interfaces*, Philips research report, Eindhoven, The Netherlands.

[2] Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Fosler, E., and Morgan, N. (1994) The Berkeley Restaurant Project, *Proc. of the 3rd International Conference on Spoken Language Processing*, Acoustical Society of Japan, Yokohama, Japan, pp. 2139–2142.

[3] van der Hoeven, G., Andernach, J., van der Burgt, S., Kruijff, J., Nijholt, A., Schaake, J., and de Jong, F. (1995) A Natural Language Accesible Theatre Information and Booking System, *Proc. of the 1st International Workshop on Applications of Natural Language to Data Bases*, AFCET, Versailles, France, pp. 271–285.

[4] Eckert, W., Kuhn, T., Niemann, H., Rieck, S., Scheuer, A., and Schukat-Talamazzini, E.G. (1993) A Spoken Dialogue System for German Intercity Train Timetable Inquiries, *Proc. of the 3rd European Conference on Speech Communication and Technology*, ISCA, Berlin, Germany, pp. 1871–1874.

[5] Allen, J.F., Schubert, L.K., Ferguson, G., Heeman, P., Hwang, C.-H., Kato, T., Light, M., Martin, N.G., Miller, B.W., Poesio, M., and Traum, D.R. (1995) The TRAINS Project: A Case Study in Building a Conversational Planning Agent, *Journal of Experimental and Theoretical AI*, Taylor and Francis Ltd, pp. **7** 7–48.

[6] Ipšić, I., Mihelič, F., Dobrišek, S., Gros, J., and Pavešić, N. (1999) A Slovenian Spoken Dialogue System for Air Flight Inquires, *Proc. of the 6th European Conference on Speech Communication and Technology*, ISCA, Budapest, Hungary, pp. 2659–2662.

[7] Stallard, D. (2000) Talk'n'Travel: A Conversational System for Air Travel Planning, *Proc. of the 6th Applied Natural Language Processing Conference*, Association for Computational Linguistics, Seattle, USA, pp. 68–75.

[8] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., and Hetherington, L. (2000) Jupiter: A Telephone Based Conversational Interface for Weather Information, *IEEE Transactions on Speech and Audio Processing*, IEEE, pp. **8(1)** 85–96.

[9] Fraser, N.M. and Gilbert, G.N. (1991) Simulating Speech Systems, *Computer, Speech and Language*, Academic Press, pp. **5(1)** 81–99.

[10] Dahlbäck, N., Jönsson, A., and Ahrenberg, L. (1993) Wizard of Oz Studies: Why and How, *Proc. of the International Workshop on Intelligent User Interfaces*, ACM Press, Orlando, USA, pp. 193–200.

[11] Zoltan-Ford, E. (1991) How to Get People to Say and Type What Computers Can Understand, *Journal of Man-Machine Studies*, Academic Press, pp. **34** 527–547.

[12] Žibert, J., Martinčić-Ipšić, S., Hajdinjak, M., Ipšić, I., and Mihelič, F. (2003) Development of a Bilingual Spoken Dialogue System for Weather Information Retrieval, *Proc. of the 8th European Conference on Speech Communication and Technology*, ISCA, Geneva, Switzerland, pp. 1917–1920.

[13] Hajdinjak, M. and Mihelič, F. (2003) The Wizard of Oz System for Weather Information Retrieval, *Lecture Notes in Artificial Intelligence 2807: Text, Speech and Dialogue*, pp. 400–405. Matoušek, V. and Mautner, P. (eds). Berlin, Springer.

[14] Hajdinjak, M. and Mihelič, F. (2004) Information-Providing Dialogue Management, *Lecture Notes in Artificial Intelligence 3206: Text, Speech and Dialogue*, pp. 595–602. Sojka, P., Kopecek, I. and Pala, K. (eds). Berlin, Springer.

[15] Walker, M.A., Litman, D., Kamm, C.A., and Abella, A. (1997) PARADISE: A General Framework for Evaluating Spoken Dialogue Agents, *Proc. of the 35th Annual Meeting of the Association of Computational Linguistics*, Association for Computational Linguistics, Madrid, Spain, pp. 271–280.

[16] Gros, J., Pavešić, N., and Mihelič, F. (1997) Text-to-Speech Synthesis: a Complete System for the Slovenian Language, *Journal of Computing and Information Technology*, University Computing Centre Zagreb, pp. **5(1)** 11–19.

[17] Walker, M.A., Litman, D.A., Kamm, C.A., and Abella, A. (1998) Evaluating Spoken Dialogue Agents with PARADISE: Two Case Studies, *Computer, Speech and Language*, Academic Press, pp. **12(3)** 317–347.

[18] Barras, C., Geoffrois, E., Wu, Z., and Liberman, M. (2001) Transcriber: Development and Use of a Tool for Assisting Speech Corpora Production, *Speech Communication: Special Issue on Speech Annotation and Corpus Tools*, Elsevier Science, pp. **33(1)** 5-22.

[19] Di Eugenio, B. and Glass, M. (2004) The Kappa Statistic: A Second Look, *Computational Linguistics*, The MIT Press, pp. **30**(1) 95–101.

Morpho-Syntactic Descriptions in MULTEXT-East — the Case of Serbian

Cvetana Krstev
Faculty of Philology, University of Belgrade
Studentski trg 3, 11000 Begrade, Serbia and Montenegro
cvetana@matf.bg.ac.yu

Duško Vitas
Faculty of Mathematics, University of Belgrade
Studentski trg 16, 11000 Begrade, Serbia and Montenegro
vitas@matf.bg.ac.yu

Tomaž Erjavec
Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Keywords: natural language processing, language resources, Serbian language, multilinguality

Received: July 21, 2004

MULTEXT-East is a multilingual dataset for language engineering research and development. This standardised and linked set of resources covers a large number of mainly Central and Eastern European languages and includes the EAGLES-based morphosyntactic specifications, defining the features that describe word-level syntactic annotations; medium scale morphosyntactic lexica; and annotated parallel, comparable, and speech corpora. The most important component is the linguistically annotated corpus consisting of Orwell's novel "1984" in the English original and translations. MULTEXT-East has already seen several editions, with the latest one being Version 3, where the most important addition are the Serbian language resources, including the structurally annotated "1984", the morphosyntactic specifications, the morphosyntactic lexicon and the linguistically annotated "1984". The complete dataset, unique in terms of languages and the wealth of encoding, is extensively documented, and freely available for research purposes.

Povzetek: članek opisuje uporabo MULTEXT-East v srbščini.

1 Introduction

The mid-nineties saw – to a large extent via EU projects – the rapid development of multilingual language resources and standards for human language technologies [6, 2]. However, while the development of resources, tools, and standards was well on its way for EU languages, there had been no comparable efforts for the languages of Central and Eastern Europe. The MULTEXT-East project (Multilingual Text Tools and Corpora for Eastern and Central European Languages) was a spin-off of the EU MULTEXT project [6]; MULTEXT-East ran from '95 to '97 and developed standardised language resources for six CEE languages [1], as well as for English, the 'hub' language of the project. The project also adapted existing tools and standards to these languages. The main results of the project were lexical resources and an annotated multilingual corpus. The most important resource turned out to be the parallel corpus — heavily annotated with structural and linguistic information — which consists of Orwell's novel "1984" in the English original and translations.

One of the objectives of MULTEXT-East has been to make its resources freely available for research purposes. In the scope of the TELRI concerted action (Trans European Language Resources Infrastructure), the results of MULTEXT-East had been extended with several new languages and first released on a CD-ROM, and later through Web download via TRACTOR, the TELRI Research Archive of Computational Tools and Resources.

The Serbian language did not have its representative in the MULTEXT-East project. The researchers from the Faculty of Mathematics, however, participated in the TELRI concerted action. One of the results of this participation was the Serbian "1984" structurally annotated corpus, but the morphosyntactic specification, lexicon and linguistically tagged "1984" were not produced.

Following the TELRI release, the MULTEXT-East resources were used in a number of studies and experiments. In the course of such work, errors and inconsistencies were discovered in the MULTEXT-East specifications and data, most of which were subsequently corrected. But because this work was done at different sites and in different manners, the encodings of the resources had begun to drift

apart.

The '98–'00 EU Copernicus project CONCEDE (Consortium for Central European Dictionary Encoding) offered the possibility to bring the versions back on a common footing. Although CONCEDE was primarily devoted to machine readable dictionaries and lexical databases, one of its workpackages did consider the integration of the dictionary data with the MULTEXT-East corpus. The CONCEDE release contained the revised and expanded morphosyntactic specifications, the revised lexica, and the significantly corrected and re-encoded linguistically annotated "1984" corpus.

In addition to delivering resources per-se, a focus of the MULTEXT-East, TELRI and CONCEDE projects was also the adoption and promotion of encoding standardisation. On the one hand, the morpholexical annotations and lexica were developed in the formalism of the (EAGLES-based) specifications for six Western European languages of the MULTEXT project [6]. On the other, in the TELRI edition, all the corpus resources were encoded in SGML, in CES, the Corpus Encoding Standard [5]. For the corpus taken forward into the second edition, the Text Encoding Initiative Guidelines were adopted, in particular TEI P3 [9].

Finally, in 2004 the third version of the MULTEXT-East resources was released [3]. This release offers several contributions: it brings together the first two, i.e., offers both the TELRI and CONCEDE versions in one package; all the resources have been recoded in XML, according to TEI P4 [10], thus enabling them for processing with XML-based tools; and resources for new languages have been added, in particular the morphosyntactic specification for Resian, a dialect of Slovene, and, crucially the morphosyntactic specification and the annotated Orwell for Serbian.

Version 3 also contains extensive documentation, e.g., navigational HTML pages, which serve to structure and link the resources, and which include the list of participants and indexes to the resource by type and language. While the TEI headers give the most precise and up-to-date information on the corpus components, the documentation also contains a bibliography with copies of the MULTEXT-East project reports (giving details of the resources, e.g., the corpus markup process), published papers, a mirror of the TEI P4 and CES documentation and certain related MULTEXT and EAGLES reports.

A complete description of the Version 3 resources is given in [3] and in the on-line documentation, while this paper concentrates on the Serbian resources. In the next section we introduce the structurally annotated Serbian "1984" (already a part of the TELRI release), in Section 3 we describe INTEX, the system that has for a long time served as the infrastructure for developing LR resources for Serbian, Section 4 explains the MULTEXT-East (Serbian) morphosyntactic specification, Section 5 the linguistically annotated "1984", Section 6 the Serbian lexicon and the last section gives some conclusions and direction for further work.

```
<text id="mteo-sr." lang="sr">
 <body id="Osr" lang="sh">
  <div id="Osr.1" n="1" type="part">
   <head>Prvi deo</head>
   <div id="Osr.1.2" n="1" type="chapter">
    <head>1.</head>
    <p id="Osr.1.2.2">
      <s id="Osr.1.2.2.1">Bio je vedar i
hladan aprilski dan; na &#x10D;asovnicima
je izbijalo trinaest.</s>
      <s id="Osr.1.2.2.2"><name>Vinston
Smit</name>, brade zabijene u nedra da
izbegne ljuti vetar, hitro zama&#x10D;e u
staklenu kapiju stambene zgrade
<hi rend="it">Pobeda</hi>, no nedovoljno
hitro da bi spre&#x10D;io jednu spiralu
o&#x161;tre pra&#x161;ine da u&#x111;e
zajedno s njim.</s>
    </p>
```

Figure 1: The structurally annotated Orwell

2 Structural "1984"

The MULTEXT-East multilingual parallel corpus consists of the novel "1984", about 100,000 words in length. The corpus contains extensive headers and markup for document structure, sentences, and various sub-sentence annotations, which have been harmonised over languages. As an example, the start of the text from the Serbian Orwell is given in Figure 1.

The translations of "1984" have been automatically sentence aligned with the English original, and the alignments hand-validated. The bilingual alignments are stand-off, i.e., they are stored not with the primary data but in separate documents, as references to sentence IDs.

The cesDoc encoded novel served as the basis for producing the linguistically annotated version. The link between the two is maintained via sentence identifiers.

The Serbian version was produced already in the scope of TELRI. The digital source was the same as for the English and Slovene versions, namely the Oxford Text Archive, via the ECI multilingual CD-ROM. This version was plain ASCII, so it was first marked up, similar to other versions in SGML, and then sentence segmented and aligned with the English original. Also, many typographical errors were corrected.

3 Serbian INTEX resources

Before discussing the MULTEXT-East Serbian morphosyntactic resources (the specifications, lexicon and linguistically annotated "1984") we first describe the basis for these resources, which had been developed independently of European projects, namely the Serbian morphological lexicon in the format of the INTEX system, which is based on the technology of finite-state transducers [8].

In this dictionary a lemma is of the form $W_t, W_l.Cn +$

$SSD : (Codes)*$ where W_t represents the textual word, W_l the corresponding lexical word, C is the part of speech, n is the code of inflective class, SSD is the set of syntactic and semantic attributes of the lemma that is classified as Cn, and $codes$ describe the values of morphological categories that realized with the form W_t. For instance, the dictionary entry `prozorom,prozor.N01+Com:ms6q` descibes the form *prozorom* as the form of *prozor* (Engl. window), which is common (+Com) musculine (m) inanimate (q) noun (N) from the inflective class 01 in instrumental (6) of singular (s). It can be seen that this format is not as compact as MSD, as the relevant information is distributed among the inflective code, syntactic and semantic information, both associated to lexical word, and the grammatical codes which are assigned to the textual word. The $codes$ are not positional — for a part of speech one alphanumeric character represents a value of one and only one of its attributes.

The present size of the Serbian morphological dictionary is 74,000 lemmas and more than 1 million word forms, which enables morphological text analysis with a high percentage of success, around 92% for literary texts. Some of the word-forms that are not covered by the dictionary itself can be successfully morphologically tagged by additional tools (lexical transducers) incorporated in Intex. The use of this specifically constructed set of lexical transducers enables the recognition of various derived word-forms, such as several classes of compounds, possessive adjectives, diminutives, augmentatives, etc. [7].

The team from the University of Belgrade plans to convert its full INTEX lexicon to a MSD-type lexicon. It is also planned to tag with MSDs the corpus of contemporary Serbian that is being developed at the Faculty of Mathematics [12], where it plans to use INTEX and the lexica incorporated in it as a preprocessor.

4 Morphosyntactic Specifications

The MULTEXT-East morphosyntactic specifications give the syntax and semantics of the morphosyntactic descriptions (MSDs) used in the lexica and corpora. The MSDs, are structured and more detailed than is commonly the case for part-of-speech tags; they are compact string representations of a simplified kind of feature structures. The first letter of a MSD encodes the part of speech, e.g., `Noun` or `Adjective`. The letters following the PoS give the values of the position determined attributes. The specifications define, for each part of speech, its appropriate attributes, their values and one-letter codes. So, for example, `Ncmpi` expands to `PoS:Noun, Type:common, Gender:masculine, Number:plural, Case:instrumental`. It should be noted that in case a certain attribute is not appropriate (1) for a language, (2) for the particular combination of features, or (3) for the word in question, this is marked by a hyphen in the attribute's position. Slovene verbs in the indicative, for example, are not marked for gender or voice, hence the two hyphens in `Vcip3s--n`.

The specifications have been developed in the formalism and on the basis of specifications of the EU MULTEXT project [6] and in cooperation with EAGLES, the Expert Advisory Group on Language Engineering Standards. Originally, these specifications were released as a report of the MULTEXT-East project but have been revised for both subsequent releases, and have become, if not a standard, then at least a reference for comparison [4].

The MULTEXT-East morphosyntactic specifications have the following structure: (1) introductory matter; (2) the common specification; and (3) a language particular section for each language.

The common part of the specifications first defines the parts of speech and their codes; MULTEXT-East distinguishes the following, where not all PoS are used for all languages - we mark in italic those that are not used for Serbian: Noun (N), Verb (V), Adjective (A), Pronoun (P), *Determiner (D)*, *Article (T)*, Adverb (R), Adposition (S), Conjunction (C), Numeral (M), Interjection (I), *Residual (X)*, Abbreviation (Y), and Particle (Q).

The formal core of the specifications resides in the common tables; they define the features, their codes for MSD representation, and their appropriateness for each language — an example is given in Figure 2.

Technically, the complete specifications are a LaTeX document (with derived HTML and PDF renderings), where the common tables are plain ASCII in a strictly defined format. This format is suitable for a printed version, tolerable for one in HTML, and reasonably manageable for modification and addition of new languages. However, it is not suitable for processing needs, in particular to enable smooth manipulation and linking to an XML encoded corpus using the MSDs.

We have therefore implemented a (Perl) conversion of the common tables into XML, using the TEI.fs module, a tagset devoted to encoding feature-structures. This tagset is currently being used as the basis of an evolving ISO standard (currently a Draft International Standard), as part of work of ISO/TC 37/SC4 Language Resource Management.

The XML version of the common tables has one feature library for each category, e.g., ⟨*fLib type="Noun"*⟩. Each feature in such a library is comprised of the identifier, which enables the linkage to corpus MSDs, the name of the attribute, the languages the feature is appropriate for, and the symbol that is its value; examples are given in Figure 3.

The Serbian specifications was produced on the basis of the Croatian one (which was added in the scope of CONCEDE), with some modifications stemming less from the differences between languages and more by the set of morphosyntactic attributes already incorporated in the Intex e-dictionaries for Serbian. For instance, in the verb table the value gerund for the attribute VForm is the most appropriate to account for present and past gerund active in Serbian. Also, the attribute Clitic is applicable to Serbian

3.2 Verb (V)

P	ATT	VAL	C	EN	RO	SL	CS	BG	ET	HU	HR	SR	SL-ROZAJ
1	Type	main	m	x	x	x	x	x	x	x	x	x	x
		auxiliary	a	x	x	x	x	x	x	x	x	x	x
		modal	o	x	x	x	x		x		x	x	x
		copula	c		x	x	x				x	x	x
		base	b	x									
2	VForm	indicative	i	x	x	x	x	x	x	x	x	x	x
		subjunctive	s		x								x
		imperative	m		x	x	x	x	x	x	x	x	x
		conditional	c	x		x	x		x	x	x	x	
		infinitive	n	x	x	x	x		x	x	x	x	x
		participle	p	x	x	x	x	x	x		x	x	x
		gerund	g		x			x	x			x	
		supine	u			x			x				x
		transgressive	t				x						
		quotative	q						x				

Figure 2: Start of Common Table for Verbs

```
<fLib type="Verb">
<f id="V0."
   select="en ro sl cs bg et hu hr sr sl-rozaj"
   name="PoS"><sym value="Verb"/></f>
<f id="V1.m"
   select="en ro sl cs bg et hu hr sr sl-rozaj"
   name="Type"><sym value="main"/></f>
<f id="V1.a"
   select="en ro sl cs bg et hu hr sr sl-rozaj"
   name="Type"><sym value="auxiliary"/></f>
<f id="V1.o"
   select="en ro sl cs et hr sr sl-rozaj"
   name="Type"><sym value="modal"/></f>
```

Figure 3: Morphosyntactic specifications as TEI features

copula verbs, as well as the attribute Aspect to the most of the other verbs. Both these attributes are already encoded for all verbs in Serbian e-dictionary. One of the other differences between Croatian and Serbian tables is the recognition of the value ‘paukal’ for the attribute ‘Number’ for several PoS in Serbian.

5 Lexicons

The MULTEXT-East morphosyntactic lexicons have a simple structure, where each lexical entry is composed of three fields:

1. the *word-form*, which is the inflected form of the word, as it appears in the text, modulo sentence-initial capitalisation;
2. the *lemma*, which is the base-form of the word; where the entry is itself the base-form, the lemma is given as the equal sign; and
3. the *MSD*, the morphosyntactic description.

To produce the lexica, the token lists of the MULTEXT-East corpus were first fed through morphological analysers in order to produce the lemma list; this list was further extended from the comparable corpus, to arrive at at least 15,000 lemmas – some languages have further extended this, e.g., Romanian to 41,000 lemmas. In the next step, these lemmas were fed back to morphological generators (except for the agglutinative languages) in order to produce the complete inflected lists, i.e., the full paradigms of the lemmas, which constituted the final lexica of the project.

The MULTEXT-East lexica serve as medium sized morphological lexica for the languages. In addition to explicating the inflectional behaviour of the most common (and, typically, morphologically the most complex) words of the languages, the lexica also serve to establish the definitive set of valid MSDs for the languages.

For Serbian, currently, only a minimal lexicon was produced, which contains just the word-forms that in fact appear in the annotated “1984” corpus. This lexicon has 20,294 entries, 16,907 different word-forms, 8,392 lemmas and 906 MSDs.

To serve as a standard registry of MSDs, we converted the lexical MSDs to TEI feature structure libraries, ⟨*fsLib*⟩, one for each category. Here each MSD is expressed as a feature structure specifying its *id*, the language(s) it is appropriate for, and its decomposition into features. Some examples are given in Figure 4.

Both ⟨*fsLib*⟩s and ⟨*fLib*⟩s are stored in dedicated ⟨*TEI.2*⟩ element, complete with its TEI header; this document also constitutes a part of the linguistically annotated MULTEXT-East corpus.

```
<fsLib type='Verb'>
<fs id="Van" select="en et"
    feats="V0. V1.a V2.n"/>
<fs id="Van----an----n" select="cs"
    feats="V0. V1.a V2.n V7.a V8.n V13.n"/>
<fs id="Van----an-n---p" select="sr"
    feats="V0. V1.a V2.n V7.a V8.n V10.n
           V14.p"/>
<fs id="Van----ay----n" select="cs"
    feats="V0. V1.a V2.n V7.a V8.y V13.n"/>
<fs id="Vanp" select="ro"
    feats="V0. V1.a V2.n V3.p"/>
```

Figure 4: MSDs as TEI feature structures

6 Linguistically annotated "1984"

The centrepiece of the MULTEXT-East resources is the linguistically annotated "1984"; it contains word level markup, namely context disambiguated lemmas and MSDs. Because it was the first such resources for many of the MULTEXT-East languages, also Serbian, it was the most difficult and time-consuming to produce as the work had to proceed mostly manually. The annotated novel is useful as a dataset for tagger and lemmatiser induction and testing, and has already been used for this purpose in a number of experiments; c.f. the bibliography section on the MULTEXT-East web site.

The work on the Serbian annotation proceeded in the following steps. First, using Intex as the tool and all of the Serbian lexical resources "1984" was morphologically tagged. As a result, a textual file is obtained that contains the finite automaton of the text represented in the form of a regular expression.

In the second step the text annotated in this way was manually checked and disambiguated. It means that not only have right lemmas and morphological categories been chosen for ambiguous word-forms and added for those words that had not been recognized, but non-ambiguous forms have also been checked in case that they had been incorrectly recognized.

As a result, a non-ambiguous representation of a text is obtained in the same format. This step had been done iteratively, which enabled both the correction of the used dictionaries and other lexical resources, and their enhancement.

In the third step, a Perl script was written and used to convert the Intex annotated text with to the MULTEXT-East annotation. The conversion is not, however, a straightforward task, not only because of the different encoding systems, as described in section 3, but also because of the differently chosen attributes. This difference can be most easily described in the case of verbs. For verbs, in MULTEXT-East the second attribute specifies a verb form, and the third a tense. However, due to the composite tenses, some verb forms are used for the construction of different tenses. For instance, in Serbian, verb form *imao* is the active past participle of the verb *imati* (Engl. *to have*), and is used to produce both perfect tense if used with the indicative form of the present tense of the copula verb *biti* . In Intex, however, only the verb forms are recognized, which in the case of simple tenses enables the recognition of a tense as well (for instance, for present or aorist). For analytical tenses, the word-form recognition is not enough and more complex tools have to be used, as described in [11]. These tools, as not yet being fully developed, were not used in the first step of the annotation process, and thus the precise mapping from Intex to MULTEXT-East tags was not possible. As a consequence, despite having different functions, the active past participle is always given the same value in the third attribute, that is `tense=past`, as being the most frequent.

The TEI P4 markup of the linguistically annotated Serbian "1984" obtained through this process is exemplified in Figure 5 by the same two sentences.

```
<s id="Osr.1.2.2.1">
<w lemma="biti" ana="Vmps-sman-n---p">Bio</w>
<w lemma="jesam" ana="Va-p3s-an-y---p">je</w>
<w lemma="vedar" ana="Afpmsnn">vedar</w>
<w lemma="i" ana="C-s">i</w>
<w lemma="hladan" ana="Afpmsnn">hladan</w>
<w lemma="aprilski" ana="Aopmpn">aprilski</w>
<w lemma="dan" ana="Ncmsn--n">dan</w>
<c>;</c>
<w lemma="na" ana="Spsa">na</w>
<w lemma="&#x10D;asovnik"
   ana="Ncmsa--n">&#x10D;asovnicima</w>
<w lemma="jesam" ana="Va-p3s-an-y---p">je</w>
<w lemma="izbijati"
   ana="Vmps-snan-n---e">izbijalo</w>
<w lemma="trinaest" ana="Mc---l">trinaest</w>
<c>.</c>
</s>
```

Figure 5: The linguistic annotation of "1984"

7 Conclusions

The paper presented Version 3 of the MULTEXT-East resources, and, in particular, its Serbian language portion. As the resources cover a number of inflectionally rich languages, are interlinked, harmonised, have a standardised encoding, and have been manually validated and tested in practice, they can serve as a "gold standard" dataset for language technology research and development.

While portions of the resources are distributed without any restrictions, the resources as a whole are available free of charge for research purposes only, as this was the condition imposed by some copyright holders of the sources.

Version 3 of the resources can be downloaded from the MULTEXT-East home page, *http://nl.ijs.si/ME/*. Access is enabled by filling out and submitting a Web based agreement, which is modelled after the one used by Edinburgh's Language Technology Group.

Currently, there are no plans to start working on Version 4; rather, the focus will be on the utility of V3, in our own

research, and in enabling others to use the resources, by providing maintenance, continuing to support their accessibility and correcting errors.

Acknowledgments

The work presented in this paper was, in part, supported by the bi-lateral project on scientific and technological cooperation between Slovenia and Serbia “The development of language resources for machine translation between the Slovene and Serbian languages”.

References

[1] Dimitrova, L., Erjavec, T., Ide, N., Kaalep, H.J., Petkevič, V., and Tufiş, D., 1998. Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In *COLING-ACL ’98*. Montréal, Québec, Canada.

[2] EAGLES, 1996. Expert advisory group on language engineering standards. Http://www.ilc.pi.cnr.it/EAGLES/home.html.

[3] Erjavec, T., 2004. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In *Fourth International Conference on Language Resources and Evaluation, LREC’04*. Paris: ELRA. Http://nl.ijs.si/et/Bib/LREC04/.

[4] Erjavec, T., Krstev, C., Petkevič, V., Simov, K., Tadić, M., and Vitas, D., 2003b. The MULTEXT-East Morphosyntactic Specifications for Slavic Languages. In *Proceedings of the EACL 2003 Workshop on Morphological Processing of Slavic Languages*. Budapest.

[5] Ide, N., 1998. Corpus Encoding Standard: SGML guidelines for encoding linguistic corpora. In *First International Conference on Language Resources and Evaluation, LREC’98*. Granada: ELRA. Http://www.cs.vassar.edu/CES/.

[6] Ide, N., and Véronis, J., 1994. 1994. Multext (multilingual tools and corpora). In *Proceedings of the 15th International Conference on Computational Linguistics*. Kyoto.

[7] Pavlović-Lažetić, G., Vitas, D., and Krstev, C., 2004. Towards full lexical recognition. In *Text, Speech and Dialogue*, Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, pp.179–186.

[8] Silberztein, M., 2000. *INTEX*. Masson.

[9] Sperberg-McQueen, C. M. and Burnard, L. (eds.), 1999. *Guidelines for Electronic Text Encoding and Interchange, Revised Reprint*. The TEI Consortium.

[10] Sperberg-McQueen, C. M. and Burnard, L. (eds.), 2002. *Guidelines for Electronic Text Encoding and Interchange, The XML Version of the TEI Guidelines*. The TEI Consortium.

[11] Vitas, D., 2003. Composite tense recognition and tagging in serbian. In *Proceedings of the EACL 2003 Workshop on Morphological Processing of Slavic Languages*. Budapest.

[12] Vitas, D., Krstev, C., Pavlović-Lažetić, G., and Obradović, I., 2003. An Overview of Resources and Basic Tools for Processing of Serbian Written Texts. In *Proc. of the Workshop on Balkan Language Resources, 1st Balkan Conference in Informatics*.

Computer Education and Social Changes in Slovenia

Franci Pivec
IZUM Maribor,
e-mail: franci.pivec@izum.si

Vladislav Rajkovič
FOV Kranj,
e-mail: vladislav.rajkovic@ijs.si

Andrej Jus
Infos Ljubljana,
e-mail: andrej.jus@infos.si

Keywords: WCC; computer education; computer programming competition

Received: May 25, 2004

The World Computer Congress (WCC) held in Ljubljana in 1971 played a very important role in the promotion of computer science. Slovenian school authorities enjoyed relative autonomy in the former Yugoslavia and this made it possible for them to promote computer education. Informal computer education and the introduction of computer science subjects into regular school curriculum started very early in the 1970s. Strong support from civil society was of significant importance in that process. This expressed orientation towards an information society was one of the major differences between Slovenia and the rest of the former Yugoslavia and one of the causes contributing to the attainment of Slovenia's independence.

Povzetek: članek obravnava uvajanje računalništva v slovenski izobraževalni sistem in njegov vpliv na družbene spremembe.

1 Introduction

The history of computer education is almost half a century long. Consequently, written records documenting its beginnings and subsequent development have been growing in number. At the 18th World Computer Congress, held in August 2004 in Toulouse, the first Conference on the History of Computing in Education was organised; it seemed worthwhile, as well as necessary, to present the Slovenian experience there. (Impagliazzo&Lee 2004) This contribution recapitulates, to the largest extent possible, that presentation so that it can be verified and upgraded.

The former Yugoslavia, a socialist country, was determined to demonstrate its progressiveness, among other things, in science. Consequently, large scientific and technology centres were built in the three main university cities as innovative cores capable of being competitive in the international level. The centre in Ljubljana, Slovenia, had decided to purchase a Zuse Z-23 computer as early as the end of the 1950s. This was actually the beginning of computing in Slovenia (The pioneer times of computing in Slovenia are dealt with in a special round table discussion within the Slovenian Informatics Society, web page: http://hist-ri.slo.net.) Computer science was considered to be a field of strategic national importance. The applied mathematics team in Ljubljana was well aware of the significance and attractiveness of the new technology; thanks to their efforts, Jožef Stefan Institute (Institut Jožef Stefan) became one of the first computer education centres in the world with an incredibly progressive motto stating that all people, aged 3 to 73, with an interest in the new technology were welcome.

2 WCC 1971 in Ljubljana

Ljubljana was the only city "on the other side of the Iron Curtain" where the IFIP World Computer Congress was held and that as early as 1971, and those facts had far-reaching consequences. Slovenian computer experts established excellent connections with highly developed environments. As one of the consequences of the IFIP World Computer Congress, the use of computers became the imperative of progress in Slovenian public opinion. The largest computer corporations concluded, due in part to the influence of the WCC, that Ljubljana was a suitable location for establishing their representative offices for doing business with the "East"; we can mention Intertrade as an example in this respect. Soon, they began to license the manufacture of computer equipment (ISKRA, Elektrotehna, DELTA, GORENJE, etc.), on which special export restrictions had actually been imposed in the countries of origin.

The chronology of events leads to the conclusion that the WCC had a large impact on computer education in Slovenia. In the beginning of the 1970s, the first major decisions which made the introduction of the new technology into schools possible were reached.

Today it can be estimated that the attitude towards computer science as the technology of the future—leading to changes also in the social organisation and with a great impetus from the WCC—was one of the delicate differences between Ljubljana and Belgrade. As a result, categorical demands for a change in the direction of development increased. On the basis of information technology, Slovenia defended modern managerial approaches and a complete openness to the world. The opposition to governing practices and to the anti-technocratic policy of the federal authorities was becoming increasingly evident.

3 The beginnings of computer education

At the end of the 1960s, computer science was introduced into schools in a rather informal way. Young employees of the computer centre at the Jožef Stefan Institute (Bratko, Rajković, Lajovic, Hafner, Trampuž, Cokan) offered to conduct optional computer courses in various secondary schools. Such courses, most of them consisting of approximately 40 lessons, usually included elementary computer functions and basic programming in FORTRAN. Fairly quickly, these activities received a favourable reaction from the Slovenian Board of Education under the leadership of B. Lipužič and, consequently, the Commission for the Introduction of Computer Education into Secondary Schools was established. In education circles, France Strmčnik (1972) and Barica Marentič-Požarnik (1972) were of special importance; as early as the beginning of the 1970s, they had started to prepare teachers for computer-assisted teaching and to arouse their enthusiasm for it. In view of the declarations by the Slovenian computer "avant-garde", it is appropriate to mention here also that co-operation with the Multimedia Centre of the Referral Centre of the University of Zagreb was of considerable importance. There Mužić and Šoljan systematically researched the use of computers in education.

In 1971, the first seminar for computer science teachers was held. (Bratko et al 1972) At that time computer science was taught, in one way or another, in a quarter of all Slovenian secondary schools. In 1974, the first Slovenian computer science textbook was published. (Bratko&Rajković 1974) One year later, when computer science was being taught at a half of all Slovenian schools, the first evaluation was carried out, and the report produced was presented at the IFIP (International Federation of Information Processing) Computer in Education Conference. (Bratko, Rajković, Roblek 1975) In 1973, the International Conference on the Use of Computers in Chemical Education was held in Ljubljana. The conference was led by Aleksandra Kornhauser, and it represented the basis for the establishment of the UNESCO International Centre for Chemical Studies (ICCS). (Kornhauser 1975) In 1974, a symposium on computer in education was organised at Jožef Stefan Institute. It triggered an enthusiastic reaction, and a number of interesting presentations were delivered, e.g., those by V. Bufon (1974), P. and S. Tancig (1974) and B. Marentič-Požarnik. (1974)

Not many schools had their own computers at that period of time. Commercial and public organisations with computer equipment in their possession made terminal or direct access available to schools. This kind of co-operation, in most cases free of charge and with a generous support provided by computer experts, though few in number at the time, was undoubtedly something uniquely Slovenian; it confirmed the then wide-spread conviction that computer technology was crucial to the future of Slovenia. Within the special interest community for computer education, schools and future employers, hand in hand, took care to improve computer equipment in schools. The ideological décor of the time put aside, the imaginative approach in Slovenia engendered positive surprise among experts.

The emergence of microcomputers was the beginning of radical changes. According to I. Gerlič (2000), this period of time can be divided into two phases:

- the multiform phase, when schools purchased all types of computers available (Comodore, Sinclair, Apple, BBC, ORIC, etc.) and
- the uniform phase, after the decision was made to purchase only IBM-PC compatible computers with the uniform MS-DOS operating system.

Commodore and Sinclair triggered such enthusiasm among the first computer generations ("X generations") that they would strongly deserve to have memorial tablets erected in Slovenian schools. In 1983, the first regulation governing hardware and software standardization in schools was issued. Also in the following years, the import of personal computers represented a special chapter in the development. Yugoslavia belonged to the group of states which hindered the proliferation of computer equipment outside controlled central systems.

This was also opposed by Slovenia at the official level; in October 1984, the then Slovenian Executive Council submitted the "Information on the Current Situation in the Field of Computer Literacy" (IS-SRS 1984) to the Parliament. That document had been drawn up also with the participation of the authors of this article. Among the delegates, Erik Vrenko and Ciril Baškovič from the then Republic Committee for Research and Technology were the ones who defended this document. The document conflicted with the strictly confidential (!) federal regulations banning the import of home computers; in this respect it referred to the public appeal which had been previously published by the Republic Conference of the Association of Socialist Youth of Slovenia.

Then computer education was introduced as an optional subject in primary schools and as a range of compulsory subjects in all secondary schools. A Computer Science

Secondary School established in 1981 in the framework of a school reform—which, however, eventually proved to be rather unsuccessful—was a Slovenian phenomenon. Almost 2% of the total school population enrolled, which clearly indicated that this school was among the more popular ones. This confirmed the euphoric openness to the new technology and the great expectations Slovenian society had for it.

In the 1980s, the efforts for computer education intensified. A new textbook with the title "Computer Science with a Collection of Exercises" (Benkovič et al 1980) was published. At that time, in publications, a connection between computer education and an information society was mentioned; this was, at that time, in 1983, also rare worldwide. (Rajković&Kušče-Zupan 1983) There were contributions on educational software (Batagelj 1986) and on the necessity to plan how to supply schools with hardware. (Sovič 1987) Vladislav Rajkovič and Tomaž Skulj produced a study with the title: "How to Proceed with the Proliferation of Computer Literacy in our Schools" (Rajković&Skulj 1987) which helped present Slovenia as a country highly developed in this field at the Yugoslav Conference on Computers in Education held in 1987 in Nova Gorica. (ZŠ 1987)

The first university computer course was introduced in 1973; electrical engineering and mathematics students could enrol in it after their second year of study. Soon, computer science spread to other fields of study and also achieved the status of an independent university programme. However, the development of this deserves more attention and is not dealt with in this contribution. It is of special significance that Education Faculties introduced computer education for teachers as early as 1984. It is of the utmost importance that the state supported this development trend with a strong information infrastructure (www.arnes.si), which was accessible to everyone participating in the education process, free of charge, and with a very popular co-operative online public library service (www.cobiss.si) in which Slovenian libraries have been participating since the end of the 1980s. Regarding the usage frequency of the abovementioned infrastructure, Slovenia is at the top of EU countries. (SIBIS 2003)

4 The civil movement for the use of computers

The Association of Organisations for Technical Culture with its widespread network and the goal of popularizing technical innovations, and non-governmental organisations played a large, sometimes even essential, role in the introduction of computer education in Slovenia. In the 1970s, the use of computers became their major focus. Throughout Slovenia, they organised computer workshops and exhibitions (such as "Računalnica" and "Računalniški dnevi"), which represented a meeting-point for hardware and software providers and users, the latter ones consisting of teachers and pupils united in their common effort to master the new technology. As a consequence, computer clubs emerged and these were to great assistance to computer education. The following activities also deserve to be mentioned: the organisation of visits to computer equipment producers, summer schools of computer science, modelled on youth research camps, and publishing. Popular presentations on computer science appeared in all types of media, from the Ciciban newspaper to the Radio Študent radio programme. Freely accessible software was provided. At that time, the idea of specialized computer newspapers, BIT and Moj mikro, was conceived. Nowadays an exhibition referred to as Information Technology in Slovenia (Informatika na Slovenskem), or INFOS for short, reminds one of past events in this field. (Skulj 2003)

Programming competitions were particularly popular; the first ideas concerning these competitions can be traced back to 1974, earlier than anywhere else in the world, and the first competition was organised in 1977 (Hafner 1977). Initially, these competitions were targeted towards secondary schools but later also to primary schools including first grade pupils. Without a doubt, P. Azalov is wrong when he claims that Bulgarian computer competitions were the first ones of this kind in the world; they were held no earlier than 1982, and, by that time, Slovenia already had a long tradition in the field. (Azalov 1989) The highlight of these competitions was the first International Computer Olympics, held in 1988 in Nova Gorica, Slovenia. Azalov has also ignored this fact and claims that the first Computer Olymics were held in 1989 in the city of Pravec in Bulgaria. (Azalov 1994) Fortunately, the Olympics held in Nova Gorica, have been assuredly documented with the publication of its "Problems in Programming" by the Wiley Publishing House. (Vitek et al 1991)

5 Conclusion

In the former Yugoslavia, the introduction of ICT (information and communications technology) became a field of confrontation between the different concepts of social development. The field of computer science symbolised an openness to the world, good management practices and ideological neutrality. From this aspect, its significance was similar to "perestrojka", "solidarnošč" or "svoboda vyrazu", which occurred much later in other socialist countries. ICT also had an impact on the radical changes taking place in the former Yugoslavia. The meteoric rise of computer education in Slovenia in the 1970s and 1980s undoubtedly contributed to the historic changes in Slovenia. Based on this was the idea that the existence of a small nation like Slovenia no longer depended on the "shelter" of some powerful state but on the creative participation in the global network of the information society. After the attainment of its independence, Slovenia became a member of the EU and NATO; however, it is of an even greater importance that it adopted its own strategy of transition into an information society. In 2003, ITU (International Telecommunication Union), on the basis of DAI (Digital

Access Index) ranked Slovenia among 25 highly developed countries and proclaimed it as "an early adopter of IT" (ITU 2003). This gave the greatest satisfaction possible to the large number of people who had devoted their efforts to computer education in the last four decades. The objective of Slovenia's attainment of independence was not to become isolated but to make stronger and more advanced links with the rest of the world. That is what can be pointed to as the very essence of the computer education paradigm from its very beginning and onwards.

6 Reference

[1] Azalov, P. (1989) Bulgarian competitions in informatics. Matematics Competition, 2 (1); 60–66.
[2] Azalov, P. (1994) IOI'93 – To Mendoza and back. M&IQ, 3(4); 122–131.
[3] Batagelj, V. (1986) Izobraževalna programska oprema. Vzgoja in izobraževanje, 1(5).
[4] Benkovič, J. in dr. (1980) Računalništvo: Zbirka nalog. Ljubljana:DZS
[5] Bratko, I., J. Grad, M. Kac, J. Lesjak, J. Virant, E. Zakrajšek (1972) Računalništvo (Gradivo za srednješolske profesorje). Ljubljana: Zavod SRS za šolstvo.
[6] Bratko, I., V. Rajković (1974) Uvod v računalništvo. Ljubljana: DZS.
[7] Bratko, I., V. Rajković, B. Roblek (1975) What should secondary school students know about computers: analysis of an experiment. V: O. Lecarne, R. Lewis (ed.) Computer in education. IFIP, North Holland Publishing, 841–846.
[8] Bufon, V. in dr. (1974) Computer managed achievement tests using the programme package KOLOK. Zbornik Informatica '74. Ljubljana: IJS.
[9] Gerlič, I. (2000) Sodobna informacijska tehnologija v izobraževanju. Ljubljana: DZS.
[10] Hafner, I. (1977) Prvo republiško tekmovanje iz računalništva. Delo, 21.april; 7.
[11] Impagliazzo, J., Lee, J.A.N. (ed.) (2004) History of computing in education. IFIP 18th WCC, TC3/TC9 1st Conference on the history of computing in Education, 22–27 August 2004, Toulouse. Boston: Kluwer Academic Publishers.
[12] ITU (2003) ICT access categories: how are economies around the world doing? ITU News, 10 (Dec.); 14–17.
[13] Izvršni svet SRS (1984) Informacija o stanju na področju računalniške pismenosti v SR Sloveniji (ESA – 558). Priloga Poročevalca, 9/10 1984, 8–14.
[14] Kornhauser, A. (1975) Uporaba računalnika v kemijskem izobraževanju. Vzgoja in izobraževanje, 5; 3–24.
[15] Marentič-Požarnik, B. (1972) Vloga računalniške tehnike v poučevanju. Naši razgledi, XXI, 10.
[16] Marentič-Požarnik, B. (1974) Can computor help us to improve instruction? Zbornik Informatica '74. Ljubljana: IJS.
[17] Rajkovič, V., Kušče-Zupan, S. (1983) Izobraževanje za prehod v informacijsko družbo. Anthropos, 5–6, 274–286.
[18] Rajkovič, V., Skulj, T. (1987) Kako naprej od računalniškega opismenjevanja v naših šolah? Prispevek za posvet "Idejno sporočilo na pragu inovacijske družbe" (Arhiv avtorjev).
[19] SIBIS (2003) Measuring the Information Society in the EU. Bonn: Empirica.
[20] Skulj, T. (2003) Ostrenje osti in pršenje na vse strani. Povezave, November; 6–7.
[21] Sovič, B. (1987) Priporočila v zvezi z opremljanjem šol z računalniki: strojna oprema (hardware). Vzgoja in izobraževanje, 2(1).
[22] Strmčnik, F. (1972) Kibernetična smer programiranega pouka. Sodobna pedagogika, 7–8.
[23] Tancig, P., Tancig, S. (1974) Uporaba računalnika pri konstrukciji testov znanja in pri obdelavi rezultatov. Zbornik Informatica '74. Ljubljana: IJS.
[24] Vitek, A., I. Tvrdy, K. Reinhart, B. Mohar, M. Martinec, T. Dolenc, V. Batagelj (1991) Problems in programming. Experience through practice. Chicester: John Wiley&Sons.
[25] Zavor SRS za šolstvo (1987) Bilten II. jugoslovanske konference o politiki modernizacije izobraževalne tehnologije: Računalnik v izobraževanju. Nova Gorica.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of S**love**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

In the last year on the site of the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

At the present time, part of the Institute is being reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of Economic Relations and Development, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

CONTENTS OF *Informatica* Volume 28 (2004) pp. 1–441

Papers

Editorials

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please submit three copies of the manuscript with good copies of the figures and photographs to one of the editors from the Editorial Board or to the Contact Person. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible directly on the manuscript, from typing errors to global philosophical disagreements. The chosen editor will send the author copies with remarks. If the paper is accepted, the editor will also send copies to the Contact Person. The Executive Board will inform the author that the paper has been accepted, in which case it will be published within one year of receipt of e-mails with the text in Informatica LaTeX format and figures in `.eps` format. The original figures can also be sent on separate sheets. Style and examples of papers can be obtained by e-mail from the Contact Person or from FTP or WWW (see the last page of Informatica).

Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the Contact Person.

QUESTIONNAIRE

☐ Send Informatica free of charge

☐ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1111 Ljubljana, Slovenia.

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than ten years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering the European computer science and informatics community - scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: .....

Title and Profession (optional): .....

.....

Home Address and Telephone (optional): .....

.....

Office Address and Telephone (optional): .....

.....

E-mail Address (optional): .....

Signature and Date: .....

Informatica WWW:

http://ai.ijs.si/informatica/

Referees:

Witold Abramowicz, David Abramson, Adel Adi, Kenneth Aizawa, Suad Alagić, Mohamad Alam, Dia Ali, Alan Aliu, Richard Amoroso, John Anderson, Hans-Jurgen Appelrath, Iván Araujo, Vladimir Bajič, Michel Barbeau, Grzegorz Bartoszewicz, Catriel Beeri, Daniel Beech, Fevzi Belli, Simon Beloglavec, Sondes Bennasri, Francesco Bergadano, Istvan Berkeley, Azer Bestavros, Andraž Bežek, Balaji Bharadwaj, Ralph Bisland, Jacek Blazewicz, Laszlo Boeszoermenyi, Damjan Bojadžijev, Jeff Bone, Ivan Bratko, Pavel Brazdil, Bostjan Brumen, Jerzy Brzezinski, Marian Bubak, Davide Bugali, Troy Bull, Sabin Corneliu Buraga, Leslie Burkholder, Frada Burstein, Wojciech Buszkowski, Rajkumar Bvyya, Giacomo Cabri, Netiva Caftori, Particia Carando, Robert Cattral, Jason Ceddia, Ryszard Choras, Wojciech Cellary, Wojciech Chybowski, Andrzej Ciepielewski, Vic Ciesielski, Mel Ó Cinnéide, David Cliff, Maria Cobb, Jean-Pierre Corriveau, Travis Craig, Noel Craske, Matthew Crocker, Tadeusz Czachorski, Milan Češka, Honghua Dai, Bart de Decker, Deborah Dent, Andrej Dobnikar, Sait Dogru, Peter Dolog, Georg Dorfner, Ludoslaw Drelichowski, Matija Drobnič, Maciej Drozdowski, Marek Druzdzel, Marjan Družovec, Jozo Dujmović, Pavol Ďuriš, Amnon Eden, Johann Eder, Hesham El-Rewini, Darrell Ferguson, Warren Fergusson, David Flater, Pierre Flener, Wojciech Fliegner, Vladimir A. Fomichov, Terrence Forgarty, Hans Fraaije, Stan Franklin, Violetta Galant, Hugo de Garis, Eugeniusz Gatnar, Grant Gayed, James Geller, Michael Georgiopolus, Michael Gertz, Jan Goliński, Janusz Gorski, Georg Gottlob, David Green, Herbert Groiss, Jozsef Gyorkos, Marten Haglind, Abdelwahab Hamou-Lhadj, Inman Harvey, Jaak Henno, Marjan Hericko, Henry Hexmoor, Elke Hochmueller, Jack Hodges, Doug Howe, Rod Howell, Tomáš Hruška, Don Huch, Simone Fischer-Huebner, Zbigniew Huzar, Alexey Ippa, Hannu Jaakkola, Sushil Jajodia, Ryszard Jakubowski, Piotr Jedrzejowicz, A. Milton Jenkins, Eric Johnson, Polina Jordanova, Djani Juričič, Marko Juvancic, Sabhash Kak, Li-Shan Kang, Ivan Kapustøk, Orlando Karam, Roland Kaschek, Jacek Kierzenka, Jan Kniat, Stavros Kokkotos, Fabio Kon, Kevin Korb, Gilad Koren, Andrej Krajnc, Henryk Krawczyk, Ben Kroese, Zbyszko Krolikowski, Benjamin Kuipers, Matjaž Kukar, Aarre Laakso, Sofiane Labidi, Les Labuschagne, Ivan Lah, Phil Laplante, Bud Lawson, Herbert Leitold, Ulrike Leopold-Wildburger, Timothy C. Lethbridge, Joseph Y-T. Leung, Barry Levine, Xuefeng Li, Alexander Linkevich, Raymond Lister, Doug Locke, Peter Lockeman, Vincenzo Loia, Matija Lokar, Jason Lowder, Kim Teng Lua, Ann Macintosh, Bernardo Magnini, Andrzej Małachowski, Peter Marcer, Andrzej Marciniak, Witold Marciszewski, Vladimir Marik, Jacek Martinek, Tomasz Maruszewski, Florian Matthes, Daniel Memmi, Timothy Menzies, Dieter Merkl, Zbigniew Michalewicz, Armin R. Mikler, Gautam Mitra, Roland Mittermeir, Madhav Moganti, Reinhard Moller, Tadeusz Morzy, Daniel Mossé, John Mueller, Jari Multisilta, Hari Narayanan, Jerzy Nawrocki, Rance Necaise, Elzbieta Niedzielska, Marian Niedq'zwiedziński, Jaroslav Nieplocha, Oscar Nierstrasz, Roumen Nikolov, Mark Nissen, Jerzy Nogieć, Stefano Nolfi, Franc Novak, Antoni Nowakowski, Adam Nowicki, Tadeusz Nowicki, Daniel Olejar, Hubert Österle, Wojciech Olejniczak, Jerzy Olszewski, Cherry Owen, Mieczyslaw Owoc, Tadeusz Pankowski, Jens Penberg, William C. Perkins, Warren Persons, Mitja Peruš, Fred Petry, Stephen Pike, Niki Pissinou, Aleksander Pivk, Ullin Place, Peter Planinšec, Gabika Polčicová, Gustav Pomberger, James Pomykalski, Tomas E. Potok, Dimithu Prasanna, Gary Preckshot, Dejan Rakovič, Cveta Razdevšek Pučko, Ke Qiu, Michael Quinn, Gerald Quirchmayer, Vojislav D. Radonjic, Luc de Raedt, Ewaryst Rafajlowicz, Sita Ramakrishnan, Kai Rannenberg, Wolf Rauch, Peter Rechenberg, Felix Redmill, James Edward Ries, David Robertson, Marko Robnik, Colette Rolland, Wilhelm Rossak, Ingrid Russel, A.S.M. Sajeev, Kimmo Salmenjoki, Pierangela Samarati, Bo Sanden, P. G. Sarang, Vivek Sarin, Iztok Savnik, Ichiro Satoh, Walter Schempp, Wolfgang Schreiner, Guenter Schmidt, Heinz Schmidt, Dennis Sewer, Zhongzhi Shi, Mária Smolárová, Carine Souveyet, William Spears, Hartmut Stadtler, Stanislaw Stanek, Olivero Stock, Janusz Stokłosa, Przemysław Stpiczyński, Andrej Stritar, Maciej Stroinski, Leon Strous, Ron Sun, Tomasz Szmuc, Zdzislaw Szyjewski, Jure Šilc, Metod Škarja, Jiři Šlechta, Chew Lim Tan, Zahir Tari, Jurij Tasič, Gheorge Tecuci, Piotr Teczynski, Stephanie Teufel, Ken Tindell, A Min Tjoa, Drago Torkar, Vladimir Tosic, Wieslaw Traczyk, Denis Trček, Roman Trobec, Marek Tudruj, Andrej Ule, Amjad Umar, Andrzej Urbanski, Marko Uršič, Tadeusz Usowicz, Romana Vajde Horvat, Elisabeth Valentine, Kanonkluk Vanapipat, Alexander P. Vazhenin, Jan Verschuren, Zygmunt Vetulani, Olivier de Vel, Didier Vojtisek, Valentino Vranić, Jozef Vyskoc, Eugene Wallingford, Matthew Warren, John Weckert, Michael Weiss, Tatjana Welzer, Lee White, Gerhard Widmer, Stefan Wrobel, Stanislaw Wrycza, Tatyana Yakhno, Janusz Zalewski, Damir Zazula, Yanchun Zhang, Ales Zivkovic, Zonling Zhou, Robert Zorc, Anton P. Železnikar

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: http://, Europe: http://ai.ijs.si/informatica, Asia: http://www.comp.nus.edu.sg/ liuh/Informatica/index.html.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2004 (Volume 28) is
– USD 80 for institutions,
– USD 40 for individuals, and
– USD 20 for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, Kranj, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Žibertova 1, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 1 4773 900, Fax (+386) 1 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)
ACM Slovenia (Dunja Mladenič)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik

The issuing of the Informatica journal is financially supported by the Ministry of Education, Science and Sport, Trg OF 13, 1000 Ljubljana, Slovenia.

Volume 28 Number 4 December 2004 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **28** (2004) Number 4, pp. 335–441