Volume 18 Number 2

ACTA CYBERNETICA

Szeged, 2007

ACTA CYBERNETICA

Information for authors. Acta Cybernetica publishes only original papers in the field of Computer Science. Manuscripts must be written in good English. Contributions are accepted for review with the understanding that the same work has not been published elsewhere. Papers previously published in conference proceedings, digests, preprints are eligible for consideration provided that the author informs the Editor at the time of submission and that the papers have undergone substantial revision. If authors have used their own previously published material as a basis for a new submission, they are required to cite the previous work(s) and very clearly indicate how the new submission offers substantively novel or different contributions beyond those of the previously published work(s). Each submission is peer-reviewed by at least two referees. The length of the review process depends on many factors such as the availability of an Editor and the time it takes to locate qualified reviewers. Usually, a review process takes 6 months to be completed. There are no page charges. Fifty reprints are supplied for each article published.

Manuscript Formatting Requirements. All submissions must include a title page with the following elements:

- title of the paper
- author name(s) and affiliation
- name, address and email of the corresponding author
- An abstract clearly stating the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

References should appear in a separate bibliography at the end of the paper, with items in alphabetical order referred to by numerals in square brackets. Please prepare your submission as one single PostScript or PDF file including all elements of the manuscript (title page, main text, illustrations, bibliography, etc.). Manuscripts must be submitted by email as a single attachment to either the most competent Editor, the Managing Editor, or the Editor-in-Chief. In addition, your email has to contain the information appearing on the title page as plain ASCII text. When your paper is accepted for publication, you will be asked to send the complete electronic version of your manuscript to the Managing Editor. For technical reasons we can only accept files in LaTeX format.

Subscription Information. Acta Cybernetica is published by the Institute of Informatics, University of Szeged, Hungary. Each volume consists of four issues, two issues are published in a calendar year. Subscription rates for one issue are as follows: 5000 Ft within Hungary, €40 outside Hungary. Special rates for distributors and bulk orders are available upon request from the publisher. Printed issues are delivered by surface mail in Europe, and by air mail to overseas countries. Claims for missing issues are accepted within six months from the publication date. Please address all requests to:

Acta Cybernetica, Institute of Informatics, University of Szeged
P.O. Box 652, H-6701 Szeged, Hungary
Tel: +36 62 546 396, Fax: +36 62 546 397, Email: acta@inf.u-szeged.hu

Web access. The above informations along with the contents of past issues are available at the Acta Cybernetica homepage http://www.inf.u-szeged.hu/actacybernetica/ .

Bálint Dömölki
IQSOFT
Budapest, Hungary
domolki@iqsoft.hu

Zoltán Ésik
Department of Foundations of
Computer Science
University of Szeged
Szeged, Hungary
ze@inf.u-szeged.hu

Zoltán Fülöp
Department of Foundations of
Computer Science
University of Szeged
Szeged, Hungary
fulop@inf.u-szeged.hu

Ferenc Gécseg
Department of Computer Algorithms
and Artificial Intelligence
University of Szeged
Szeged, Hungary
gecseg@inf.u-szeged.hu

Jozef Gruska
Institute of Informatics/Mathematics
Slovak Academy of Science
Bratislava, Slovakia
gruska@savba.sk

Helmut Jürgensen
Department of Computer Science
Middlesex College
The University of Western Ontario
London, Canada
helmut@csd.uwo.ca

Alice Kelemenová
Institute of Computer Science
Silesian University at Opava
Opava, Czech Republic
Alica.Kelemenova@fpf.slu.cz

László Lovász
Department of Computer Science
Eötvös Loránd University
Budapest, Hungary
lovasz@cs.elte.hu

Gheorghe Păun
Institute of Mathematics of the
Romanian Academy
Bucharest, Romania
George.Paun@imar.ro

András Prékopa
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
prekopa@cs.elte.hu

Arto Salomaa
Department of Mathematics
University of Turku
Turku, Finland
asalomaa@utu.fi

László Varga
Department of Software Technology
and Methodology
Eötvös Loránd University
Budapest, Hungary
varga@ludens.elte.hu

Heiko Vogler
Department of Computer Science
Dresden University of Technology
Dresden, Germany
vogler@inf.tu-dresden.de

Gerhard J. Woeginger
Department of Mathematics and
Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
gwoegi@win.tue.nl

Symposium of Young Scientists on Intelligent Systems

Guest Editors:

László Csink

Institute of Software Technology
Budapest Tech
Budapest, Hungary
csink.laszlo@nik.bmf.hu

Bálint Molnár

Department of Information Systems
Corvinus University of Budapest
Budapest, Hungary
molnar@informatika.bke.hu

Péter Szeredi

Department of Computer Science
and Information Theory
Budapest University of Technology
and Economics
Budapest, Hungary
szeredi@cs.bme.hu

Tibor Gregorics

Department of Software Technology
and Methodology
Eötvös Loránd University
Budapest, Hungary
gregorics@inf.elte.hu

Edit Sántáné-Tóth

Department of Software Technology
and Methodology
Eötvös Loránd University
Budapest, Hungary
santa@inf.elte.hu

Zoltán Vámossy

Institute of Software Technology
Budapest Tech
Budapest, Hungary
vamossy.zoltan@nik.bmf.hu

László Zsolt Varga

System Development Department
Computer and Automation Research
Institute of the Hungarian Academy
of Sciences (MTA SZTAKI)
Budapest, Hungary
laszlo.varga@sztaki.hu

Preface

This issue of *Acta Cybernetica* contains six papers originally presented at the First Symposium of Young Scientists, entitled *Intelligent Systems 2006.* The Symposium was organised by the Artificial Intelligence Section of the *John von Neumann Computer Society* (JvNCS), the Hungarian member of the *European Coordinating Committee for Artificial Intelligence* (ECCAI). The Symposium was held in Budapest, on November 23, 2006, and included 11 talks and 14 poster presentations by young scientists. The Symposium also featured an invited talk "The mystery of intelligence, what is the solution?" by András Lõrincz, who was awarded the prestigious title of ECCAI Fellow in 2006. More information on the Symposium can be obtained from the home page: `http://sas.ilab.sztaki.hu/njszt-mi/szimpozium2006.htm`.

The Symposium was part of international celebrations of the golden anniversary of the 1956 Dartmouth Conference that marks the beginning of Artificial Intelligence as a research field (`http://www.eccai.org/50years.shtml`).

After the Symposium the authors of both standard and poster presentations were invited to submit papers to a Special Issue of *Acta Cybernetica.* Eleven papers were received and were then subjected to the normal refereeing process of the Journal. The six accepted papers cover a broad spectrum of topics, and report on progress both in the theory and in the practice of AI. The first three papers in this volume discuss advances in game theory (Melkó et al.), new techniques in image processing (Jankó et al.) and in signal processing (Szabó et al.). The other three articles describe AI applications: production planning utilising multiagent technology (Egri et al.), scheduling using Operation Research techniques (Hanák et al.), and extending Web browsers to support semantic technologies (Jeszenszky).

Thanks are due to the authors for submitting their contributions and to the reviewers for their help in the preparation of this issue.

The Second Symposium of Young Scientists, *Intelligent Systems 2007* is being held on November 23, 2007, and a similar Special Issue of Acta Cybernetica is scheduled for 2008.

László Csink, Edit Sántáné-Tóth, László Zsolt Varga
Tibor Gregorics, Péter Szeredi
Bálint Molnár, Zoltán Vámossy

Guest Editors
Members of the Executive Board of the
Artificial Intelligence Section of JvNCS

Acta Cybernetica 18 (2007) 171–192.

Optimal strategy in games with chance nodes

Ervin Melkó* and Benedek Nagy†

Abstract

In this paper, games with chance nodes are analysed. The evaluation of these game trees uses the expectiminimax algorithm. We present pruning techniques involving random effects. The gamma-pruning aims at increasing the efficiency of expectiminimax (analogously to alpha-beta pruning and the classical minimax). Some interesting properties of these games are shown: for instance, a game without draw can be fair. A fair game may not be fair any more if it is played iteratively. To handle these phenomena, the use of additional indicators, such as the minimal guaranteed outcome value, is suggested.

Keywords: Game Theory, Artificial Intelligence, Game tree, Games with chance nodes, Expectiminimax algorithm, Pruning, Iterative games, Fair games

1 Introduction

Game Theory is an important field of Artificial Intelligence. Modern game theory was defined by von Neumann and Morgenstern [6, 7]. Game theory deals with decision problems in an environment where another agent or agents may have different aims. The theory of two player strategic games is well developed [5, 9, 10]. A subfield, the theory of fixed sum games, is equivalent to the family of zero-sum games, where two players have opposite aims. In games with perfect information players have all the information about the game to help to make their decisions (also called steps or moves). These games are represented by game trees. In these graphs there are two kinds of nodes representing the decision points of the two players. Game theory deals with well-defined games where players can choose among a fixed set of actions. NIM, Tic-Tac-Toe, Othello [1] and Chess are representative elements of this set. The first computer chess program was developed by Shannon and Turing in 1950. One of the main aims of the artificial intelligence research was to write a chess program that can beat the human world chess champion. It was a long process, but nowadays computer programs do beat human champions.

*University of Debrecen, Debrecen, Hungary, E-mail: melko.ervin@gmail.com

†University of Debrecen, Debrecen, Hungary and Rovira i Virgili University, Tarragona, Spain, E-mail: nbenedek@inf.unideb.hu

Game and decision theories are very important fields of Economics, as well. The Decision Theory deals, for instance, with problems represented by decision trees, in which a person has some decision points and there are some other (uncertain) events represented by (chance) points. Usually, these other points represent cases which occur randomly. The expected values can be computed to help in choosing a branch at decision points.

In this paper, we deal with fixed sum two player strategic games that include an element of chance. A well-known example for this type of game is the Backgammon. The order of players and possible random events with their probabilities are known by the players, therefore they can compute their optimal strategies. In the next section we recall some well-known elements of the theory of decision trees (one player "game" with random effects) and of the theory of two player non-random games. In Section 3.1, connecting these two theories, a method is shown to compute an optimal strategy in games with chance nodes. In Section 3.2 this algorithm, called *expectiminimax* is extended by introducing the pruning of the game trees. Because the game tree contains chance nodes, we introduce new pruning techniques, resulting in the algorithms of gamma-pruning. These methods give fast exact evaluation of the game, therefore one can find the optimal strategy faster than by analysing the full game-tree.

The games including chance nodes are more complicated than the games without chance. For example, a fair game without random effects remains fair if it is played repeatedly. In Section 4, we present an interesting example, where a fair game may cease to be fair if it is played repeatedly. We show variants of the previous algorithms that can help in the evaluation of such games by computing the minimal guaranteed outcome. Finally, in Section 5 we summarise the conclusions of the paper.

We note here that the topic of the paper has a strong relation with multiplayer games (or games with few players [4]).

2 Preliminaries

In this section we recall some basic definitions and basic algorithms, facts about the topic of decision trees and two player game trees based mostly on [10].

2.1 The theory of decision trees

Let A be a player who has some decision points. There are also some random events (their probabilities are known).

Formally, let the problem be the following. There is a tree with two kinds of nodes. At the so-called *decision nodes*, the player chooses a successor node (branch of the tree). At the nodes of the other type, the *chance nodes*, a random event happens: the successor node is chosen randomly with probabilities known in advance. The leaves of the tree represent situations where the utility function (the payoff, the value of the situation) is known. The question is the strategy, i.e.

what should the player's choice be at the decision points to achieve the maximal (expected) payoff.

The aim of the player is to maximise the outcome value and for this purpose she/he chooses the child-node (i.e. the branch of the tree) with the highest possible expected value.

Let $P(N)$ denote the probability of the event N (supposing that we are at the parent of the node representing N). In Figure 1, an example is shown. There are decision nodes, where the player chooses among the next possible nodes (represented by rectangles in the figure). At nodes marked by a circle a random event will determine the next node. (The probabilities are written on the edges.)

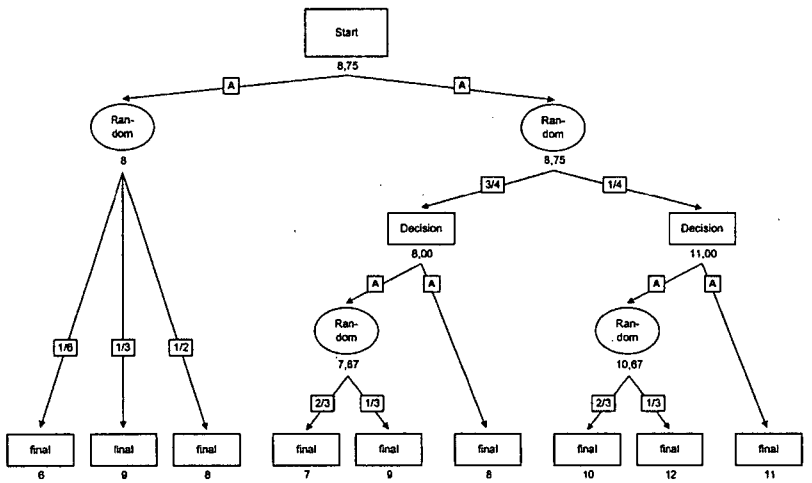


Figure 1: A decision tree

Algorithm 1. (DECISION)

```
1  function D(N) is
2  begin
3    if N is a leaf then
4      return the value of this leaf
5    else
6      Let N_1, N_2, ..., N_m be the successors of N
7      if N is an A node then
8        return max{D(N_1), ...,D(N_m)}
9      if N is a C node then
10       return P(N_1)D(N_1) + ... + P(N_m)D(N_m)
11 end D
```

In Figure 1 the nodes of the tree are assigned a numeric value (shown under the rectangles and circles representing the nodes of the tree), as described by Algorithm 1.

The technique presented here is called *expectimax*, since at decision points of the player (A nodes) it computes the maximum of the expected values of the successor nodes, while at the random effects (chance nodes or C nodes, for short) the expected value of the values of the successor nodes is computed.

Theorem 1. *If the decision tree is finite, Algorithm 1 gives the possible maximal expected value that the player can achieve.*

Note that every decision tree can be transformed to a *layered* (stratified) decision tree, i.e. a tree in which the nodes at the same depth are of the same type. If the player has to make multiple decisions in a sequence, then this sequence can be replaced by a complex decision. Similarly if random events follow each other, then this can be viewed as a single complex random effect (using probability theory).

In this paper we deal only with problems having a bounded set of possible outcomes. This is important to avoid some paradoxes of the probability theory. For instance the following game is known as the Saint-Petersburg paradox [8]. Drop a coin, if it is a head then drop again. If it is a tail, then you get \$ 2^{k+1} and the game ends, where k is the number of heads before. What is the value of this game? How much should you pay for a game if we want to do it in a fair way? On one hand, the expected value is $\frac{1}{2}\cdot 2^1+\left(1-\frac{1}{2}\right)\cdot\frac{1}{2}\cdot 2^2+\left(1-\frac{1}{2}\right)^2\cdot\frac{1}{2}\cdot 2^3+\ldots+\left(1-\frac{1}{2}\right)^k\cdot\frac{1}{2}\cdot 2^{k+1}+\ldots = \sum_{k=1}^{\infty}\left(\frac{1}{2^k}\cdot 2^k\right) = \sum_{k=1}^{\infty} 1 = \infty$. So, in principle, it is worth to buy the right to play for an arbitrarily large price. On the other hand, everybody has the feeling that it would not be worth to play for a very high price. In practice there are no people who would pay, let us say, \$ 1 000 000 000 for a game. The chance to win more than this amount is less than 1 : 500 000 000, since at least 29 consecutive heads are needed to begin the sequence. Thus the chance to loose a very high amount of money is very close to 1.

2.2 The theory of two player strategic games

A game is defined as follows. Let two players be given. They take turns in making moves, until the game is over. At every point of time the game has a state (e.g. the state of the board) and the player to make the next turn is known. In every state the possible moves form a well-defined (finite) set. Player A starts the game from its initial position. There are terminal states, in which the utility value of the state is given, for each player. In case of fixed sum games, when the sum of utility values is known, one utility value is enough to determine the result of the game at a given state.

In simple games, there are only two outcomes: A wins or B wins. In some games the result can be a draw. In more complex games a score is computed, and the final score is of importance. In these games the set of utility values can be very large. However, recall that we deal only with games which have a finite set of

utility values.

The following graphical representation of a game is called a game tree. The nodes of the tree represent the game states, while the arcs represent moves. There are two kinds of nodes representing the decision situations of player A and B, respectively. At *A nodes* player A chooses a successor node, while at *B nodes* the decision is B's. At the root node (the initial state) player A has the decision. The leaves represent terminal positions with separate utility values for the two players.

We only deal with fixed-sum games. Therefore it is enough to specify the utility value for player A, as the value for B can be easily computed. A terminal position where A wins is thus assigned a value which is always greater than the value of a position where B wins. We also allow draws, and arbitrary intermediate values.

We are thus considering two player, fixed-sum, perfect information finite games. We assume that the two players (A and B) take turns and try respectively to maximise and minimise the utility function, the value of the game.[1] Since players take turns, successive nodes represent positions where different players must move. We consider only games with a finite game tree.

In Figure 2 a part of the game tree of the well-known game Tic-Tac-Toe is shown [10]. In the upper part of the figure the states are shown, while in the lower part, in a more abstract way, the triangles show the player who can choose from the next states (if any). Triangles Δ represent decision points for player A, while triangles ∇ represent decision points for player B. The numbers in the figure represent so called minimax values, which are discussed below.

2.2.1 The minimax algorithm

Assuming a perfect opponent for deterministic games, a feasible strategy is as follows: choose a move to a position with the highest minimax value. In this way the best achievable payoff against best play can be reached.

The minimax game strategy for player A (B) is to select the move that leads to the successor node with the highest (lowest) value. The values are computed starting from the leaves of the tree and are propagated upwards to their predecessors in accordance with the minimax algorithm. Algorithm 2 below [2] is a concise formulation of the minimax principle which works properly, even if the nodes types do not alternate (similarly to Algorithm 1).

[1] If a game allows a sequence of steps performed by the same player, then this can be viewed as a single complex step. If, in the original game, only a fixed number of such step sequences is allowed, then the resulting game will be finite, as well. Such a transformation can help to use the theory described in the sequel.

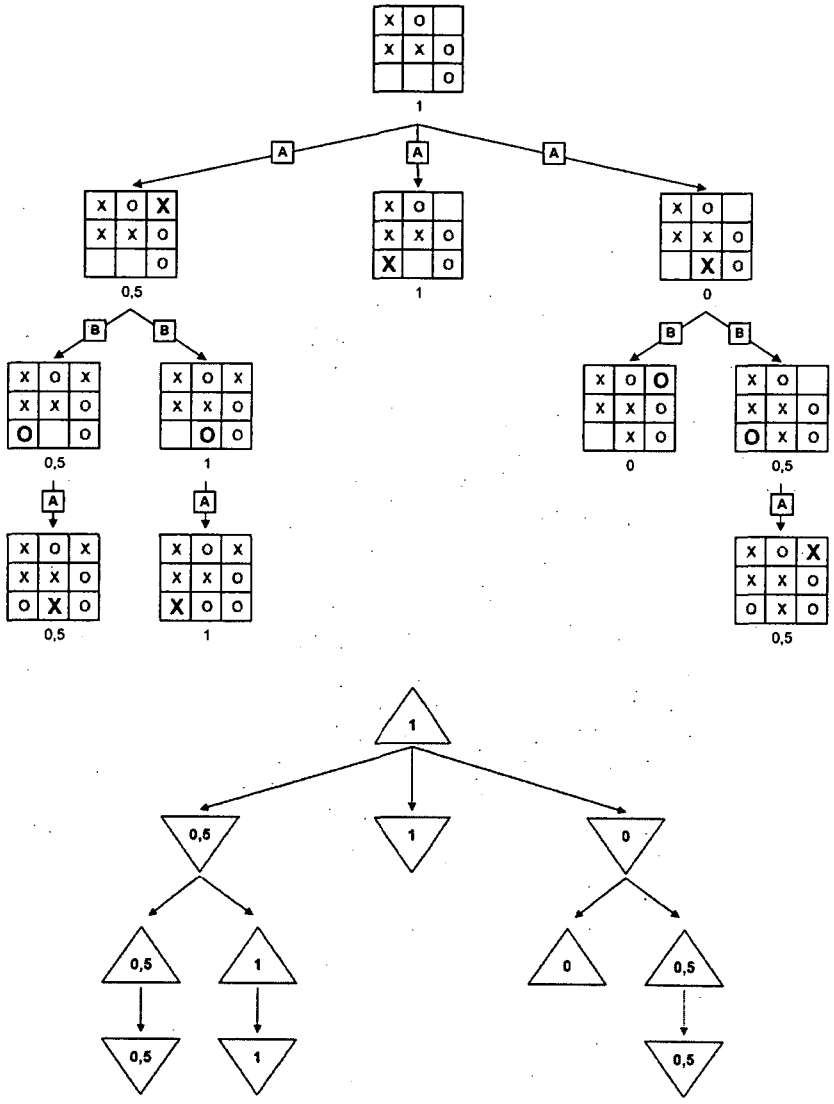


Figure 2: A part of the game tree of Tic-Tac-Toe

Algorithm 2. (MINIMAX)

```
1  function MM(N) is
2  begin
3    if N is a leaf then
4      return the value of this leaf
5    else
6      Let N_1, N_2, ..., N_m be the successors of N
7      if N is an A node then
8        return max{MM(N_1), ...,MM(N_m)}
9      if N is a B node then
10       return min{MM(N_1), ...,MM(N_m)}
11 end MM
```

The *value* of a (non-random) game is the payoff that both players can guarantee. So, if both players are playing well, the game will be finished in a state having the value of the game.

A (non-random) game is said to be *fair* if both players can guarantee at least the draw.

The following theorem is well-known about the correctness of the algorithm.

Theorem 2. *The minimax algorithm gives the exact value of any two player strategic game represented by a finite game tree.*

Theorem 2 is von Neumann's minimax theorem [6] for game trees (as formulated in [2]).

As a consequence of the previous theorem we can say that a two player strategic game cannot be fair, if draws are not allowed. In this case one of the players must have a winning strategy.

In Figure 2 the numbers beneath the nodes are the values assigned to the nodes by the minimax algorithm.

The problem with this algorithm is that it explores all nodes of the tree. In the next subsection we recall some methods which prune parts of the game tree and thus accelerate the evaluation.

2.2.2 Alpha and beta pruning

Alpha-beta cutoff is a method for reducing the number of nodes explored by the minimax algorithm described above. For each node it explores, it computes an alpha value and a beta value.

One can make the decision about the best choice without knowing the values of all successor nodes. The alpha and beta values, which are estimations of possible maximum and minimum values, can help in this.

As a first step we divide Algorithm 2 to define two separate functions for computing the value of A nodes and B nodes, respectively. This form of the algorithm can be found, for instance, in [10].

Algorithm 3. (MINI and MAX)

```
1  function maxvalue(N)
2   begin
3   if N is a leaf then
4      return the value of this leaf
5   else
6      let v = −∞
7      For every successor N_i of N do
8        let v = max{v,minvalue(N_i)}
9   return v
10  end maxvalue

1  function minvalue(N)
2   begin
3   if N is a leaf then
4      return the value of this leaf
5   else
6      let v = +∞
7      For every successor N_i of N do
8        let v = min{v,maxvalue(N_i)}
9   return v
10  end minvalue
```

Step 8 of the functions does the main operation, it calculates the maximal and the minimal value, respectively. (This new form of the minimax algorithm works only if the levels of A nodes and B nodes alternate in the tree.)

We now discuss how short-cuts can be used to eliminate some parts of this evaluation process. The alpha and beta pruning algorithm can abandon the evaluation of a step when at least one possibility has been found that proves the step to be worse than a previously examined step. Such moves need not be evaluated further. For this, we explain the role of alpha and beta values that are approximations of the real value of the node.

The *alpha value* of a node is a value, which is never greater than the real value of this node. Initially it is the value of that node if the node is a leaf, otherwise it is minus infinity. Then, at an A node, it is set to the largest of the values of its successors explored up to now, and at a B node, to the alpha value of its predecessor.

The *beta value* of a node is a value, which is never smaller than the real value of this node. Initially it is the value of that node if the node is a leaf, otherwise it is (plus) infinity. Then, at a B node, it is set to the smallest of the values of its successors explored up to now, and at an A node, to the beta value of its predecessor. It is guaranteed that:

The real value of a node will never be less than the alpha value and never greater than the beta value of that node. As the algorithm evolves, the alpha and beta

values of a node may change, but the alpha value will never decrease and the beta value will never increase. When a node is visited for the last time, its value is set to the alpha value of that node if it is an A node, otherwise it is set to the beta value. For more details we refer to [3, 10]. We formalise this method as Algorithm 4 below.

Algorithm 4. (ALPHA and BETA PRUNING)

```
1  function alphamax(N, α, β)
2   begin
3   if N is a leaf then
4      return the value of this leaf
5   else
6      let v = −∞
7      For every successor N_i of N do
8        let v = max{v,betamin(N_i, α, β)}
9        if v ≥ β then return v
10       let α = max{α, v}
11 return v
12 end alphamax

1  function betamin(N, α, β)
2   begin
3   if N is a leaf then
4      return the value of this leaf
5   else
6      let v = +∞
7      For every successor N_i of N do
8        let v = min{v,alphamax(N_i, α, β)}
9        if v ≤ α then return v
10       let β = min{β, v}
11 return v
12 end betamin
```

Observe that $\alpha \leq \beta$ for every node at every time.

The evaluation technique using Algorithm 4 is the so-called alpha-beta pruning (or alpha-beta minimax) algorithm.

The two functions call each other because the type of the nodes in the game tree alternate, so we can speak about A-layers and B-layers of the game tree.

If player A starts the game, then the alphamax function is called with the root of the game tree, as its first parameter, minus infinity ($-\infty$), as the value of α, and plus infinity ($+\infty$) as the value of β. We note here that usually (always in case of games with bounded set of outcomes) the value $-\infty$ (∞) is replaced by any number smaller (larger) than the lower (upper) bound of the outcomes.

The alpha-beta pruning algorithm is correct as the next theorem (based on [1]) states.

Theorem 3. *For finite non-random games Algorithm 4 results the same value as Algorithm 2.*

Thus, pruning does not affect the final result, so the same (exact) value can be computed in a much shorter time.

Note that the speed of the alpha-beta algorithm depends very much on the order in which the branches of the game tree are considered [3].

3 Two player games with random effects

We now introduce some extensions of the previous theories. Games with random events in the successor function (such as dropping a coin, rolling a dice) can be analysed using the methods described in this section. The so-called expectiminimax algorithm is presented, and some acceleration techniques are shown.

The problems we deal with are represented by trees. These trees are related to game trees and decision trees, as well. There are three types of nodes. At A nodes and B nodes player A and B makes the decision, respectively. The third type of node represents the random events. These new nodes, called the chance nodes (or type C nodes) play the same role as in decision trees.

Let us see an example.

Let the game be a modified version of Tic-Tac-Toe. Player A starts the game (by placing an 'X' in an empty cell), player B follows (by placing a 'O'), and then a random event happens, as Figure 3 shows. Places 1 and 4 will be interchanged (event w_1) with a probability $P(w_1) = \frac{1}{2}$, or places 2 and 6 will be interchanged (w_2) with a probability $P(w_2) = \frac{1}{3}$, or places 5 and 9 will be interchanged ($P(w_3) = \frac{1}{6}$). The game is continued for 4 iterations of steps A-B-random. Finally, player A wins if she/he has 3 'X' signs in a row (or in a column or diagonally), but B has no 3 'O'-s in such a way. B wins if there are 3 'O'-s in a row (in a column or diagonally), but there are no 3 'X'-s in this way. The result is a draw, if both or none of the players have 3 of their signs in a row (column or diagonal). Let us fix the value of the game as 1 point. The winner gets the whole point. At draw both players get a half point.

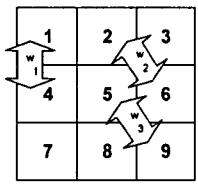


Figure 3: TIC-TAC-TOE with some random events

Because of the influence of random effects, the aim is to optimise the expected value of the game. (In games with no random effects, the game has an exact value;

in games of chance the expected value plays the same role.) The concept of *fair game* can be extended to these games: a game with chance nodes is fair if both players can guarantee the draw as the expected value.

Proposition 4. *If the game tree has no layers, then it can be modified to an equivalent one which has layers in a fixed order.*

This can be done by uniting consecutive nodes of the same type, and by introducing some new nodes where the branching factor of the tree is 1.

3.1 Modified minimax

In this section the basic evaluation algorithm is presented. While player A chooses the possible maximum, and player B chooses the possible minimum value, because of their aims, at chance nodes (of type C) there is a random event where the expected value (weighted average) is calculated. $P(N)$ denotes the probability of the fact that node N will be chosen as the successor of its type C parent node at a game. Now we formulate the general form of the so-called expectiminimax algorithm, adapted from [10].

Algorithm 5. (EXPECTIMINIMAX)

```
1  function EMM(N) is
2  begin
3    if N is a leaf then
4      return the value of this leaf
5    else
6      Let N_1, N_2, ..., N_m be the successors of N
7      if N is an A node then
8        return max{EMM(N_1), ...,EMM(N_m)}
9      if N is a B node then
10       return min{EMM(N_1), ...,EMM(N_m)}
11     if N is a C node then
12       return P(N_1)EMM(N_1) + ... + P(N_m)EMM(N_m)}
13 end EMM
```

We say that player A plays well, if she/he is playing with a strategy to maximise the expected payoff, while player B plays well, if she/he is minimising the expected payoff of the game.

Theorem 5. *If both players play well, then Algorithm 5 gives the expected value of the game.*

If any of the players follows another strategy, then her/his expected outcome will be less favourable.

Let us analyse the game shown in the previous section. For simplicity, let our start state be a Tic-Tac-Toe board shown at the root of Figure 4. The evaluation of this part of the game tree by Algorithm 5 can be seen in the figure.

Note here that in non-random games players can choose a strategy so that the value of the game is always guaranteed. In games with chance nodes this is not true, only the expected value is optimised. In Section 4 we will show an interesting consequence of this fact.

If the nodes of the same type form layers in the tree, and these layers have a fixed order, then this algorithm can be written as three functions (in the same way as Algorithm 3 uses two functions), each of which computes the value of a different node type.

The fact that the optimal strategy depends not only on the order of the possible outcomes of the game, but on their values, leads to a phenomenon which is similar to the Saint-Petersburg paradox. The presence of a very improbable branch with a very high value may drastically change the situation. (We discuss this issue at the beginning of the next subsection.)

While [10] states that in the presence of random effects it is impossible and useless to develop strategies for the players, we believe that in some cases it is worth to compute the possibilities, and the game-evaluation can be accelerated by pruning techniques (when the set of the possible game values is bounded), as well.

In the following example we show that a game without a draw can be fair and correct if random events can happen.

An example for this phenomenon is the following game. The base (non-random) game is the well-known NIM [2]. (There are 3 groups of beans, two players take turns in removing some beans from an arbitrary group of beans. The player who removes the last bean wins the game.) The winner gets a point, the looser gets nothing. Let there be a random event with two outcomes, after every step of each player. The first outcome, w_1, where $P(w_1) = \frac{1}{2}$, implies that a bean is moved from the first group to the second group (provided there is a bean in the first group, otherwise nothing happens). At the second outcome, w_2, where $P(w_2) = \frac{1}{2}$, a bean is moved from the second group to the third group, if this is possible. If the starting position is $(0, 2, 1)$ then it can be easily seen that player A has an optimal strategy with the expected value of $\frac{1}{2}$. So, the game is fair. It cannot be a draw, but both players have the same chance to win.

It can also be shown that the value of this game is $v = \max\{P(w_1), P(w_2)\}$. Figure 5 presents the possible outcomes and the evaluation of the game for the probabilities $P(w_1) = \frac{1}{4}, P(w_2) = \frac{3}{4}$.

Figure 5 is a good example of a layered game tree, where on every branch the three types of nodes (A, C, and B) follow each other in a fixed order. When modifying Algorithm 5 to involve three separate functions one should care about the layers of the tree, and the functions should call each other according to this order. In each case the appropriate evaluating function, corresponding to the type of the successor nodes, must be applied.

3.2 Extensions of pruning techniques

Pruning techniques can be used to speed up the evaluation of games of chance, as well. In this section several variants of pruning are shown, most of them involve

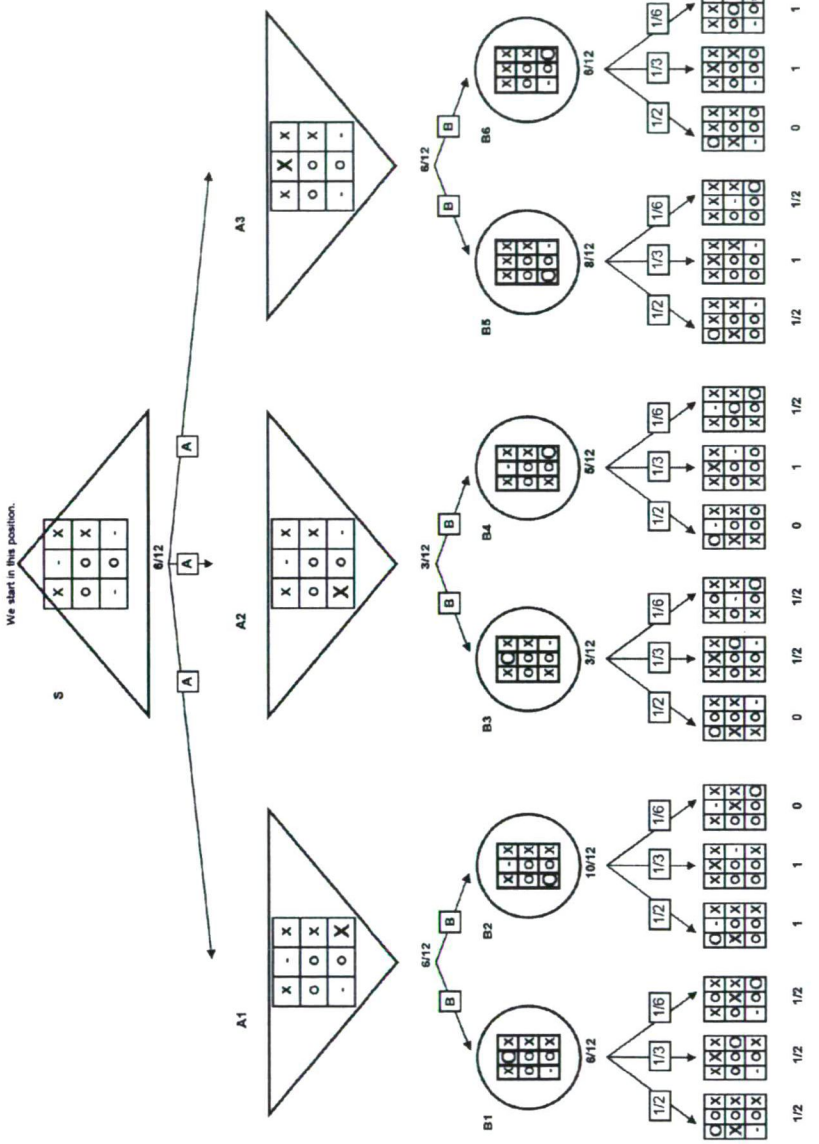


Figure 4: Evaluation of a part of the game tree of Tic-Tac-Toe with chance nodes

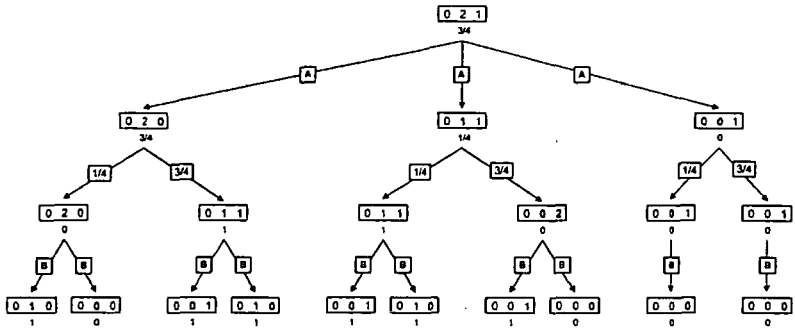


Figure 5: NIM with random events

chance nodes.

One could think that the branches of chance nodes, which have a very small probability can be ignored. However, doing this is a very rough estimation, because ignoring a branch with a small probability that has a very high or low value may drastically change the expected value. For instance, if there is a game with small leaf values (e.g. 0, 1, 2) in general, but also involving a leaf with a chance 10^{-8} that has a value 10^{12}, then this branch of the tree will be very significant. Observe here that changing the value of this leaf to 10^5, which is also much higher than the other values, causes this branch to lose its importance.

We think that the expected value itself may not be the correct quantity to evaluate the game, but it is still probably the most important value. Discussing this issue in detail is beyond the scope this paper. We only mention that situations similar to the Saint-Petersburg paradox can arise, in that very improbable effects may have a very big influence on the decision, if the latter is based (only) on the expected value. Note that in Section 4 we present some important values that can be used to improve the decision process.

The alpha (beta) pruning can be applied in a game of chance, if an A-layer is followed by a B-layer (a B layer followed by an A layer), exactly in the same way as for non-random games. This is because the layers below the immediate successors do not have any effect on the evaluation. So, in these cases the appropriate function of Algorithm 4 can be applied, but since the game tree has three kinds of nodes, rather than calling the other function (as in Algorithm 4), some parts of the expectiminimax algorithm should be used. (This means that at least line 8 of one of the functions must be changed, as required by the order of layers in the tree.)

We now present a new pruning technique, the so-called gamma pruning. It is based on the fact that knowing an estimated maximal or minimal possible outcome, one can easily decide whether a given branch can contribute to the solution, and so needs further exploration, or not.

The gamma pruning technique has two variants. The *gamma-B* (*gamma-A*) pruning can be applied if a B-layer (A-layer) of the tree is followed by a C-layer.

To introduce these techniques let us analyse two iterations of the expectimin-

imax algorithm together. So let us start from Algorithm 4 by replacing line 8 of both functions by the following:

8 let $v = \max\{v, \text{average}(N_i, \alpha, \beta)\}$

in alphamax and

8 let $v = \min\{v, \text{average}(N_i, \alpha, \beta)\}$

in betamin (where the function average computes the expected value at the node N_i as the weighted average of the values of its successors).

Now we discuss how one computes the average with pruning. If it is not necessary we do not compute the exact average, a lower or an upper estimation may still be of help in the decision making.

Let v_{max} and v_{min} be the maximal and minimal payoff of the game, respectively. These values will be used in the estimations.

Algorithm 6. (GAMMA PRUNING)

```
1  function gamma-A(N)
2   begin
3   let v = −∞
4   For every successor N_i of N do **** they are C nodes
5     let s = 0 and p = 0
6     For every successor N_{i,j} of N_i do
7       let the value of this node be v_{i,j}
8       let s = s + v_{i,j}P(N_{i,j}) and p = p + P(N_{i,j})
9       let α = s + (1 − p)v_max
10      if α ≤ v then exit from the loop
11    let v = max{α, v}
12  return v
13  end gamma-A

1  function gamma-B(N)
2   begin
3   let v = +∞
4   For every successor N_i of N do **** they are C nodes
5     let s = 0 and p = 0
6     For every successor N_{i,j} of N_i do
7       let the value of this node be v_{i,j}
8       let s = s + v_{i,j}P(N_{i,j}) and p = p + P(N_{i,j})
9       let β = s + (1 − p)v_min
10      if β ≥ v then exit from the loop
11    let v = min{β, v}
12  return v
13  end gamma-B
```

The idea of the algorithm is the following. The unknown (not calculated yet) values of the successors of a chance node are estimated as the best possible values for the player. If the expected value of this choice is less favourable (using this ideal estimation) than the known best option, then the evaluation of this option should not be continued.

In the special case of $v_{min} = 0$, which holds in our examples, the gamma-B function is evaluated in a faster way, because in this case s has the same value as β.

When computing the values $v_{i,j}$, pruning techniques can also be applied, depending on the order of layers of the game tree. If an A-layer and a B-layer follow each other, then the original alpha and beta pruning can be applied (depending on the order of layers). If an A-layer or a B-layer is followed by a C-layer, then the new technique, the gamma-pruning can be applied. For a given game, following the order of the layers, an appropriate recursive procedure can perform the evaluation. In this case, in line 7 of the gamma-pruning algorithm, the evaluation of the next layer can also be done recursively. For instance, if the order of the layers is A, B, C, then an alpha pruning calls a gamma-B pruning, etc.

Theorem 6. *The expected value of a game with chance nodes can be computed by using the functions of gamma-pruning algorithm. The expected value will be the same as the result of the expectiminimax algorithm.*

As an application of the gamma-B pruning, let us consider Figure 6, where a part of the game tree of the Tic-Tac-Toe game described previously (c.f. Figure 4) is shown. The figure also presents the values of the nodes. Observe that the part framed by a solid line can be pruned by gamma-B. The part framed by a dashed line can be eliminated by alpha pruning.

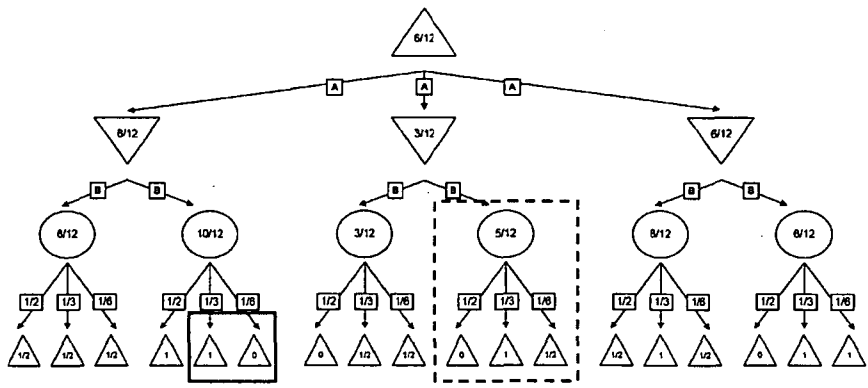


Figure 6: Accelerated evaluation by pruning: gamma-B pruning (solid box) and alpha pruning (dashed box)

Note that in the random NIM game both kinds of gamma pruning methods are applicable.

Of course, the speed-up due to these new pruning techniques also depends on the order of the branches. In case of gamma pruning, we believe that best results can be achieved when the successor nodes are tried in the order of decreasing probabilities (as it is shown in Figure 6). Usually the branches with higher probability have stronger effects on the computed values than the branches with lower probabilities.

4 A sequence of games as a two player game

In this section a new phenomenon is discussed. Let us consider a game as an iteration of a simple game as follows.

Let a simple base game be given (one can choose any of the games that can be described using the methods discussed in the previous sections). Now the players play several rounds of the base game, one after the other. The number of the base games can be fixed (say, for instance, n), or may depend on some events. The results of the games are accumulated. The winner of the iteration game will be the player who has collected more points than the other player at the end of the last game.

Let us consider a variant of the Tic-Tac-Toe game with random effects as shown in Figure 7. Here, the game starts from the state presented at the root of the tree, instead of the empty board, and so it involves only a single decision for each player. The expected value of this game is $\frac{1}{2}$, so the game seems to be fair.

Let the iteration game be a sequence of 4 base games of Figure 7. We show that player A has a strategy to win this game with a much higher probability than player B.

Let us first discuss the base game. Player A starts the game. She/he has three possibilities, but we can ignore branch A2, as this is bad for player A. Choice A1 (A3) involves placing an 'X' in the bottom-left corner (top-right corner) of the board, respectively. If player A chooses branch A3, she/he has a chance of $\frac{1}{2}$ to win, and the same chance to loose. If she/he chooses this strategy in every base game the outcomes of the iteration game can be the following (W stands for 'A wins', L for 'A looses', and D for a draw):

A wins:	WWWW, LWWW, WLWW, WWLW, WWWL	$P_A = \frac{5}{16}$
B wins:	LLLL, WLLL, LWLL, LLWL, LLLW	$P_B = \frac{5}{16}$
draw:	WWLL, WLWL, WLLW, LLWW, LWLW, LWWL	$P_{draw} = \frac{6}{16}$

If A chooses step A1 in each game, the draw can be guaranteed for both players.

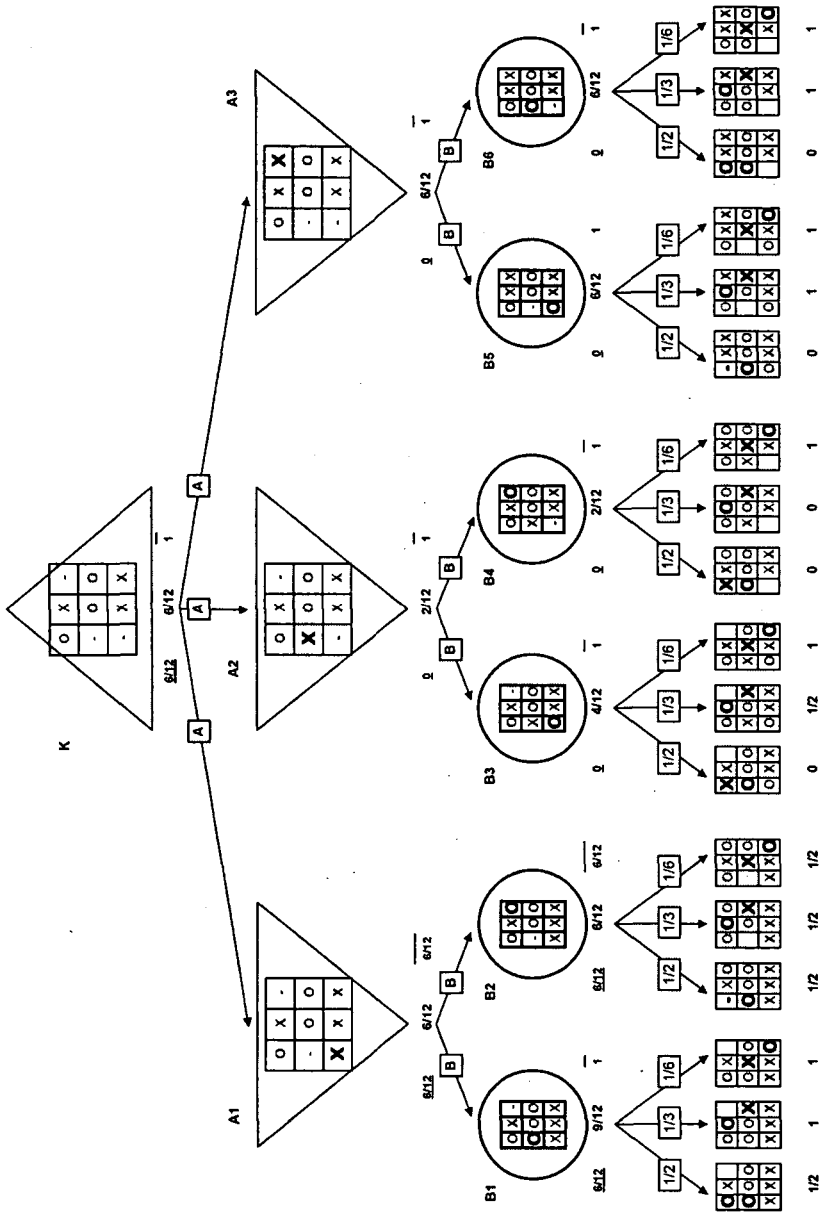


Figure 7: Evaluation with minimal guaranteed value in Tic-Tac-Toe with chance nodes

Let the strategy of player A in the iteration game be the following. In the first game she/he chooses the third possibility. In the subsequents games A plays as follows. If she/he has more points than player B has, then she/he chooses the step A1 to guarantee draw in every future game in the sequence. In this way A keeps her/his advantage. If A has not got more points than B, she/he chooses step A3 again in the next game.

It is easy to check that the possible outcomes of the sequence are the following:

A wins: WDDD, LWWD,	$P_A = \frac{10}{16}$
B wins: LLLL, LWLL, LLWL, LLLW	$P_B = \frac{4}{16}$
draw: LLWW, LWLW,	$P_{draw} = \frac{2}{16}$

This means that player A has a "probabilistic" winning strategy on this fair-like game. The techniques described in the previous sections do not show this phenomenon. This suggests that in the presence of iterative games we need something more than the expected value.

In our example we may introduce the concept of minimal guaranteed outcome. This is the value that the chosen strategy guarantees for A. It is impossible to have a smaller value for her/him if she/he plays well. (Thus it is the value that can be obtained in the case, when player A plays well, but the random events play against her/him.) At each node, in addition to the expected value, the minimal guaranteed value can also be computed for player A, and used in the decision process.

The computation of the minimal guaranteed outcome value can be done by a simple modification of the EXPECTIMINIMAX algorithm, by choosing the minimum at C nodes:

Algorithm 7. (MINIGMINIMAX)

```
1  function MGMM(N) is
2  begin
3    if N is a leaf then
4      return the value of this leaf
5    else
6      Let N_1, N_2, ..., N_m be the successors of N
7      if N is a A node then
8        return max{MGMM(N_1), ...,MGMM(N_m)}
9      if N is a B node then
10       return min{MGMM(N_1), ...,MGMM(N_m)}
11     if N is a C node then
12       return min{MGMM(N_1), ...,MGMM(N_m)}
13 end MGMM
```

It is easy to see that the algorithm is correct:

Theorem 7. *Algorithm 7 results in the (guaranteed) minimal possible payoff of player A if she/he plays well.*

This evaluation usually offers a different value than the expectiminimax algorithm, and can imply a different decision for player A.

Note that the evaluation for player B can be done in a similar way, computing the worst possibility for B at chance nodes. This requires replacing the function minimum by maximum in line 12. So the algorithm maxigminimax for player B is the same as Algorithm 7, except for line 12 being the following:

12 `return` $\max\{\texttt{MGMM}(N_1),...,\texttt{MGMM}(N_m)\}$

From A's point of view we can call this value the maximal possible value by choice (player B plays well, and the random events help A). Usually the evaluation for player B gives a possible game which differs from the game that Algorithm 7 suggests by evaluating the game for A.

In Figure 7 we have shown the minimal guaranteed values (underlined) and the maximal possible values by choice (overlined) on the left and right hand side of the expected values of the nodes, respectively.

This new type of information gives the possibility for player A to choose between the guaranteed draw (first option, A1 in the figure) and the random outcome (third option, A3). At step A1, the minimal guaranteed value is the same as the expected value at that node. This shows that the draw can be guaranteed. Therefore, using this additional information, A has a probabilistic winning strategy for the sequence of games, as detailed above.

Note that there is no problem regarding the fair outcome of the games in the sequence. If A employs the above winning strategy, and wins the sequence (which has a probability $P_{A,1} = \frac{5}{8}$), she/he wins exactly by 1 point (which is the minimal possible difference). There is a chance for draw as well ($P_{draw} = \frac{1}{8}$). At same time when B wins, she/he wins by 2 points ($P_{B,2} = \frac{3}{16}$) or by 4 points ($P_{B,4} = \frac{1}{16}$). In this way the expected value of the points achieved in the sequence of games shows a fair equilibrium between the two players. However, normally, the person who won matters and not the difference of scores.

Note that the algorithms for calculating the minimal guaranteed value and the minimal possible value by choice use only the maximum and minimum functions. Therefore alpha and beta pruning can be applied in a straightforward way in these algorithms. C nodes can be viewed as B nodes (in the evaluation for A) or as A nodes (in the evaluation for B). Using Proposition 4 these trees can be transformed to normal game trees, and on these new trees the evaluation can follow Algorithm 4.

5 Conclusions

Combining the theory of decision trees and two player strategic games, we have presented and analysed the theory of games with chance nodes. In these games, the expected value plays the role of the utility value of the non-random games.

In the presence of random events, a game without a draw outcome can still be fair, i.e. guarantee that the chance of winning for both players is the same (assuming

they play well).

The tree of games with chance nodes is related to both decision trees and non-random game trees. To handle the chance nodes appearing in these game trees, new pruning techniques have been developed. By using *gamma-A* and *gamma-B pruning* the exact value (expected value) of the game is computed more efficiently than without these. In several cases the expected value is only an average value, and there is no game which results in that outcome.

Games of chance are more complicated than non-random games, therefore further analysis is needed. As we have seen in Section 4, the expected value of a game with chance nodes does not provide enough information if, for instance, the game is played iteratively. Although only the whole distribution (containing the probabilities of all possible values) has all the information, we believe that some significant indicators can effectively help in finding an optimal strategy.

As important and useful examples of such indicators, we have introduced the *minimal guaranteed value* and the *maximal possible value by choice*, and have shown how can these help in making better decisions. Of course, the expected value is still the most important indicator, but other ones can help in finding the correct decision. There are several open questions in this area. Future research work is needed to develop appropriate indicators, together with efficient computation algorithms and usage scenarios.

Acknowledgements

The helpful comments and advice of the reviewers are gratefully acknowledged. This work is partly supported by the grant of the Hungarian National Foundation for Scientific Research (OTKA T049409), by a grant of Ministry of Education, Hungary and by the Öveges programme of the Agency for Research Fund Management and Research Exploitation (KPI) and National Office for Research and Technology (NKTH).

References

[1] Fekete, I., Gregorics, T. and Nagy, S. Introduction to Artificial Intelligence: Heuristic Graph Search, Two Player Games, Automatic Theorem Proving (in Hungarian) LSI Oktatóközpont, Budapest, 1990.

[2] Futó, I., editor *Artificial Intelligence* (in Hungarian). Aula Kiadó, Budapest, 1999.

[3] Knuth, D. E. and Moore, R. W. An analysis of Alpha-Beta pruning. *Artificial Intelligence* 6:293–326, 1975.

[4] Lakatos, G. and Nagy, B. Games with few players. In Csőke, L., Olajos P., Szigetváry P. and Tómács T., editors *6th International Conference on Applied Informatics*, Volume II, pages 187–196, Eger, 2004.

[5] Luce, R. D. and Raiffa, H. *Games and Decisions: Introduction and Critical Survey.* Dover Publications, New York, 1989.

[6] von Neumann, J. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen* 10:295–320, 1928.

[7] von Neumann, J. and Morgenstern, O. *The Theory of Games and Economic Behavior.* Princeton University Press, 2nd edition, 1947.

[8] Resnik, M. D. *Choices: An Introduction to Decision Theory.* University of Minnesota Press, 1987.

[9] Rich, E. and Knight, K. *Artificial Intelligence.* McGraw-Hill, New York, 1991.

[10] Russell, S. and Norvig, P. *Artificial Intelligence, a Modern Approach.* Prentice-Hall, 1995. (2nd edition, 2003.)

Acta Cybernetica 18 (2007) 193–212.

Using Genetic Algorithms in Computer Vision: Registering Images to 3D Surface Model*

Zsolt Jankó† Dmitry Chetverikov† and Anikó Ekárt‡

Abstract

This paper shows a successful application of genetic algorithms in computer vision. We aim at building photorealistic 3D models of real-world objects by adding textural information to the geometry. In this paper we focus on the 2D–3D registration problem: given a 3D geometric model of an object, and optical images of the same object, we need to find the precise alignment of the 2D images to the 3D model.

We generalise the photo-consistency approach of Clarkson et al. who assume calibrated cameras, thus only the pose of the object in the world needs to be estimated. Our method extends this approach to the case of uncalibrated cameras, when both intrinsic and extrinsic camera parameters are unknown. We formulate the problem as an optimisation and use a genetic algorithm to find a solution.

We use semi-synthetic data to study the effects of different parameter settings on the registration. Additionally, experimental results on real data are presented to demonstrate the efficiency of the method.

Keywords: photo-consistency, uncalibrated images, photorealistic models

1 Introduction

Building photorealistic 3D models of real-world objects is a fundamental problem in computer vision and computer graphics. During the last years a number of ambitious projects [4, 17, 25] have been started around the world to digitise cultural heritage objects. Exhibiting 3D models of these objects in a virtual museum provides easy access to them. Furthermore, 3D models of real objects can also be used for surgical simulations in medical imaging, for e-commerce, architecture or entertainment (movies, computer games).

*This work was supported by EU Network of Excellence MUSCLE (FP6-507752)

†Computer and Automation Research Institute, Budapest, Kende u. 13-17, H-1111 Hungary and Eötvös Loránd University, Budapest; E-mail: {janko,csetverikov}@sztaki.hu

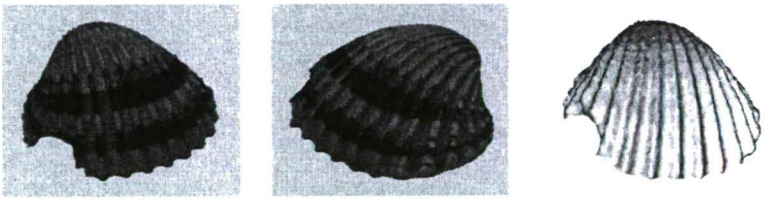
‡Aston University, School of Engineering and Applied Science, Computer Science, Aston Triangle, B4 7ET Birmingham, United Kingdom; E-mail: A.Ekart@aston.ac.uk

Photorealistic 3D models must have precise geometry as well as detailed texture on the surface. Active and passive methods for creating such models are discussed in [35]. The methods are based on different principles. They use different techniques to reconstruct the object surface, acquire its texture and map the texture onto the surface. The geometry can be measured by various methods of computer vision. When precise measurements are needed, laser scanners are often used. However, most laser scanners do not provide texture and colour information. Even when they do, the data provided are not accurate enough. (See [35] for a detailed discussion.)

Whatever the sources of geometric and textural information are, the problem of data fusion, or registration, is to be addressed. In this paper we consider the case when the two sources are independent. We approach the problem of combining precise geometry with high quality images by using genetic algorithms.

A number of approaches to the above registration problem have been proposed. In [19] and [20] we introduced a novel method based on photo-consistency. The novelty of our method consists in using uncalibrated cameras—in contrast to Clarkson et al. [8] who need a calibrated setup—and applying a genetic algorithm. Below we describe the problem of photo-consistency based registration and give a summary of our approach.

The mathematical formulation of the registration problem is the following. Two input images, I_1 and I_2, and a 3D model are given. They represent the same object. (See an example in Figure 1.) The only assumptions about the environment are that the lighting conditions are fixed and the cameras have identical sensitivity[1]. All other camera parameters may differ and are unknown. The 3D model consists of a 3D point set $\mathcal{P}$ and a set of normal vectors assigned to the points. $\mathcal{P}$ is obtained by a hand-held 3D scanner and then triangulated by the robust algorithm of Kós [22]. This algorithm provides the normal vectors as well.



Images 3D model

Figure 1: Shell dataset.

To project the object surface to the image plane, the finite projective camera model [14] is used: $\mathbf{u} \simeq P\mathbf{X}$, where $\mathbf{u}$ is an image point, P the 3×4 *projection matrix* and $\mathbf{X}$ a surface point. ($\simeq$ means that the projection is defined up to an unknown scale.)

The task of registration is to determine the precise projection matrices, P_1 and P_2, for both images. The projection matrix P has 12 elements but only 11 degrees of

[1]The latter can be easily achieved if the images are taken by the same camera.

freedom, since it is up to a scale factor. We denote the collection of the 11 unknown parameters by p, which represents the projection matrix P as an 11-dimensional parameter vector.

Values of p_1 and p_2 are sought such that the images are *consistent* in the sense that the corresponding points—different projections of the same 3D point—have the same colour value. Note that the precise mathematical definition is valid only when the surface is Lambertian, that is, the incoming light is reflected equally to every direction on the surface. This is usually true for diffuse surfaces. Formally, we say that images I_1 and I_2 are consistent by P_1 and P_2 (or p_1 and p_2) if for each $X \in \mathcal{P}$: $\mathbf{u}_1 = P_1\mathbf{X}$, $\mathbf{u}_2 = P_2\mathbf{X}$ and $I_1(\mathbf{u}_1) = I_2(\mathbf{u}_2)$. (Here $I_i(\mathbf{u}_i)$ is the colour value in point $\mathbf{u}_i$ of image I_i.) This type of consistency is called *photo-consistency* [8, 23].

The photo-consistency holds for accurate estimates for p_1 and p_2. Inversely, misregistered projection matrices mean much less photo-consistent images. The cost function introduced in [20] is the following:

$$C_\phi(p_1, p_2) = \frac{1}{|\mathcal{P}|} \sum_{\mathbf{X} \in \mathcal{P}} \|I_1(P_1\mathbf{X}) - I_2(P_2\mathbf{X})\|^2 . \tag{1}$$

Here ϕ stands for *photo-inconsistency* while $|\mathcal{P}|$ is the number of points in $\mathcal{P}$. Difference of the colour values $\|I_1 - I_2\|$ can be defined by a number of different colour models. (Details are discussed in section 5.1.) Finding the minimum of the cost function (1) over p_1 and p_2 yields estimates for the projection matrices.

In spite of the simplicity of the cost function $C_\phi(p_1, p_2)$, finding the minimum is a difficult task. Due to the 22-dimensional parameter space and the unpredictable shape of $C_\phi(p_1, p_2)$, the standard local nonlinear minimisation techniques failed to provide reliable results. We have tested a number of widely used optimisation methods: Newton-like methods, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method and the Levenberg-Marquardt algorithm. Experiments have shown that local search techniques terminate every time in local minima quite far from the expected global optimum.

The global nonlinear optimisation technique of Csendes [9] has also been tested. However, the stochastic optimisation method did not yield acceptable results either. The randomness of a stochastic method is excessive, and it does not save nearly good solutions. In contrast, elitist genetic algorithms preserve the most promising results and try to improve them. (Running a GA without elitism yields unstable and imprecise results, similarly to the stochastic optimisation.)

The methods mentioned above and other modern techniques, such as simulated annealing and tabu search process one single solution. In addition to performing a local search, simulated annealing and tabu search have specific built-in mechanisms to escape local optima. In contrast, genetic algorithms work on a population of potential solutions, which compete for survival. The competition is what makes GAs essentially different from single solution processing methods [26]. Consequently we decided to apply a genetic algorithm, as a time-honoured global search strategy.

Parts of this work have already been presented in papers [19, 20] and [21]. In this study we present the complete method, including a detailed discussion of

implementation problems, as well as the analysis of the method by systematic tests using different genetic settings and different colour models.

The structure of this paper is as follows. In section 2 we give an overview of genetic algorithms and related work on the 2D–3D registration problem. Section 3 presents our registration method based on a genetic algorithm, while section 4 discusses implementation details. In section 5 the method is analysed by tests on semi-synthetic data using different parameter settings. Experimental results on real data are also shown. Finally, section 6 concludes the paper by summarising its contribution.

2 Overview

2.1 Genetic Algorithms

To make this study more accessible, we need to devote a section to a brief overview of genetic algorithms, without aiming at completeness. For details the reader is referred to [12] or [2, 3].

Genetic algorithm is a global search technique that imitates natural biological evolution: the algorithm, starting from an initial population of potential solutions and preserving the best individuals, produces new population after new population, obtaining better and better approximations to a solution. At each generation, individuals are selected and bred together, resulting in a new set of approximations. The higher the level of fitness of an individual, the greater its chance of being selected. This fitness-driven selection leads to the evolution of populations of individuals that are better than the populations of their ancestors.

In nature individuals are determined by their *genes* in their *chromosomes*. In computing genes and chromosomes can be represented by strings: lacing the strings of the genes sequentially gives the string of the chromosome. The most commonly used alphabet of the strings is binary {0,1}, but other alphabets are also used, e.g., integer or real-valued numbers, depending on which is the most suitable for the given problem. Note that further on we shall use the word *allele* instead of gene, which is a possible variant of the same gene occupying a position (locus) in a chromosome.

To improve the population, better individuals should have larger chance to be selected than worse individuals. The goodness of an individual is given by a *fitness function*. Various *selection strategies* can be used, those using the fitness values (e.g., roulette wheel selection), or the simple uniform selection, that does not use the level of fitness.

Selected individuals are "mated" and new individuals are produced from them, for instance, by interchanging their corresponding alleles. The method of mating is determined by the selected *crossover* strategy. New individuals can also be created by *mutation*. The mutation strategy determines how an individual can be mutated. For instance, in Gaussian mutation the value of a selected allele is changed by a random value taken from a Gaussian distribution.

The simple GA cannot guarantee the improvement of the population. However, carrying the best individual(s) over to the next generation assures that it will not be worse either. This behaviour of GA is referred to as *elitism.*

It is important to emphasise that GA is non-deterministic: different runs yield different results depending on the seed of the random number generator. However, when the problem does not have one single solution, or when different solutions close to the best one are acceptable, GA is useful and often works better than traditional methods.

2.2 Related Work

Several 2D–3D registration methods exist in computer vision and its medical applications. Most of these methods are based on corresponding feature pairs: features are extracted both from the 3D surface and in the images, and correspondences are searched for. The simplest features are points [10]. Haider and Kaneko [13] look for edges both in 2D and in 3D, and define a 3D edge as a set of 3D surface points which is a 2D edge in the projected space. Stamos et al. [31] localise 2D and 3D lines and search for correspondences. This method has limited applicability, but can be useful when the objects are buildings with many line features. Ikeuchi et al. [17] also use lines and edges to calibrate cameras. The disadvantages of feature-based methods are that features are often difficult to localise precisely in 3D and, in addition, defining a similarity function between 2D and 3D features is not easy.

Another approach is to use the contour or the shape of the object to match to its projection. Hernández [16] defines a silhouette coherence criterion, but he does not use the 3D model of the object. 3D reconstruction from silhouettes and camera calibration are accomplished simultaneously. Such methods are more precise than feature-based methods, but in case of symmetric objects they are completely useless.

Intensity-based techniques can also be applied to align the images to the 3D model. In order to find correspondences, colour information of image pixels can be used as well as constraints on the gradient. The method of Umeda et al. [32] is based on range intensity images. They use a special range sensor that measures the property of the reflected light. The amount of the reflected light is related to the reflectance ratio of the measured point, thus the obtained image describes the reflectance of the object. This image is referred to as range intensity image. Colour images taken by a camera are registered to the range images by using constraints on the gradients of the colour images and of the range intensity images.

Viola et al. [33] search for the alignment of a 3D model and an optical image by maximising their mutual information. They use a geometric transformation that maps model points to image points, and also use an imaging function describing lighting conditions, surface properties, and imaging device characteristics. The registration problem can then be formulated as maximising the mutual information between the optical image intensities and the surface normal vectors of the model.

Leventon et al. [24] and Clarkson et al. [8] have shown that a registration algorithm based on maximising the mutual information can be improved by using multiple rather than single images. The algorithm in [8] applies photo-consistency

to find precise registration of 2D optical images of a human face to a 3D surface model. It uses calibrated images, thus the problem is reduced to estimating the pose of the cameras. Our method generalises this approach to the case of uncalibrated cameras, when both intrinsic and extrinsic parameters are unknown.

Genetic Algorithms in Registration Methods

None of the methods mentioned above use genetic algorithms, since the optimisation problems they consider are easier and are solved faster by conventional non-linear iterative strategies. In our case the size of the parameter space and the complexity of the cost function motivated the use of genetic algorithm-based optimisation.

While the application of genetic algorithms in 2D–3D registration methods has not been significant so far, several methods use evolutionary techniques to register 3D data or range images. In [29] Renner and Ekárt provide a summary of genetic algorithms in computer aided design. Jacq and Roux [18] use GAs for registration of 3D medical images. GAs are also used to register 3D surfaces [5] as well as range images [7]. In contrast to the 2D–3D registration, numerous methods exist to precisely register 3D data by iterative algorithms like the Iterative Closest Point and its variants [6]. Here, the task of GA-based methods is usually to automatically provide rough pre-registration of the surfaces needed by the iterative methods as an initial state close to the solution.

3 Genetic Algorithm-Based Optimisation

In section 1 we have formulated the registration problem and introduced the cost function. In this section the optimisation method is described.

Incorporating domain knowledge always makes GAs more effective. To narrow the search domain and accelerate the method, it is worth starting the search from a good initial state. We decided to pre-register the images and the 3D model manually, since this operation is simple and fast compared to the 3D scanning, which is also done manually. Our assumption was that the photo-consistency based registration would make the result more accurate. The tests justify this assumption.

Figure 2 summarises the proposed two-step method. The manual pre-registration provides the rough estimates P_1^0 and P_2^0 of the projection matrices P_1 and P_2, respectively. Then P_1^0 and P_2^0 are refined by minimising the photo-consistency based cost function (1) by a genetic algorithm. Note that each image is pre-registered to the 3D model *separately*, while the final precise registration involves *simultaneous* registration of both images to the model.

Designing the genetic algorithm for a given problem needs careful consideration. For general practical advice, the reader is referred to [26]. We use fixed-length vectors of bounded real numbers as representation. The individuals of the population are chosen from the neighbourhood of the parameter vector obtained by the manual pre-registration. The individual that encodes the pre-registered parameter vector is inserted in the initial population to avoid losing it. The values of the genes of the

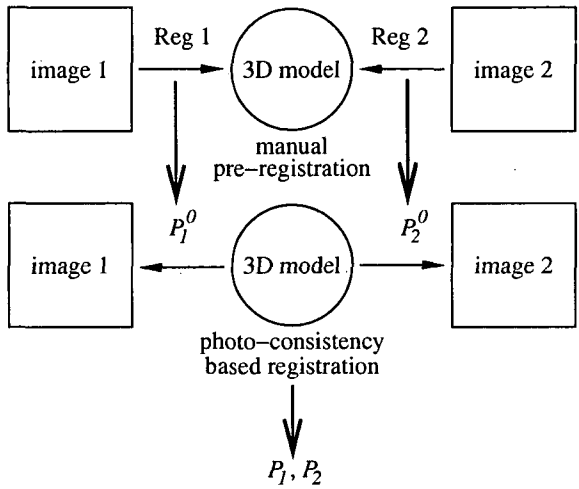


Figure 2: Block-diagram of proposed method.

remaining individuals are from the intervals defined by the pre-registered values plus a margin of $\pm\epsilon$. In our experiments ϵ was set to values between 1% and 3%, depending on the meaning and the importance of the corresponding parameter. For instance, small changes in parameter principal point can yield great deformations in projection, hence the interval of this parameter is set to $\pm 0.5\%$, in contrast to the focal length, where the interval is $\pm 2\%$. Details of camera parameters will be discussed in section 4.2.

During the initialisation the individuals are pre-selected: the useless individuals for which the cost function yields an extreme value are omitted. This prevents the genetic algorithm from jumping around the search space; since the aim of the method is to refine the initial state, small changes in the values are sufficient. Nevertheless, we tested the method without this restriction, as well, but the convergence was slower and the results were worse.

To avoid premature convergence we decided to run the algorithm three times: the algorithm starts three times from the beginning, preserving only the best individual from the previous step and re-initialising the whole population. An iteration is finished if N_g generations have been created, or if the best of the population has changed 10 times. (The setting of N_g is discussed later, in section 5.1.)

Our genetic algorithm is shown in Algorithm 1.

4 Implementation Details

We have chosen the GAlib package [34] written by Matthew Wall at the MIT, to implement our genetic algorithm-based method. For tests we mainly used the following parameter settings of GA as default: Steady state algorithm, Tournament

Algorithm 1 Genetic algorithm.

```
BEST ⟵ manual pre-registration.
for i = 1, ..., 3 do
    Generate initial population around BEST.
    BEST ⟵ best of the population.
    repeat
        Calculate cost function values.
        Apply genetic operators, create new population.
        BEST ⟵ best of the population.
    until (N_g generations created) or (BEST changed 10 times)
end for
```

selector, Swap mutation, Uniform crossover, 250 individuals in the population, mutation probability 0.1 and crossover probability 0.7.

4.1 Robustness

In registration and correspondence, robustness is a critical issue. Minimising the cost function (1) is a least-squares method, therefore it is not robust, due to the inconsistencies produced by outliers, typically, by occluded points. In [8], the visibility is checked by ray tracing, but here we use surface normals for this purpose. Our implementation is less accurate but much faster, which is more important in this case. The essence of our algorithm is to discard the point when the scalar product of the normal vector and the unit vector pointing towards the camera falls below a threshold. The product is the cosine of the angle between the two vectors. Typically, the threshold is set at 0.5, which discards some mutually visible points, but still leaves enough points for reliable registration. In computer vision this method is usually referred to as *backface culling*.

Backface culling works well when the surface is convex but fails when it is concave. The 3D models we use at registration are reduced to contain only 1000–1500 points, which is usually enough to obtain good result. It means that in most cases the input 3D models are rough and smooth, and they are nearly convex. However, it is clear that the trivial method for checking visibility cannot guarantee that all invisible points will be filtered out. The remaining false points are considered as outliers, as well as the false inconsistencies caused by texture periodicity or the object boundary.

To suppress the remaining outliers, the cost function (1) was modified in a robust manner. Two variants of modification were considered: the Trimmed Squares (TS) and the α-trimmed mean [28]. Both techniques have a single parameter, α. In TS, α is the rate of the largest squares which are discarded. In the α-trimmed mean, both smallest and largest values are rejected: when α is close to 0.5, the median is used. In our experiments, we used $\alpha = 0.2$.

In attempts to improve the method, we have tested a few other cost functions. However, the variance of the colour values [8] or the Modified Normalised Cross

Correlation [30] yielded worse results than the robust least-squares described above.

The sizes of our test images are 512×512 or 1024×1024. It seemed reasonable to reduce the size and apply image pyramids for the registration, but the results did not improve significantly.

4.2 Constraints on the Camera Model

As already mentioned, our original method applied optimisation in the full 22-dimensional parameter space. The size of the space and the non-smoothness of the cost function are two critical problems that make the search difficult and time-consuming despite the restrictions due to the manual pre-registration. To improve the efficiency of the optimisation process, we impose some reasonable constraints on the camera model, as suggested in [14].

Note that using the finite projective camera model without camera distortion is already a constraint which works well in practice. The projection matrix can be decomposed as $P = K[R \mid -R\underline{t}]$, where K is the 3×3 camera calibration matrix and the 3×3 rotation matrix R describes the orientation, the 3-dimensional translation vector $\underline{t}$ the location of the camera. The camera calibration matrix can be expressed in form

$$K = \begin{bmatrix} \alpha_u & s & -u_0 \\ & \alpha_v & -v_0 \\ & & 1 \end{bmatrix}. \tag{2}$$

Here α_u and α_v represent the focal length of the camera in terms of pixel dimensions in the u and v directions of the image plane, respectively, s is the *skew* parameter and (u_0, v_0) is the so-called *principal point*. For most cameras the skew parameter is zero. It is also usual to assume that the pixels are squared, that is the ratio of α_u and α_v is equal to 1. Thus the camera calibration matrix can be simplified to the so-called *pinhole camera model*:

$$K = \begin{bmatrix} f & & -u_0 \\ & f & -v_0 \\ & & 1 \end{bmatrix}, \tag{3}$$

with focal length f.

These simplifications reduce the number of the degrees of freedom from 22 to 18. Although the decrease is not large, in this case every reasonable reduction is important. Therefore we also applied a commonly used assumption, that the principal point is close to the image centre. This assumption does not reduce the number of the parameters, but the search space becomes more restricted.

In the previous sections we did not specify the ϵ values for the intervals of the genes. Considering the simplifications detailed above, the values we use are the following:

- focal length: $\pm 2\%$
- principal point: $\pm 0.5\%$

- camera translation: ±3%
- camera rotation: ±1°.

5 Experiments

5.1 Quantitative Assessment on Semi-Synthetic Data

To quantitatively assess the results, the algorithm was run on semi-synthetic data with ground truth. We obtain these data by covering the triangular mesh of the original Shell, Frog and Bear datasets (see Figures 3, 4 and 5) with different textures. The textures were obtained from the photos of the original objects. Two views of these objects produced by a visualisation program provide the input images for which the projection matrices are completely known.

The projection error is measured as follows: the 3D point set $\mathcal{P}$ is projected onto the image planes by both the ground truth and the estimated projection matrices, and then the average distance between the corresponding image points is calculated. Formally, if P_i^G are the ground truth, P_i the estimated projection matrices, then

$$E(P_1, P_2) = \frac{1}{2} \sum_{i=1,2} \frac{1}{|\mathcal{P}|} \sum_{\mathbf{X} \in \mathcal{P}} \left\| P_i^G \mathbf{X} - P_i \mathbf{X} \right\|. \tag{4}$$

By this metric the average error of the manual pre-registration is 18–20 pixels for the Shell and 12–15 pixels for the Bear and the Frog. Tables 1–8 show the results of the method executed with different settings. In each case 10 runs were performed and the mean and the confidence interval were calculated.

Next we show the influence of different colour models and different genetic settings on the registration.

Colour Models

An important question in the case of methods which compare colours is how to calculate colour differences, which colour model provides the best result. We have tested four different models: RGB, XYZ, CIE LAB and CIE LUV. Each model consists of three components, and colour differences were calculated as the simple sum of squared differences in the three components. Table 1 shows the results for the Shell, the Bear and the Frog datasets.

In the literature CIE LUV and CIE LAB are usually used to compare colours, since the perception of colour difference in RGB and XYZ is highly non-uniform. Indeed, our tests have shown that uniform models perform slightly better, but the difference in projection error is not significant. In our experimental data the illumination changes are small.

Tests were run both on diffuse and specular data. As it is expected, the error of specular data is significantly greater than that of the diffuse data. This is obvious, since photo-consistency supposes Lambertian reflection. However, one can see

that specular errors do not grow extremely and remain below a reasonable limit, due to the robustness of the method and the application of geometric constraints discussed above. If the initial model is relatively good, and consequently geometric constraints are correct, then registration of a specular dataset is fairly good, see the Frog in table 1.

Table 1: Projection error using different colour models.

	Shell		Bear		Frog	
Col. model	Diffuse	Specular	Diffuse	Specular	Diffuse	Specular
RGB	7.3 ± 0.4	8.8 ± 0.7	7.8 ± 0.5	9.6 ± 0.2	4.6 ± 0.3	5.9 ± 0.4
XYZ	6.7 ± 0.5	11.4 ± 2.2	7.8 ± 0.6	9.3 ± 0.2	4.9 ± 0.3	5.7 ± 0.6
CIE LAB	6.9 ± 0.2	8.9 ± 1.3	6.5 ± 0.5	10.0 ± 0.6	5.2 ± 0.4	6.0 ± 0.3
CIE LUV	6.9 ± 0.5	8.9 ± 1.0	6.7 ± 0.5	9.8 ± 0.5	4.4 ± 0.2	5.7 ± 0.4

Genetic Settings

A number of tests have been carried out to check the effects of different genetic settings on the registration. First, we tried two different *algorithms*: the simple GA of Goldberg [12] and the steady state GA. In the simple GA, we use non-overlapping populations: in each generation an entirely new population is created by crossover and mutation. We also use elitism: the best individual is carried over to the next generation. Elitism is set during all the tests in order to avoid losing good results.

The steady state algorithm is similar to the one described by De Jong [11]. Here, we use overlapping populations with an overlap of 25%. Each generation a temporary population of individuals is created and added to the previous population. Then the worst individuals are removed to reduce the population to its original size.

The results shown in table 2 are not surprising: the steady state algorithm performs better than the simple GA. The difference in speed can be explained by the termination criterion of the algorithm. As we mentioned above, each iteration terminates if N_g generations were created or if the best individual of the population changed 10 times. The steady state algorithm preserves a number of the best individuals of the population, hence it can create better new individuals than the simple GA, which preserves only the very best individual. Our tests have shown that in the case of the steady state algorithm the best individual changed every 4–5 generations, while in the case of the simple GA this number was 10–15. Therefore, the steady state algorithm terminates much sooner, after fewer iterations.

In the next test three different *selectors* were tried: roulette wheel selector, tournament selector and uniform selector (table 3). *Roulette wheel selector* picks an individual based on its fitness score relative to the rest of the population. The higher the score, the more likely an individual will be selected. *Tournament selector* uses the roulette wheel method to choose two individuals, then picks the

Table 2: Results of different genetic algorithms.

	Projection error			Time (min)		
Algorithm	Shell	Bear	Frog	Shell	Bear	Frog
Simple	7.4 ± 1.0	6.7 ± 0.4	8.9 ± 1.0	11.1 ± 0.2	7.8 ± 0.3	9.9 ± 0.2
Steady State	6.3 ± 0.8	5.1 ± 0.3	6.6 ± 0.5	6.4 ± 0.3	4.5 ± 0.3	4.7 ± 0.4

one with the higher score. *Uniform selector* selects each individual with uniform probability. According to our tests, the non-uniform selectors are slightly better than the uniform selector; however, the difference is not significant.

Table 3: Results for different selectors.

	Projection error			Time (min)		
Selector	Shell	Bear	Frog	Shell	Bear	Frog
Roul.wheel	6.0 ± 0.7	5.1 ± 0.4	6.6 ± 0.8	6.4 ± 0.3	4.6 ± 0.3	4.8 ± 0.3
Tournament	6.0 ± 0.5	4.6 ± 0.1	6.1 ± 0.8	6.3 ± 0.3	4.5 ± 0.3	4.3 ± 0.3
Uniform	6.7 ± 0.7	4.9 ± 0.4	7.1 ± 0.6	6.6 ± 0.3	4.6 ± 0.3	5.0 ± 0.3

Next we have tested a number of different types of *mutation* and *crossover*. Here we give only a short description of the operators; for details the reader is referred to [15].

In *Gaussian* mutation the value of the selected allele is changed by a random value taken from a Gaussian distribution (and then adjusted to the range). *Flip* mutation gives a random value to a randomly chosen allele, considering bounds. *Swap* mutation picks two alleles at random and exchanges their values.

In *uniform* crossover the value of each allele of the offspring is randomly chosen from the same alleles of the parents. *Even-odd* crossover considers two operators: for even crossover, the even alleles of the offspring are taken from one parent and the odd alleles from the other parent. For odd crossover, the opposite is done. In *one-point* crossover the two parents' codes are cut at a randomly chosen position and the end-parts of the parents are exchanged to produce two offspring. In *two-point* crossover two crossover points are selected at random and the middle part of the parents' code is exchanged.

In *partial-match* crossover a random position is picked. Let us suppose the values in this position in the two parents are x and y. Then x and y are interchanged throughout both parents' code. The operation is repeated several times. In *order* crossover about half of the elements of the offspring are picked from one parent, keeping the order, and the rest from the other parent, also keeping the order. *Blend* crossover chooses a new value for the offspring allele in the neighbourhood of parent values. Suppose $x < y$, then the value for the offspring is uniformly chosen from the interval $[x - \alpha(y - x), x + \alpha(y - x)]$. *Arithmetic* crossover creates an offspring "between" the two parents. If x and y are the parental allele values, the offspring

will become $z = x + (1 - \alpha)y$ where $\alpha \in [0, 1]$.

Table 4 shows that different mutations yield very similar error. Flip mutation is slightly better, swap mutation is slightly faster than the others. In the case of the crossovers (table 5), blend and arithmetic crossovers perform better than the others.

Table 4: Results for different types of mutation.

	Projection error			Time (min)		
Mutation	Shell	Bear	Frog	Shell	Bear	Frog
Gaussian	6.4 ± 0.5	4.7 ± 0.3	5.9 ± 0.6	9.8 ± 0.3	6.2 ± 0.4	6.1 ± 0.3
Flip	6.0 ± 0.7	4.7 ± 0.2	5.7 ± 0.5	9.5 ± 0.4	6.2 ± 0.5	6.6 ± 0.6
Swap	6.0 ± 0.6	5.0 ± 0.3	6.4 ± 0.8	6.4 ± 0.3	4.3 ± 0.3	4.3 ± 0.3

Table 5: Results for different types of crossover.

	Projection error			Time (min)		
Crossover	Shell	Bear	Frog	Shell	Bear	Frog
Uniform	6.1 ± 0.7	5.4 ± 0.8	6.6 ± 0.9	6.4 ± 0.4	4.5 ± 0.5	4.4 ± 0.6
Even-Odd	6.4 ± 1.0	5.6 ± 0.7	7.3 ± 1.3	6.2 ± 0.4	4.5 ± 0.5	4.7 ± 0.4
One-Point	6.7 ± 0.8	5.5 ± 0.9	7.0 ± 1.3	6.6 ± 0.4	4.8 ± 0.4	4.9 ± 0.6
Two-Point	6.7 ± 1.2	5.4 ± 0.4	6.6 ± 0.5	6.5 ± 0.5	4.8 ± 0.4	4.5 ± 0.5
Partial-match	8.7 ± 0.8	7.1 ± 1.2	10.0 ± 2.7	3.0 ± 0.1	3.4 ± 0.4	4.0 ± 0.1
Order	8.2 ± 1.0	6.8 ± 0.8	10.5 ± 1.7	3.3 ± 0.4	3.1 ± 0.3	3.1 ± 0.3
Blend	6.1 ± 0.9	4.9 ± 0.4	6.0 ± 0.6	6.4 ± 0.4	4.6 ± 0.6	4.3 ± 0.4
Arithmetic	5.0 ± 0.5	5.5 ± 0.8	4.9 ± 0.4	6.3 ± 0.4	4.6 ± 0.4	4.0 ± 0.4

Table 6 is self-evident. Increasing the *population size* improves the result and makes the method slower. The difference in error between populations with 250 or 500 individuals is insignificant, hence setting the size of the population to a vicinity of 250 is reasonable. On the other hand, convergence is reached somewhere between 100 and 200 generations, so the results do not significantly depend on whether N_g is set to 100 or 200. Therefore we used 100 generations.

Table 6: Results for different population sizes.

	Projection error			Time (min)		
Pop.size	Shell	Bear	Frog	Shell	Bear	Frog
100	6.8 ± 0.8	6.6 ± 0.8	8.2 ± 1.5	2.5 ± 0.3	1.8 ± 0.3	1.8 ± 0.3
250	6.0 ± 0.5	4.9 ± 0.4	6.1 ± 0.4	6.5 ± 0.5	4.4 ± 0.3	4.2 ± 0.3
500	5.8 ± 0.4	4.6 ± 0.1	6.2 ± 0.5	12.2 ± 0.5	9.2 ± 0.3	8.5 ± 0.6

The results of tables 7 and 8 are also expected. More frequent mutation increases the randomness, hereby the error and the duration as well. Having the crossover probability 0.9 or 0.7 yields almost the same error, however in the case of 0.7 the method is slightly faster.

Table 7: Results for different mutation probabilities.

	Projection error			Time (min)		
Mut.prob.	Shell	Bear	Frog	Shell	Bear	Frog
0.1	6.0 ± 0.5	4.7 ± 0.2	5.9 ± 0.9	6.3 ± 0.3	4.7 ± 0.3	4.2 ± 0.3
0.3	9.9 ± 2.1	7.1 ± 0.9	9.6 ± 1.4	2.8 ± 0.3	2.9 ± 0.2	3.2 ± 0.3
0.5	9.2 ± 1.9	7.5 ± 1.0	10.4 ± 1.1	2.0 ± 0.1	2.2 ± 0.3	2.7 ± 0.4

Table 8: Results for different crossover probabilities.

	Projection error			Time (min)		
Cross.prob.	Shell	Bear	Frog	Shell	Bear	Frog
0.9	5.8 ± 0.6	4.6 ± 0.2	5.8 ± 0.4	6.4 ± 0.3	4.9 ± 0.2	5.2 ± 0.3
0.7	5.9 ± 0.6	5.1 ± 0.3	6.3 ± 0.6	5.4 ± 0.3	4.5 ± 0.3	4.6 ± 0.3
0.5	6.4 ± 1.0	5.3 ± 0.5	7.4 ± 0.6	4.6 ± 0.3	3.9 ± 0.2	4.0 ± 0.4

Based on these tests we can conclude that one reasonable setting is as follows: Steady state algorithm with Tournament selector, Flip mutation and Arithmetic crossover, with 250 individuals in the population, mutation probability 0.1 and crossover probability 0.7.

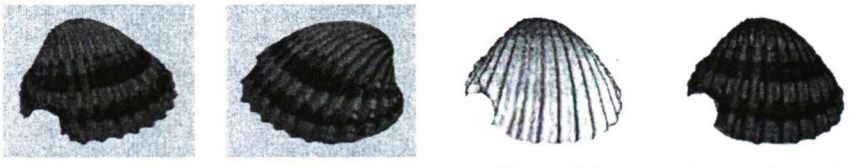
5.2 Results for Real Data

To test the efficiency of the method we also applied it to real data. A number of different datasets were used. Figures 3–7 show the Shell, the Frog, the Bear, the Cat and the Head datasets as well as the textured 3D models obtained. The Shell dataset is interesting because of the periodicity in shape and texture. The Frog and the Head are challenging as their textures are less visible and less characteristic. The Bear and the Cat have both characteristic shape and texture.

All 3D models were acquired in our laboratory using the ModelMaker [1] laser scanner, and the images were captured by a digital camera. The 3D models contain tens of thousands of points, but for registration the set was reduced to 1000–1500 points. We used the algorithm of Kós [22] for this just as for triangulation. The dimensions of the images are 512×512 or 1024×1024.

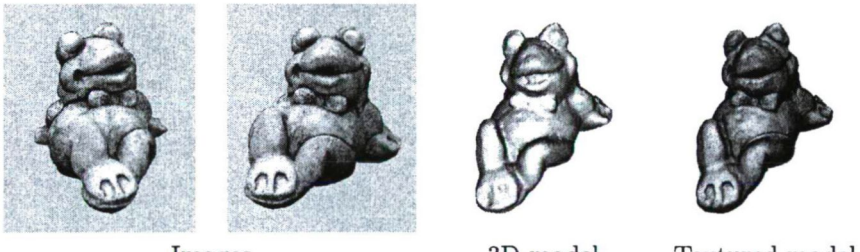
The precision of the registration can be best judged by looking at the stripes of the Shell, as well as the mouth, the eyes, the hand and the feet of the Bear and the Cat. In the case of the Shell both the shape and the texture are periodic, hence

precise registration is crucial for photorealism. One can see that the stripes of the texture are in the appropriate position.



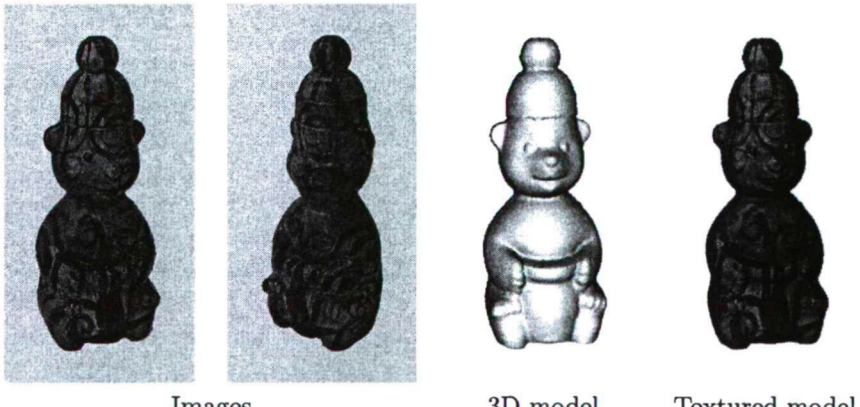
Images 3D model Textured model

Figure 3: Shell dataset and result of registration.



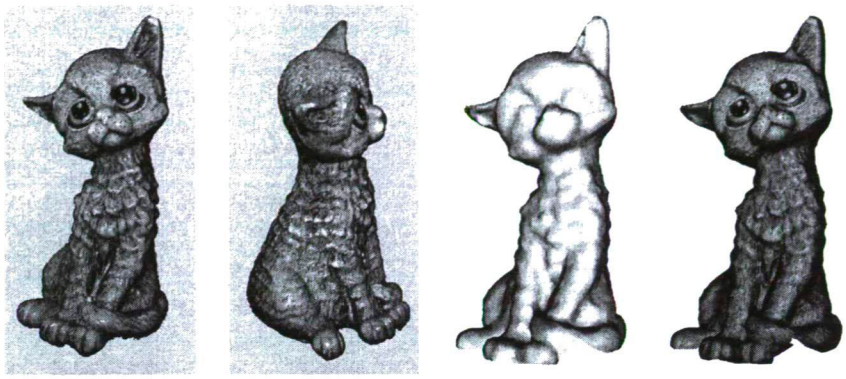
Images 3D model Textured model

Figure 4: Frog dataset and result of registration.



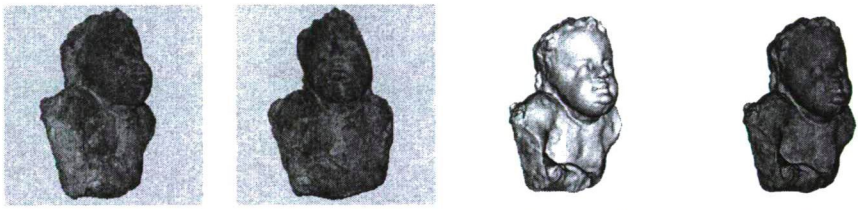
Images 3D model Textured model

Figure 5: Bear dataset and result of registration.



Images 3D model Textured model

Figure 6: Cat dataset and result of registration.



Images 3D model Textured model

Figure 7: Head dataset and result of registration.

6 Conclusion

In this paper we have presented a method for registering a pair of high-quality images to a 3D surface model using a genetic algorithm. The registration is performed by minimising a photo-consistency based cost function. The application of the genetic algorithm is reasonable, since the shape of the cost function is rough and unpredictable, with multiple minima. The results of previously studied standard local and global optimisation methods were not acceptable.

The aim of this study was to present the complete method in details. Besides the description of the optimisation method and the discussion of implementation details, a number of tests have been carried out to check the effects of different colour models and different genetic parameter settings on the registration. Different settings can lead to significantly different results. The tests were run on three different objects, by which we mean both different geometry and different textures. Results show strong analogy, verifying the general validity of the effects of parameter settings on the method. The tests have shown that by choosing the best settings the projection error of the registration can be decreased from 18–20 pixels,

which is the average error of the manual pre-registration, to 5–6 pixels for diffuse and to 8–10 pixels for specular surfaces. Figure 8 visualises the difference between the manual pre-registration and the photo-consistency based genetic registration. The quality of registration in the areas of the mouth, the eyes and the ears is visibly better after the genetic algorithm has been applied. Here again the difference is essential for photorealism.

Manual Genetic

Figure 8: Difference between manual pre-registration and genetic registration.

Choosing the best parameters for a particular problem requires a lot of experimentation or some sophisticated method. Studies on parameter control in GAs mostly consider controlling one aspect of the algorithm at a time and they use some form of self-adaptation. Michalewicz and Fogel [26] provide a detailed discussion on tuning the algorithm to the problem. Furthermore, there exist metamodelling techniques like the Response Surface Methodology (RSM) [27] to try out parameters in a systematic way. RSM uses quantitative data to describe how the parameters affect the response, to determine the interrelationships among the parameters and to describe the combined effect of all the parameters on the response. However, controlling multiple parameters simultaneously is a current research topic in GAs and is beyond the scope of this paper.

References

[1] 3D Scanners. Modelmaker. URL: `http://www.3dscanners.com/`.

[2] Beasley, David, Bull, David R., and Martin, Ralph R. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

[3] Beasley, David, Bull, David R., and Martin, Ralph R. An overview of genetic algorithms: Part 2, research topics. *University Computing*, 15(4):170–181, 1993.

[4] Bernardini, Fausto, Martin, Ioana M., and Rushmeier, Holly. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 07(4):318–332, 2001.

[5] Brunnström, K. and Stoddart, AJ. Genetic algorithms for free-form surface matching. In *Proc. 13^{th} International Conference on Pattern Recognition*, volume 4, pages 689–693, 1996.

[6] Chetverikov, Dmitry, Stepanov, Dmitry, and Krsek, Pavel. Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.

[7] Chow, Chi Kin, Tsui, Hung Tat, and Lee, Tong. Surface registration using a dynamic genetic algorithm. *Pattern Recognition*, 37:105–117, 2004.

[8] Clarkson, Matthew J., Rueckert, Daniel, Hill, Derek L.G., and Hawkes, David J. Using photo-consistency to register 2D optical images of the human face to a 3D surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1266–1280, 2001.

[9] Csendes, T. Nonlinear parameter estimation by global optimization – Efficiency and Reliability. *Acta Cybernetica*, 8:361–370, 1988.

[10] David, Philip, DeMenthon, Daniel, Duraiswami, Ramani, and Samet, Hanan. SoftPOSIT: Simultaneous pose and correspondence determination. In *Proc. 7^{th} European Conference on Computer Vision*, pages 698–714, 2002.

[11] De Jong, Kenneth Alan. *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[12] Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[13] Haider, Ali Md. and Kaneko, Toyohisa. Automated 3D–2D projective registration of human facial images using edge features. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(8):1263–1276, 2001.

[14] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[15] Haupt, Randy L. and Haupt, Sue Ellen. *Practical Genetic Algorithms.* Wiley-Interscience, 2004.

[16] Hernández, Carlos. *Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2004.

[17] Ikeuchi, Katsushi, Nakazawa, Atsushi, Hasegawa, Kazuhide, and Ohishi, Takeshi. The great Buddha project: Modeling cultural heritage for VR systems through observation. In *Proc. IEEE ISMAR03*, 2003.

[18] Jacq, JJ. and Roux, C. Registration of 3D images by genetic optimization. *Pattern Recognition Letters*, 16:823–841, 1995.

[19] Jankó, Zsolt and Chetverikov, Dmitry. Photo-consistency based registration of an uncalibrated image pair to a 3D surface model using genetic algorithm. In *Proc.* 2^{nd} *International Symposium on 3D Data Processing, Visualization & Transmission*, pages 616–622, 2004.

[20] Jankó, Zsolt and Chetverikov, Dmitry. Registration of an uncalibrated image pair to a 3D surface model. In *Proc.* 17^{th} *International Conference on Pattern Recognition*, volume 2, pages 208–211, 2004.

[21] Jankó, Zsolt, Chetverikov, Dmitry, and Ekárt, Anikó. Using a genetic algorithm to register an uncalibrated image pair to a 3D surface model. *International Journal of Engineering Applications of Artificial Intelligence*, 19:269–276, 2006.

[22] Kós, Géza. An algorithm to triangulate surfaces in 3D using unorganised point clouds. *Computing Suppl.*, 14:219–232, 2001.

[23] Kutulakos, K.N. and Seitz, S.M. *A Theory of Shape by Space Carving.* Prentice Hall, 1993.

[24] Leventon, M.E., Wells III, W.M., and Grimson, W.E.L. Multiple view 2D-3D mutual information registration. In *Proc. Image Understanding Workshop*, 1997.

[25] M. Levoy et al. The digital Michelangelo project. *ACM Computer Graphics Proceedings, SIGGRAPH*, pages 131–144, 2000.

[26] Michalewicz, Zbigniew and Fogel, David B. *How to Solve It: Modern Heuristics.* Springer, 2002.

[27] Myers, Raymond H. and Montgomery, Douglas C. *Response Surface Methodology.* John Wiley & Sons, Inc., 1995.

[28] Pitas, I. *Digital Image Processing Algorithms.* Prentice Hall, 1993.

[29] Renner, Gábor and Ekárt, Anikó. Genetic algorithms in computer aided design. *Computer Aided Design*, 35:709–726, 2003.

[30] Sara, Radim. Finding the largest unambiguous component of stereo matching. In *Proc. 7th European Conference on Computer Vision*, volume 2, pages 900–914, 2002.

[31] Stamos, Ioannis and Allen, Peter K. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, 88(2):94–118, 2002.

[32] Umeda, Kazunori, Godin, Guy, and Rioux, Marc. Registration of range and color images using gradient constraints and range intensity images. In *Proc. 17th International Conference on Pattern Recognition*, volume 3, pages 12–15, 2004.

[33] Viola, Paul and Wells III, William M. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[34] Matthew Wall. The GAlib genetic algorithm package.
URL: `http://lancet.mit.edu/ga`, 2003.

[35] Yemez, Y. and Schmitt, F. 3D reconstruction of real objects with high resolution shape and texture. *Image and Vision Computing*, 22:1137–1153, 2004.

Acta Cybernetica 18 (2007) 213–221.

Independent Subspace Analysis can Cope with the 'Curse of Dimensionality'

Zoltán Szabó* and András Lőrincz†

Abstract

We search for hidden independent components, in particular we consider the independent subspace analysis (ISA) task. Earlier ISA procedures assume that the dimensions of the components are known. Here we show a method that enables the non-combinatorial estimation of the components. We make use of a decomposition principle called the ISA separation theorem. According to this separation theorem the ISA task can be reduced to the independent component analysis (ICA) task that assumes one-dimensional components *and then* to a grouping procedure that collects the respective non-independent elements into independent groups. We show that non-combinatorial grouping is feasible by means of the non-linear f-correlation matrices between the estimated components.

Keywords: independent subspace analysis, non-combinatorial solution

1 Introduction

The technique called independent component analysis (ICA) and its independent subspace analysis (ISA) extension are in the focus of research interest for signal processing tasks. ICA applications include, among others: (i) feature extraction [4], (ii) denoising [6], (iii) processing of financial [11] and neurobiological data, e.g. fMRI, EEG, and MEG [12,26]. The ISA model is frequently applied for the analysis of EEG-fMRI signals [1].

Originally, ICA is one-dimensional in the sense that all sources are assumed to be independent real valued stochastic variables. The typical example of ICA is the so-called *cocktail-party problem*, where there are D sound sources and D microphones and the task is to separate the original sources from the observed mixed signals. Clearly, applications where not all, but only certain groups of the sources are independent may have high relevance in practice. In this case, independent sources can be multidimensional. For example, there could be independent groups

*Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary; E-mail: szzoli@cs.elte.hu

†Corresponding author; Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary; E-mail: andras.lorincz@elte.hu

of people talking about independent topics at a conference or more than one group of musicians may be playing at a party. This is the independent subspace analysis (ISA) extension of ICA.[1] Strenuous efforts have been made to develop ISA algorithms [1,3,5,7–9,13–15,18,19,22,24,25,27], where the theoretical problems concern mostly (i) the estimation of the entropy or of the mutual information, or (ii) joint block diagonalization.

Earlier ISA methods were constrained by assuming that the dimensions of the hidden components are known. Here, we show a non-combinatorial solution to the estimation of the dimensions. In the ISA problem one assumes temporally i.i.d. (independent and identically distributed) hidden sources. For the non i.i.d case, one may try the autoregressive assumption (see, e.g., [16] and references therein). This problem family is called independent process analysis (IPA). The method that we present here can be extended to IPA tasks by applying the innovation trick of [17].

The paper is built as follows: Section 2 formulates the problem domain. The estimation of the dimensions of the ISA components is described in Section 3. We illustrate our method in Section 4. Conclusions are drawn in Section 5.

2 The ISA Model

First, we define the ISA model. Assume that we have M hidden independent multidimensional and i.i.d random variables and that only the mixture of these M *components* is available for observation:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{1}$$

where $\mathbf{s}(t) := \left[\mathbf{s}^1(t); \ldots; \mathbf{s}^M(t)\right]$ is the vector concatenated form of the components $\mathbf{s}^m \in \mathbb{R}^{d_m}$. We assume that (i) for a given m, $\mathbf{s}^m(t)$ is i.i.d. in time t, (ii) there is at most a single Gaussian component amongst $\mathbf{s}^m$s, and (iii) $I(\mathbf{s}^1, \ldots, \mathbf{s}^M) = 0$, where I stands for the mutual information of the arguments. The total dimension of the components is $D := \sum_{m=1}^{M} d_m$. $\mathbf{A} \in \mathbb{R}^{D \times D}$ is the so-called *mixing matrix* that, according to our assumptions, is invertible. The goal of the ISA task is to uncover hidden components $\mathbf{s}^m$ (and the *separation matrix* $\mathbf{W} = \mathbf{A}^{-1}$) using the observations $\mathbf{x}(t)$ only. The ICA task is recovered when every components is of one-dimensional, i.e., if $d_m = 1$ $(m = 1, \ldots, M)$.

In the ISA model, we can assume without any loss of generality, that both the hidden source $\mathbf{s}$ and the observation $\mathbf{x}$ are white, that is, their expected values and covariances are $\mathbf{0}$ and $\mathbf{I}_D$, respectively. Here $\mathbf{I}_D$ denotes the D-dimensional identity matrix. Then:

- The $\mathbf{s}^m$ components are determined up to permutation and orthogonal transformation [23].

[1] ISA is also called multidimensional independent component analysis (MICA) [5] and group ICA [24] in the literature.

- One may assume that the separation matrix $\mathbf{W}$ is orthogonal: $\mathbf{W} \in \mathcal{O}^D := \{\mathbf{W} \in \mathbb{R}^{D \times D} | \mathbf{W}\mathbf{W}' = \mathbf{I}_D\}$ where $\mathcal{O}^D$ denotes orthogonal matrices of size $D \times D$ and $'$ stands for transposition.

3 Dimension Estimation of the Components in the ISA Task

Here we put forth a non-combinatorial solution that can uncover the the dimensions of the ISA components. We build our method onto (i) the ISA separation theorem [21,22] and (ii) the ISA cost function introduced in [19].

The ISA separation theorem, which was conjectured by Jean-François Cardoso [5], allows one to decompose the solution of the ISA problem, under certain conditions, into 2 steps: In the first step, ICA estimation is executed. In the second step, the ICA elements are grouped by finding an optimal permutation. Formally:

Theorem 1 (Separation Theorem for ISA). *Let* $\mathbf{y} = [y_1; \ldots; y_D] = \mathbf{W}\mathbf{x}$, *where* $\mathbf{W} \in \mathcal{O}^D$, $\mathbf{x} \in \mathbb{R}^D$ *is the whitened observation of the ISA model. Let* $\mathcal{S}^{d_m}$ *denote the surface of the* d_m*-dimensional unit sphere, that is* $\mathcal{S}^{d_m} := \{\mathbf{w} \in \mathbb{R}^{d_m} : \sum_{i=1}^{d_m} w_i^2 = 1\}$. *$H$ is Shannon's differential entropy.*

Presume that the $\mathbf{u} := \mathbf{s}^m$ *sources* $(m = 1, \ldots, M)$ *of the ISA model satisfy condition*

$$H\left(\sum_{i=1}^{d_m} w_i u_i\right) \geq \sum_{i=1}^{d_m} w_i^2 H(u_i), \forall \mathbf{w} \in \mathcal{S}^{d_m}, \tag{2}$$

and that the ICA cost function $J_{ICA}(\mathbf{W}) = \sum_{i=1}^{D} H(y_i)$ *has minimum over the orthogonal matrices in* $\mathbf{W}_{\mathrm{ICA}}$. *Then it is sufficient to search for the solution of the ISA task as a permutation of the solution of the ICA task. Using the concept of separation matrices, it is sufficient to explore forms*

$$\mathbf{W}_{\mathrm{ISA}} = \mathbf{P}\mathbf{W}_{\mathrm{ICA}},$$

where $\mathbf{P} \in \mathbb{R}^{D \times D}$ *is a permutation matrix to be determined, and* $\mathbf{W}_{\mathrm{ISA}}$ *is the ISA separation matrix.*

Sufficient conditions for Eq. (2) were eventually found by Szabó et al. (see [22] and references therein). Further, one can group the ICA components and can find the optimal permutation efficiently by means of the joint f-decorrelation (JFD) technique introduced in [19]. Roughly speaking, the JFD technique performs decorrelation over an $\mathcal{F}$ set of functions. In particular, the method aims the simultaneous block-diagonalization of covariance matrices $\mathbf{C}_f(\mathbf{W}) := cov(f[\hat{\mathbf{s}}(\mathbf{W})], f[\hat{\mathbf{s}}(\mathbf{W})])$ of all functions $f \in \mathcal{F}$, where blocks are d_m-dimensional.

However, the hidden components can be determined without knowing their dimensions, provided that the separation theorem holds. In this case, the estimated ICA elements correspond to the ISA components up to permutation. In other words, matrices $\mathbf{C}_f$ are block-diagonal with block size d_m apart from a *common*

permutation. Thus, the coupled components can be found by the following procedure. We say that two coordinates i and j are $\mathbf{C}^{\mathcal{F}}$-'connected' ($\mathbf{C}^{\mathcal{F}} := \sum_{f\in\mathcal{F}} |\mathbf{C}_f|$, $|\cdot|$ denotes absolute values for all coordinates) if $\max(C^{\mathcal{F}}_{ij}, C^{\mathcal{F}}_{ji}) > \epsilon$, where $\epsilon \geq 0$ and in the ideal case $\epsilon = 0$. Then we group the $\mathbf{C}^{\mathcal{F}}$-'connected' coordinates into separate subspaces as follows: (1) Choose an arbitrary coordinate i and group all $j \neq i$ coordinates to it which are $\mathbf{C}^{\mathcal{F}}$-'connected' with it. (2) Choose an arbitrary and not yet grouped coordinate. Find its connected coordinates. Group them together. (3) Continue until all components are grouped. This is the *gathering procedure* and it is fast. In the worst case, it is quadratic in the number of the coordinates.

4 Illustration

Here we illustrate how our method works. Test cases are introduced in Section 4.1. The quality of the solutions will be measured by the normalized Amari-error, the Amari-index (Section 4.2). Numerical results are presented in Section 4.3.

4.1 Databases

We define three databases to study our identification algorithm. The databases are illustrated in Fig. 1. In the *3D-geom* test $\mathbf{s}^m$s were random variables uniformly distributed on 3-dimensional geometric forms ($d = 3$). We chose 6 different components ($M = 6$) and, as a result, the dimension of the hidden source $\mathbf{s}$ is $D = 18$. The *celebrities* test has 2-dimensional source components generated from cartoons of celebrities ($d = 2$).[2] Sources $\mathbf{s}^m$ were generated by sampling 2-dimensional coordinates proportional to the corresponding pixel intensities. In other words, 2-dimensional images of celebrities were considered as density functions. $M = 10$ was chosen ($D = 20$). In the *ABC* database, hidden sources $\mathbf{s}^m$ were uniform distributions defined by 2-dimensional images ($d = 2$) of the English alphabet. The number of components was $M = 10$, thus the dimension of the source D was 20.

4.2 Normalized Amari-error, the Amari-index

The optimal estimation provides matrix $\mathbf{G} := \mathbf{WA}$, a block-permutation matrix made of $d \times d$ sized blocks. This block-permutation property can be measured by the Amari-index. Namely, let matrix $\mathbf{G} \in \mathbb{R}^{D\times D}$ be decomposed into $d \times d$ blocks: $\mathbf{G} = \left[\mathbf{G}^{ij}\right]_{i,j=1,\ldots,M}$. Let g^{ij} denote the sum of the absolute values of the elements of matrix $\mathbf{G}^{ij} \in \mathbb{R}^{d\times d}$. Then the normalized version of the Amari-error [2] adapted to the ISA task [24] is defined as [20]:

$$r(\mathbf{G}) := \frac{1}{2M(M-1)}\left[\sum_{i=1}^{M}\left(\frac{\sum_{j=1}^{M} g^{ij}}{\max_j g^{ij}} - 1\right) + \sum_{j=1}^{M}\left(\frac{\sum_{i=1}^{M} g^{ij}}{\max_i g^{ij}} - 1\right)\right].$$

[2] http://www.smileyworld.com

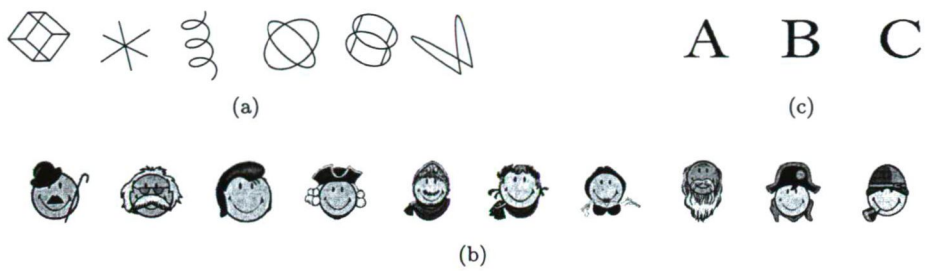


Figure 1: Illustration of the *3D-geom*, *celebrities* and *ABC* databases. (a): database *3D-geom*, 6 pieces of 3-dimensional components ($M = 6$, $d = 3$). Hidden sources are uniformly distributed variables on 3-dimensional geometric objects. (b): database *celebrities*. Density functions of the hidden sources are proportional to the pixel intensities of the 2-dimensional images ($d = 2$). Number of hidden components: $M = 10$. (c): database *ABC*. Here, the hidden sources $\mathbf{s}^m$ are uniformly distributed on images ($d = 2$) of letters. Number of components M was 10 (A-J).

We refer to the normalized Amari-error as the Amari-index. One can see that $0 \leq r(\mathbf{G}) \leq 1$ for any matrix $\mathbf{G}$, and $r(\mathbf{G}) = 0$ if and only if $\mathbf{G}$ is a block-permutation matrix with $d \times d$ sized blocks.

4.3 Simulations

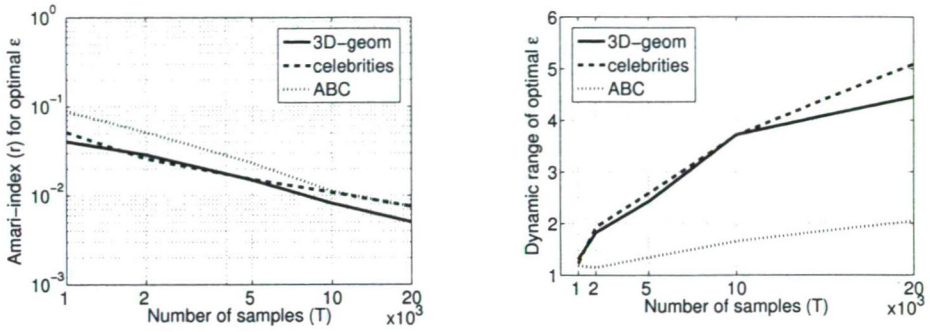
Results on databases *3D-geom*, *celebrities*, and *ABC* are provided here. Our gauge to measure the quality of the results is the Amari-index (Section 4.2) that we computed by averaging over 50 random runs.[3] These experimental studies concerned the following problems:

1. The quality of the gathering procedure depends on the threshold parameter ε. We studied the estimation error, the Amari-index, as a function of sample number. The ε values were preset to reasonably good values.

2. We studied the optimal domain for the ε values. We looked for the dynamic range, i.e., the ratio of the highest and lowest 'good ε values': We divided interval $[0, C^{\mathcal{F}}_{max}]$ ($C^{\mathcal{F}}_{max} := \max_{i,j} C^{\mathcal{F}}_{ij}$) into 200 equal parts. For different sample numbers in all databases at each division point we used the gathering procedure to group the ICA elements. For each of the 50 random trials we have computed the Amari-indices separately. For the smallest Amari-index, we determined the corresponding interval of ε's, these are the 'good ε values'. Then we took the ratio of the largest and smallest ε values in this set and averaged the ratios over the 50 runs. The average is called the dynamic range.

In our simulations, sample number T of observations $\mathbf{x}(t)$ was varied between 1,000 and 20,000. Mixing matrix $\mathbf{A}$ was generated randomly from the orthogonal

[3]Random run means random choice of quantities $\mathbf{A}$ and $\mathbf{s}$.



(a) (b)

Figure 2: Amari-index on log-log scale (a) and dynamic range (b) as a function of sample number for the *3D-geom*, *celebrities*, and *ABC* databases.

group. The fastICA [10] algorithm was chosen to perform the ICA computation. In the JFD technique, we chose manifold $\mathcal{F}$ as $\mathcal{F} := \{\mathbf{u} \mapsto \cos(\mathbf{u}), \mathbf{u} \mapsto \cos(2\mathbf{u})\}$, where the functions operated on the coordinates separately [19]. We computed correlations for matrices $\mathbf{C}_f$ ($f \in \mathcal{F}$) (instead of covariances) because it is normalized.

Our results are summarized in Fig. 2. According to Fig. 2(a), there are good ε parameters for the $\mathbf{C}^{\mathcal{F}}$-'connectedness' already for $1,000 - 2,000$ samples: our method can find the hidden components with high precision. Figure 2(a) also shows that by increasing the sample number the Amari-index decreases. For $20,000$ samples, the Amari-index is 0.5% for the *3D-geom*, 0.75% for the *celebrities*, and 0.75% for the *ABC* database, respectively on the average. The decline of the Amari-index follows power law ($r(T) \propto T^{-c}$ ($c > 0$)) manifested by straight line on log-log scale. Figure 2(b) demonstrates that for larger sample numbers threshold parameter ε that determines the $\mathbf{C}^{\mathcal{F}}$-'connected' property can be chosen from a broader domain; the dynamic range grows. For the *3D-geom*, the *celebrities* and the *ABC* databases the measured dynamic ranges are 4.45, 5.09 and 2.05 for $20,000$ samples and for the different databases, respectively on the average.

Finally, we illustrate the quality and the working of our method in Fig. 3. The figure depicts the *3D-geom* test and we used $T = 20,000$ samples. According to this figure, the algorithm was able to uncover the hidden components up to the ambiguities of the ISA task.

5 Conclusions

We have introduced a non-combinatorial solution to the estimation of the dimension of the hidden components in the ISA task. We build our method onto the ISA separation theorem and solve the ISA task in 2 steps. First, we perform ICA and then we group the ICA components. The grouping step utilizes a set of non-linear

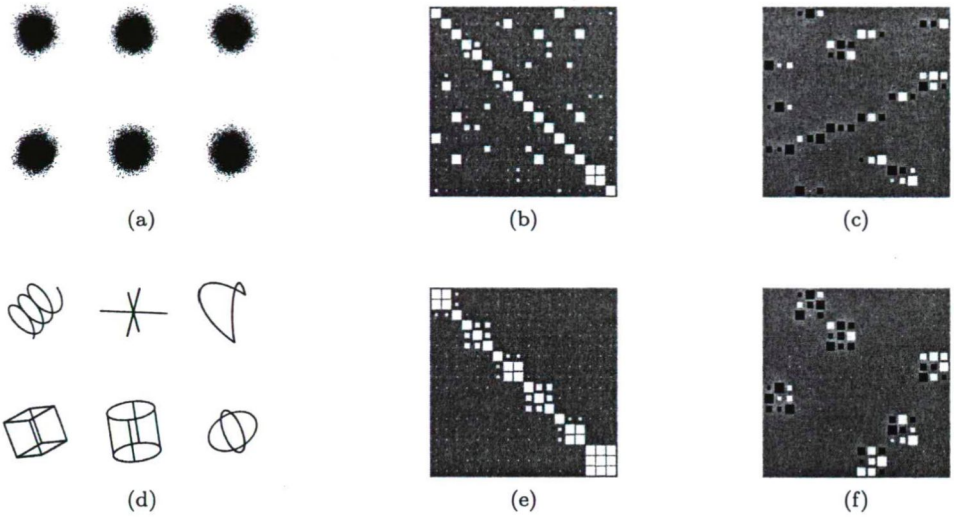


Figure 3: **Illustrations**. (a): observed mixed signal $\mathbf{x}(t)$, (b) $\mathbf{C}^{\mathcal{F}}$ - the sum of absolute values of the elements of the non-linear correlation matrices used for the grouping of the ICA coordinates, (c): the product of the ICA separation matrix and the mixing matrix, (d): estimated components $\mathbf{s}(t)$–up to ambiguities of the ISA problem–, based on (e): $\mathbf{C}^{\mathcal{F}}$ after grouping, (f) product of the estimated ISA separation matrix and the mixing matrix: with high precision, it is a block-permutation matrix made of 3×3 blocks.

correlations between the coordinates of the estimated components. Our simulations indicate that the presently known sufficient conditions of the separation theorem may be extended considerably. This remains to be shown.

References

[1] Akaho, Shotaro, Kiuchi, Yasuhiko, and Umeyama, Shinji. MICA: Multimodal independent component analysis. In *Proceedings of International Joint Conference on Neural Networks (IJCNN '99)*, pages 927–932, 1999.

[2] Amari, Shun-ichi, Cichocki, Andrzej, and Yang, Howard H. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996.

[3] Bach, Francis R. and Jordan, Michael I. Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

[4] Bell, Anthony J. and Sejnowski, Terrence J. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[5] Cardoso, Jean-François. Multidimensional independent component analysis. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, volume 4, pages 1941–1944, Seattle, WA, USA, 1998.

[6] Hyvärinen, Aapo. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, 11:1739–1768, 1999.

[7] Hyvärinen, Aapo. Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558, 2001.

[8] Hyvärinen, Aapo and Hoyer, Patrik O. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12:1705–1720, 2000.

[9] Hyvärinen, Aapo and Köster, Urs. FastISA: A fast fixed-point algorithm for independent subspace analysis. In *Proceedings of European Symposium on Artificial Neural Networks (ESANN 2006)*, Bruges, Belgium, 2006.

[10] Hyvärinen, Aapo and Oja, Erkki. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[11] Kiviluoto, Kimmo and Oja, Erkki. Independent component analysis for parallel financial time series. In *Proceedings of International Conference on Neural Information Processing (ICONIP '98)*, volume 2, pages 895–898, 1998.

[12] Makeig, Scott, Bell, Anthony J., Jung, Tzyy-Ping, and Sejnowski, Terrence J. Independent component analysis of electroencephalographic data. In *Proceedings of Neural Information Processing Systems (NIPS '96)*, volume 8, pages 145–151, 1996.

[13] Nolte, Guido, Meinecke, Frank C., Ziehe, Andreas, and Müller, Klaus-Robert. Identifying interactions in mixed and noisy complex systems. *Physical Review E*, 73(051913), 2006.

[14] Póczos, Barnabás and Lőrincz, András. Independent subspace analysis using geodesic spanning trees. In *Proceedings of International Conference on Machine Learning (ICML 2005)*, pages 673–680, Bonn, Germany, 2005.

[15] Póczos, Barnabás and Lőrincz, András. Independent subspace analysis using k-nearest neighborhood distances. *Artificial Neural Networks: Formal Models and their Applications - ICANN 2005, pt 2, Proceedings*, 3697:163–168, 2005.

[16] Póczos, Barnabás and Lőrincz, András. Non-combinatorial estimation of independent autoregressive sources. *Neurocomputing Letters*, 2006.

[17] Póczos, Barnabás, Takács, Bálint, and Lőrincz, András. Independent subspace analysis on innovations. In *European Conference on Machine Learning (ECML 2005)*, volume 3720 of *LNAI*, pages 698–706. Springer Verlag, 2005.

[18] Stögbauer, Harald, Kraskov, Alexander, Astakhov, Sergey A., and Grassberger, Peter. Least dependent component analysis based on mutual information. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 70(066123), 2004.

[19] Szabó, Zoltán and Lőrincz, András. Real and complex independent subspace analysis by generalized variance. In *ICA Research Network International Workshop (ICARN 2006)*, pages 85–88, Liverpool, U.K., 2006. http://arxiv.org/abs/math.ST/0610438.

[20] Szabó, Zoltán, Póczos, Barnabás, and Lőrincz, András. Cross-entropy optimization for independent process analysis. In *Independent Component Analysis and Blind Signal Separation (ICA 2006)*, volume 3889 of *LNCS*, pages 909–916. Springer, 2006.

[21] Szabó, Zoltán, Póczos, Barnabás, and Lőrincz, András. Separation theorem for $\mathbb{K}$-independent subspace analysis with sufficient conditions. Technical report, Eötvös Loránd University, Budapest, 2006. http://arxiv.org/abs/math.ST/0608100.

[22] Szabó, Zoltán, Póczos, Barnabás, and Lőrincz, András. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.

[23] Theis, Fabian J. Uniqueness of complex and multidimensional independent component analysis. *Signal Processing*, 84(5):951–956, 2004.

[24] Theis, Fabian J. Blind signal separation into groups of dependent signals using joint block diagonalization. In *Proceedings of International Society for Computer Aided Surgery (ISCAS 2005)*, pages 5878–5881, Kobe, Japan, 2005.

[25] Theis, Fabian J. Towards a general independent subspace analysis. In *Proceedings of Neural Information Processing Systems (NIPS 2006)*, 2006.

[26] Vigário, Ricardo, Jousmäki, Veikko, Hämäläinen, Matti, Hari, Riitta, and Oja, Erkki. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Proceedings of Neural Information Processing Systems (NIPS '97)*, volume 10, pages 229–235, 1997.

[27] Vollgraf, Roland and Obermayer, Klaus. Multi-dimensional ICA to separate correlated sources. In *Proceedings of Neural Information Processing Systems (NIPS 2001)*, volume 14, pages 993–1000, 2001.

Acta Cybernetica 18 (2007) 223–238.

Cooperative Production Networks – Multiagent Modeling and Planning

Péter Egri* and József Váncza*

Abstract

Consumer goods are mainly produced in multiple steps through a long process. These steps are often done by separate, independent production nodes (enterprises), linked by supply chains. The networks of enterprises—where members have their own objectives and act in an autonomous, rational way to reach their goals—can be naturally modeled by agent-based methodology. The inner structure of each enterprise is similar in the sense that it contains separated planning functions (e.g., production-, inventory-, capacity planning). While the operation inside an enterprise can be controlled centrally, the interaction between the nodes could be synchronized only by negotiation and coordination. Coordination can be based on protocols which regulate information, material and financial flows alike. In this paper we expose an agent-based organizational model of production networks and suggest some planning algorithms which can handle the uncertainty of demand. In addition, we outline the first results of our ongoing research, an analysis of the asymmetric information case and an appropriate coordination mechanism.

Keywords: multiagent modeling, supply chain, inventory planning

1 Introduction

Nowadays, customers of consumer goods are more demanding than ever and manufacturing must fulfill their needs to remain competitive. Naturally, there exist several manufacturing paradigms to answer the existing challenges all with their own advantages and disadvantages [16]. The *craft production*—whose golden age was before the 20th century—allows large variety of products, but requires complicated, time-consuming processes, which are also expensive. *Mass production*—the main paradigm in the 20th century—achieves higher efficiency with standardized products, exploiting economies of scale and (semi-)automated processes, but gives up the wide product scale. In the last few decades the new paradigm of *mass customization* has arisen, which tries to combine the advantages of the previous

*Computer and Automation Research Institute, Hungarian Academy of Sciences E-mail: {egri,vancza}@sztaki.hu

two approaches by offering a larger variety of products made of standardized components with mass production technology and *delayed differentiation.* As it has turned out, while this new paradigm solves some of the problems, it also poses new questions.

Satisfying demand directly from production is impossible, because production and supply lead-times are much larger than the acceptable delivery times for the customers, and, in addition, manufacturers should also exploit the economies of scale. Therefore holding inventories is necessary, which can only be based on fluctuating, uncertain forecasts. Unfortunately, the use of some non-standardizable components (e.g., customized packaging materials) is necessary, which leads to obsolete inventory when the demand for the customized product suddenly ceases. Hence, this phenomenon called *run-out* causes both significant financial losses for the manufacturers and waste of environmental resources.

Consumer goods are mainly produced in a long process of multiple steps, which are often carried out by separate, independent and rational production enterprises, linked by supply chains. This decentralization leads to suboptimal overall system performance called *double marginalization* [19].

The goal of our research is to improve the efficiency of production networks as a whole. In order to do this, we have first developed a multiagent model of the networks and the various planning functions within the enterprises. These functions cover all kinds of decision making that influence and control the future, thus it is substantial to examine and describe them in detail. Then we have analyzed industrial databases and found that component inventory levels have often been inappropriate—either too high or low. These levels are set by the supply processes that connect planning functions of different enterprises. Hence, we have introduced some new models and optimization algorithms which align to the new market conditions. While these models assume a centralized decision maker who possesses all relevant information, in a real network this is not the case. The solutions of these algorithms provide only lower bounds on the total cost in case of a decentralized network with asymmetric information. Hence, our aim is to design and develop such *channel coordination mechanisms* that achieve or approximate the outcome of theoretically optimal decisions even if the partners decide locally, by relying on incomplete (asymmetric) and uncertain information. In the end, we are going to turn our multiagent model into a simulation which will include the implemented planning algorithms in order to verify them on real industrial data.

The goal of this paper is, on one hand, to expose the results of these research steps. On the other hand, we present recent results of our related work in a unified framework that is based on the agent-based design metaphor. As we will show in the sequel, taking the agent-based approach helps a lot in clarifying complex and often blurred organizational relations and constructing an appropriate organizational model. Though the transformation of this model into an agent-based simulation model is far from being straightforward, still it is the best way towards validating and verifying the outcomes of our research.

The motivation of this work comes from a large-scale national industrial-academical R&D project aimed at realizing *real-time, cooperative enterprises.* The

participating industrial partners form a complete *focal* network: a central assembly plant with several external and internal suppliers. The assembler produces altogether several million units per week from a mix of thousands of products. The ratio of the customization follows the *80/20 Pareto-principle*: they give 80% of the product spectrum, but only 20% of the volume. The setup costs are significant and since customized products are consumed slower, their smaller lot sizes involve higher average setup costs. Service level requirements are extremely high: some retailers suddenly demand products in large quantities even within 24 hours, and if the request is not fulfilled on time, they cancel the order (i.e., backorders are not possible). This causes not only lost sales, but also decrease of goodwill.

2 Related work

Business process and supply chain modeling research has produced several methodologies in the last decades (e.g., CIMOSA, IDEF0, EPC, SCOR) [22]. These approaches, however, provide tools only for modeling and analyzing the processes and do not support decision making. A uniform model of inter-enterprise planning functions and their hierarchical layout in a matrix is described in [17]. The importance of the role-based modeling approach of collaborative networks is emphasized in [25]. Recently, several efforts have been made towards integrating modeling and optimization, e.g., an object-oriented approach is presented in [1].

From the viewpoint of production networks, there are basically two types of research utilizing agent-based concepts and methods: (i) the general approach handles supply chain management as a problem of designing and operating a multiagent system and (ii) the other kind focuses on specific problems, such as collaborative inventory management, bidding decision, material handling and inventory planning in warehouses. The majority of the literature has been focusing on the general application of agent-based supply chain management. The rich variety of multiagent approaches clearly shows the application potential of agent technology. By now, there is a common understanding that various requirements of networked manufacturing can really be met by autonomous, embodied, communicative and eventually cooperative agents operating in a society.

Still, according to our recent survey, the number of deployed multi-agent systems that are already running in real industrial environments is surprisingly small: even in the "ideal" field of supply chain management, only half a dozen applications can be found that are deployed in everyday use [13]. Other reviews also concluded with the observation that no significant advancement had been made yet in transferring agent technology to industry [12, 15]. This, relatively slow transfer has manifold reasons. First, the introduction of agents, in principle, does not reduce the complexity of problems. Next, interoperability is expensive. Just due to the increased communication overhead, the performance of a multiagent system can degrade and especially rough-grained systems (consisting of sophisticated agents) can hardly be scaled up. Although the agent metaphor is useful in system design, and there are also several methodologies to support agent-oriented software

engineering, industrial-strength support is still missing. Finally, in the behavior of a multiagent system there is always an element of emergence which can be a serious barrier to the practical acceptance of agent-based solutions. Industry needs safeguards against unpredictable behavior and guarantee regarding reliability and operational performance.

In order to bridge this gap between theory and practice, we put emphasis on the elaboration of network coordination models that have analytically provable properties and, at the same time, efficient solution techniques. In a production network, it is *logistics* that essentially links the various partners. Hence, our coordination models have been derived from models of logistics and inventory planning.

The history of *inventory planning* is almost hundred years old and the most important models were born in the 1950s. Nevertheless, because of the changing market conditions, the research in this area is still ongoing. The related idea of coordination mechanisms has also attained interest in the recent years [10]. This research can be classified along two dimensions: (i) the nature of the demand is either deterministic or stochastic, and (ii) the decision structure is either centralized or decentralized. From the four possible combinations the decentralized stochastic one is the most complex, hence the literature of this case is scattered. Existing research approaches are mostly based on the results of game theory and economy with asymmetric information. The risk of obsolete inventory and its placement is studied considering different types of contracts in [2].

3 Multiagent organizational model

We have decided to model a production network and the planning functions within an enterprise as a multiagent system and this decision needs explanation, because there already exist widely used methodologies for this purpose [22]. While these modeling tools can describe the high-level structure and processes of enterprises, they cannot be further detailed and do not directly support neither the elaboration of planning algorithms, nor the estimation of their computational complexity, nor the software implementation. On the contrary, agent-oriented methodology offers (i) a design metaphor for complex systems, (ii) technologies for handling interactions and (iii) simulation tools alike [11].

For studying the situation, we have used the Gaia methodology, which is a specification framework for analyzing and designing the organizational model of multiagent systems [24]. This methodology helps to identify and separate different roles in the planning structure of an enterprise, which are in real life sometimes mixed and overlapping.

Gaia deals with two aspects of the modeled system: the *abstract* viewpoint helps to conceptualize and analyze the organization, while the *concrete* viewpoint is used during the design phase to model entities which will be realized in the run-time system. So far we have taken only the high-level analytic approach, but as a future direction, we will continue detailing the model and use it as a basis for multiagent simulation.

The analytic part of Gaia consists of two models: *roles* and *interactions*. Since an organization is considered a collection of roles, the main challenge is to distinguish different roles, describe them and define their interactions. A role can have a set of *permissions*, which are rights associated with the role—typically these are read/create/modify permissions to certain *shared information resources*. In addition, a role has some *responsibilities*: there are *liveness* properties which declare what the role must do and *safety* properties which are invariants stating situations to be avoided. Liveness properties, which resemble regular expressions, consist of *activities*, which are autonomous computations, and *protocols*, which are interactions between roles. We have extended the roles model with the description of optimization objectives what we have found essential in planning functions.

The evolution of planning functions in production management, and recently in supply chain management, resulted in a planning hierarchy [17] that we adopt to our modeling purpose. This so-called *planning matrix* shows long-term, medium-term and short-term planning functions organized along the main flow of materials. These functions are common at each node of a production network, though, of course, manifest themselves in different forms and complexity. We have described all functions of the matrix as Gaia roles [4]; an example can be seen in Figure 1.

Role Schema: SUPPLYPLANNER
Description: This role ensures the necessary raw materials for the manufacturer by creating medium-term material requirement plan, ordering and maintaining the raw product inventory.
Protocols and Activities: Order, CustomerForecast, Call-off, Transport, Exception, CreateMaterialRequirementPlan
Permissions: **reads** *forecasts* *plannedOrders* *schedule* *technologicalData* **changes** *rawProductInventoryLevels*
Responsibilities Liveness: SUPPLYPLANNER = (CreateMaterialRequirementPlan . [Order] . [Call-off] . ([CustomerForecast])^p || (Exception | Transport)$^\omega$ Safety: – execution of the schedule must not stop because of material shortage Objectives: – minimal raw product inventory level – minimal obsolete raw product inventory

Figure 1: An example role description in Gaia.

The protocols of the roles can be further detailed in the interaction model of Gaia. While most interactions inside an enterprise are realized via Enterprise Resource Planning (ERP) systems, the inter-enterprise interactions should be precisely regulated according to some protocols. Examples of such interactions—namely forecast sharing and ordering—can be seen in Figure 2, which contains two instances of the planning matrix. In the following, we concentrate on the planning functions directly related to these interactions.

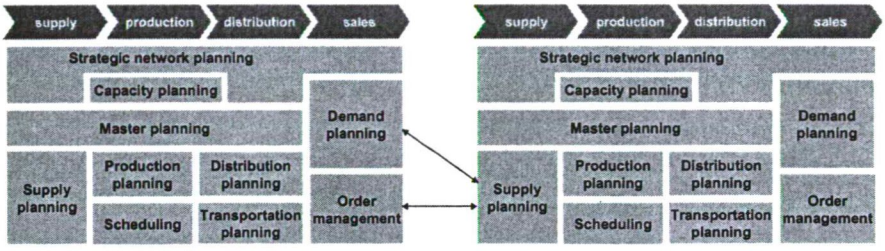


Figure 2: Interactions between enterprises.

4 Uncertainty and planning

While the golden age of inventory research was in the 1950s, the recently changed market conditions have induced paradigm change and the need for new models [3]. In order to remain competitive on global markets, today's production must be *customer-oriented*, which means that customer demand must be satisfied at high service level with short lead-times. These main requirements—which are specified by the long-term *strategic management*—must be achieved on lower levels by the *tactical* and *operational management*, which need new models and tools for optimization.

The strategic decisions are out of the scope of this research; we take their results given. I.e., we depart from an existing network structure and specified high-level goals. The tactical and operational decisions, in turn, can be detached: the medium-term (planning) level is responsible for the cost-efficient production by aggregating production into batches, while the short-term (scheduling) level—where the precise demand is known—cares for the service level requirements. Our models presented below are dealing with planning decisions, but we have also developed a framework for coordinating the two levels [5, 6].

As it was previously mentioned, the production is based on uncertain finished good forecast, which can be prepared using several statistical methods [8] and therefore the uncertainty can be expressed in terms of standard deviation. Unfortunately, this information is usually distorted by human factors [7]. In addition, when the production is planned in medium term, the uncertainty information disappears or is transformed to *safety stock* margins, because most practical planning systems cannot handle stochastic problems. The result of the planning process is

a discrete plan of production quantities, respecting the capacity and technological constraints. This is also regarded as the basis of the "component usage forecast" and the economic purchase plan can be determined from this component forecast using appropriate lot-sizing methods. This metamorphosis of demand-related information is illustrated in Figure 3.

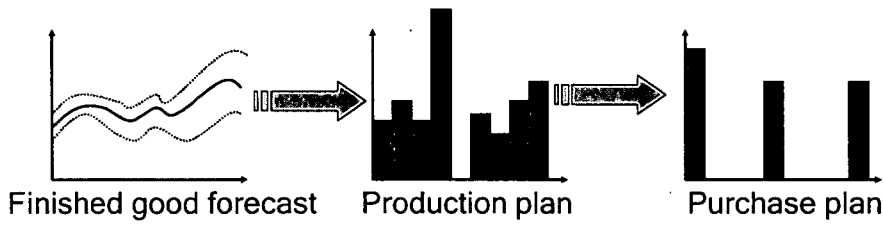


Figure 3: The distortion of demand-related information.

The general inventory planning problem can be briefly characterized in the following way: there exists a medium-term *planning horizon* with an uncertain demand. We regard the component demand derived from the production plan, since supply must be aligned to production instead of the finished good sales. Components should be produced in large batches in order to decrease the setup cost, but this comes together with an increase of inventory levels and of expected obsolete inventory costs. While searching for the optimal trade-off, the constraint of avoiding shortage must be respected.

4.1 Modeling uncertainty

Typically, the operational managers have no models and decision support tools for handling the inventory risk, therefore they usually use ad-hoc rules-of-thumb based on historical statistics—which is sometimes referred to as "driving by the rear-view mirror".

The stochastic inventory models developed using theoretical approaches are rarely used in the practice because of their complexity and lack of data. Instead, deterministic demand is assumed and reconsidered from time to time, which is called *rolling-horizon planning* [21]. We have extended this practical approach by introducing a single parameter, the weekly *probability of run-out* (p), which reflects the stability of the products and components. It can be assigned either to components or groups, based on market information, as well as on historical data.

Nevertheless, this information about probability cannot be explicitly found in existing enterprise data warehouses and determining it can be costly and time-consuming. Therefore we have proposed to use our planning methods with different run-out parameters in parallel. From these *scenarios* it can be estimated how *robust* is the solution, i.e., how much it depends on the changes of the run-out parameter. If the robustness is low, then the automatically generated plan must be reviewed by human experts or a more precise parameter value is required.

There are two fundamentally different situations according to the information available: (i) the fact of the run-out and its date are known in advance and (ii) run-out can occur with a certain probability, but no further details are known. In the first case, the standard Wagner–Whitin (WW) method [23] can be used, which plans the whole horizon and tells in which periods to produce and how much. Nevertheless, this approach can lead to an inappropriate solution when the horizon is too short: it suggests producing in one lot and disregards the possibility of a larger demand, which causes inefficient additional production. Therefore, if WW proposes producing in one lot, one should switch to a more appropriate method: the so-called *newsvendor model* for one-period. But this first case is exceptional, usually run-out is not known in advance. For this case, we have developed two heuristics and a modified version of WW what we present below.

4.2 One-period model

The inventory systems of perishable goods are usually modeled as *one-period decision problems*: the decision maker has to determine the value of a variable q, then a cost of $c(q,\xi)$ arises, where ξ is a random variable with known distribution. The *risk neutral* decision maker wishes to minimize the expected cost. In the context of inventories, this model is called the newsvendor model, since it describes well the inventory management problems of the daily newspaper markets [8].

In such a case, overplanning leads to obsolete inventory, while underplanning may lead to costly additional setups. The standard model disregards the setup cost. It only considers per unit left over cost—if the demand is below the produced quantity—and per unit shortage cost (it may be interpreted as producing in overtime)—if the demand is above. However, if the inventory is filled by manufacturing instead of ordering, then the setup cost must be included in the calculation.

In our model [6], service level has the highest priority, hence it follows that the manufacturer has to satisfy all demand. If the produced quantity is below the actual demand, it can only be satisfied by an *emergency production* which also involves an additional setup. Thus, our model involves four types of costs: (i) the certain setup cost (c_s), (ii) the production cost for satisfying actual demand (c_p per item), (iii) the expected value of obsolete left over products (with c_p per unit left over cost) and (iv) the expected cost of additional setup. Then the expected total cost becomes:

$$\mathbb{E}[TC(q)] = c_s + c_p\mathbb{E}[\xi] + c_p\mathbb{E}[\max(q-\xi,0)] + c_s\mathbb{E}[\delta(\xi-q)], \tag{1}$$

where q is the produced quantity, ξ is the random demand and

$$\delta(\xi-q) = \begin{cases} 0 & \text{if } \xi - q \leq 0 \\ 1 & \text{if } \xi - q > 0 \end{cases} \tag{2}$$

In order to minimize the expected total cost, we have to compute the derivative of the cost function, which ought to be zero:

$$\frac{\mathrm{d}\mathbb{E}[TC(q)]}{\mathrm{d}q} = c_p\Phi(q) - c_s\phi(q), \tag{3}$$

where ϕ and Φ are the density and distribution functions of the demand respectively. Here, usually normal distribution is considered, despite of its disadvantages [18].

Since this expression cannot be inverted in the general case as in the standard model, we apply the so-called *logistic distribution* with parameters b and m, which is often used instead of the normal distribution when longer tail is more appropriate. This yields the unique stationary point—which is a minimum—if $b < \frac{c_s}{c_p}$:

$$q^* = m - b \ln\left(\frac{bc_p}{c_s - bc_p}\right) \tag{4}$$

This optimal lot size gives a balance between the risk of obsolete inventory and the additional setup. It can be both more or less than the expectation value, depending on the variance and the cost parameters (see also Section 6).

4.3 Multi-period models

The more remote a forecast is, the more uncertain it is—this reasonable hypothesis was confirmed by our analysis of historical industrial data. Based on this observation, our first idea was to disregard the less trusted remote forecasts and plan only the starting segment of the horizon. Therefore we have developed two heuristic methods, which minimize the expected average cost—both per time unit and per piece—in the first segment of the horizon [20]. As it has turned out, the heuristics have several disadvantages: (i) they cannot estimate the number of setups on the horizon, (ii) disregarding a part of the available information can lead to significant inefficiency and (iii) they sometimes behave unreasonably: increasing the probability of run-out can cause higher lot size, see Figure 4.

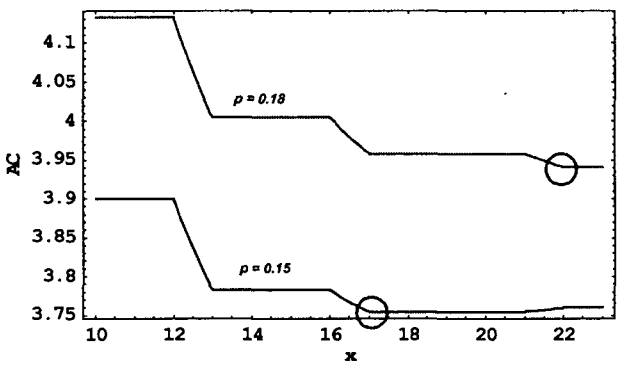


Figure 4: Anomaly of heuristics.

The lines show average costs in case of different run-out probabilities. If $p = 0.15$, producing the demand of 17 time units would minimize the average cost, while the higher p suggests 22 time units. According to our experiments, such anomalies occur rarely, and when run-out probability is relatively high ($p > 0.13$).

Note that such anomalies are known in the field of operations management; see e.g., *nervousness* in the widely used Material Requirements Planning (MRP) method, when a decrease in the demand leads to an infeasible situation [8].

Hence, our conclusion was that planning the whole horizon is necessary, therefore we have decided to use the Wagner–Whitin model ([23]) extended with the probability of run-out, which we abbreviate as WWr [5]. The main elements of this model are as follows: length of the horizon (n), forecasted demand ($F_1, \ldots, F_n$), setup cost (c_s), inventory holding cost per piece per time unit (h), production cost per piece (c_p) and the probability of run-out in an arbitrary time unit (p). The decision variables are the production quantities in the time units throughout the whole horizon ($x_0, \ldots, x_{n-1}$). We assume infinite capacity and introduce a one-period lead-time. In this setting, the Wagner–Whitin property remains valid: it is optimal not to produce, unless the inventory would become empty otherwise. The planned lot sizes can be determined by a dynamic programming algorithm briefly summarized below.

If we produce in time unit t for the period $\{t+1, \ldots, t+j\}$, this implies that (i) the expected inventory at the beginning of time unit $t+1$ is zero (Wagner–Whitin property) and (ii) the product has not run out until the beginning of time unit t (which has a probability $(1-p)^t$). Then the expected storage cost at time unit $t+i$ is

$$SC(t,j,i) = (1-p)^i h \left(\sum_{k=i+1}^{j} F_{t+k} + \frac{F_{t+i}}{2} \right) \tag{5}$$

and the cost of expected obsolete inventory is

$$OC(t,j,i) = p(1-p)^{i-1} c_p \sum_{k=i}^{j} F_{t+k}, \tag{6}$$

which expresses that with probability $(1-p)^i$ the product is still saleable, therefore storage cost must be paid, and with probability $p(1-p)^{i-1}$ it runs out in the very time unit and the remaining inventory becomes obsolete. The expected total cost of period $\{t+1, \ldots, t+j\}$ is therefore

$$C_{tj} = c_s + \sum_{i=1}^{j} \left(SC(t,j,i) + OC(t,j,i) \right). \tag{7}$$

The optimal total cost for period $\{t, \ldots, n\}$ (TC_t) can be computed by the following recursion:

$$TC_n = 0 \tag{8}$$

$$TC_t = \min_{j \in \{1, \ldots n-t\}} \left\{ C_{tj} + (1-p)^j TC_{t+j} \right\}. \tag{9}$$

Note that $(1-p)^j$ is the probability of the event that the product has not run out and further production is necessary.

With a backward induction, the optimal lot sizes and the expected number of setups can be also obtained from the optimal j values in the recursion. This provides an $\mathcal{O}(n^2)$ algorithm, which is practically acceptable.

5 Asymmetric information

In a real network, no central planner exists with all required information, as it was assumed in Section 4.1. In a two-echelon supply chain system, the supplier is familiar with the production and setup cost for the components, while the end manufacturer can have a good estimate of the finished good demand. In this case, the so-called *first-best solutions* of the presented algorithms provide only lower bounds for the achievable expected total cost.

Our goal is to design a coordination mechanism which helps the partners to reach (or approximate) the results of the first-best solutions. At first, we have concentrated on the one-period newsvendor problem. Extending this research to the multi-period case is part of future work.

In the decentralized newsvendor setting, the production cost (c_p), the setup cost (c_s) are the parameters known by the supplier, while the end manufacturer knows only the demand-related information (m and b). We assume that the lot size decision is made by the supplier—who can schedule its own production—and it also holds the inventory. For being able to do this, the end manufacturer *signals* the demand information towards the supplier. This information can be distorted—e.g., the mean can be inflated in order to decrease the risk of shortage—therefore we denote these parameters with m' and b'. Note that if there is no distortion (i.e., $m' = m$ and $b' = b$) and this is a common knowledge, supplier is facing the problem presented in Section 4.2 with all required information, therefore its rational lot sizing decision is also optimal on the system level.

This is a conflict of interests: while the optimal network performance requires truthful information sharing, the manufacturer can be better off by distorting the information. This conflict can be resolved by an appropriate payment function which aligns the objective of the manufacturer with that of the supply chain: it guarantees that the expected payment will be minimal, if the end manufacturer signals the truth and does not distort forecast information.

In this situation, the supplier takes all inventory risks, therefore the end manufacturer has to pay for the *service* of flexibility besides paying for the components. Hence, payment consists of (i) the price of the delivered components, (ii) compensation for the deviation from the forecast and (iii) compensation for the forecast uncertainty. Therefore the payment function becomes:

$$P(m', b', \xi) = c_0\xi + \frac{c_1}{b'} d(m', \xi) + c_2(b'), \tag{10}$$

where c_0 and c_1 are constants: the unit prices for required components and inappropriate demand estimation, respectively. The term $d(m', \xi)$ is the difference between the estimated and the realized demand and $c_2(b')$ is the compensation for uncertainty. Note that the payment depends only on commonly known parameters.

A possible choice for measuring the deviation from the forecast is the quadratic difference function: $d(m', \xi) = (m' - \xi)^2$. For this case we have proven that, if the uncertainty compensation is $c_2(b') = c_1 \frac{\pi^2}{3} b'$, the payment function inspires the rational manufacturer to signal the real m and b values to minimize the payment. Since the income depends only on the demand, minimal payment also maximizes the profit of the manufacturer.

6 Experiments

All presented algorithms were implemented—together with the *safety stock* calculation—and tested both on generated and real industrial data in several scenarios. We have studied how the uncertainty influences the optimal lot sizes and the expected costs. We have also simulated the inventory levels according to our models, based on industrial information of past demand. Some of these results are presented below.

An example result of the newsvendor model can be found in Figure 5: the proposed lot sizes vary with the relative deviation of the estimated demand and are compared with the forecast. Note that the solution can be both more or less than the forecast, depending on the uncertainty and cost parameters. Nevertheless, the curve is smooth, i.e., proposed quantities do not fluctuate with high frequency. The intuitive explanation of the shape of the curve is as follows: if there was no uncertainty, optimal lot size would equal to the demand. When the uncertainty increases, it is better to increase the lot size in order to avoid the additional setup. However, when the uncertainty reaches a certain threshold, the expected cost of obsolete inventory equals the expected cost of the additional setup, therefore the optimal lot size starts to decrease.

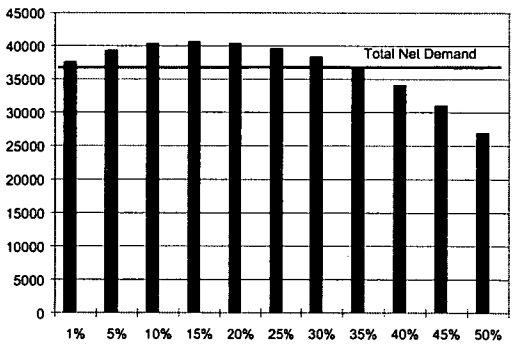


Figure 5: Results of the newsvendor model.

In Figure 6, the results of the two heuristics (AC_x minimizes the average cost for a time unit, while AC_q minimizes the average cost of an item) and the WWr are shown considering different run-out parameters. As mentioned before, WWr can

estimate the number of setups on the horizon, therefore these are also indicated. The results were also compared with the lot sizing decision made by human experts. The conclusion of several months of weekly consultations with industrial partners was that, in around 90% of the cases, WWr with probability parameter $p = 0.02$ proposed automatically almost the same lot size as the human planners—apart from the rounding. In the remaining cases the planners had additional information—received via phone or e-mail—which was not stored in the data warehouse, therefore it was not visible to us.

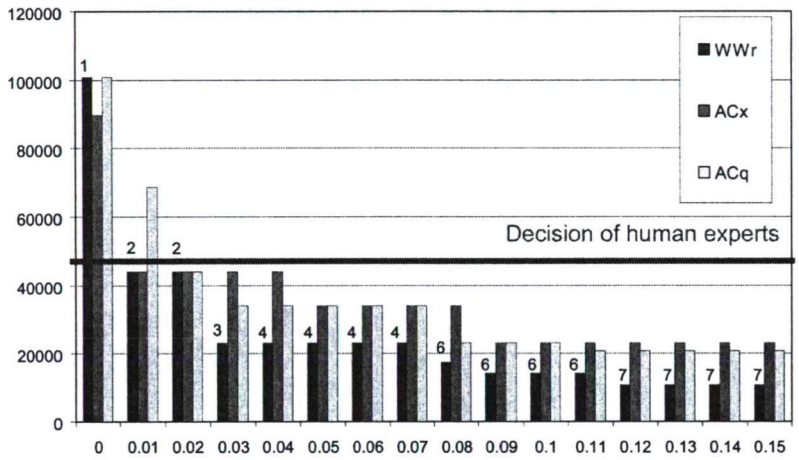


Figure 6: Results of the multi-period models.

The algorithms were also included into a pilot decision support application, which is now under testing in industrial environment.

Finally, we have started to develop a multiagent inventory simulation system in Repast [14]. The structure of the system follows the model presented in Section 3, while the internal decision making of the agents is based on the algorithms of Section 4. The stochastic data can be obtained from two sources:

- for random number generation we use the *Colt* package included in Repast, which was developed for high performance scientific and technical computing by CERN, and

- we can query real forecast and demand data from industrial databases via direct database access (JDBC).

The simulation runs are evaluated using several common indices, such as total cost, number of setups, average inventory level, service level, etc. An example interface of the system can be seen in Figure 7.

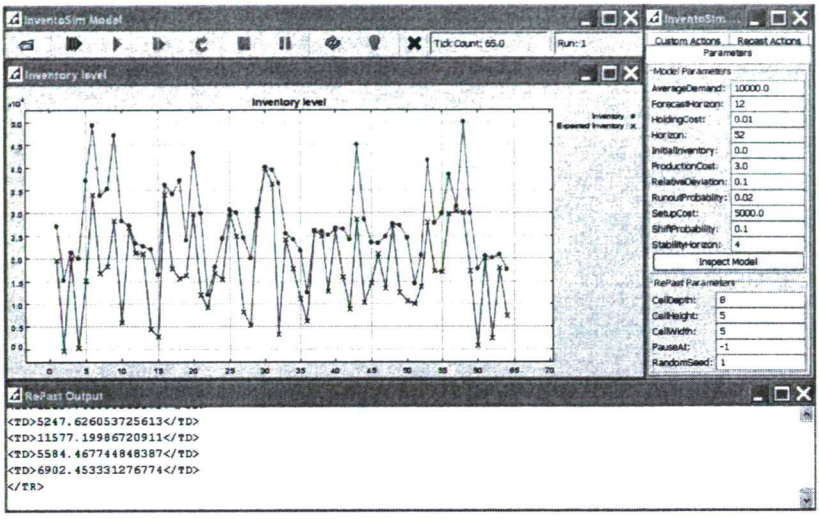


Figure 7: Inventory simulation system in Repast.

7 Future work

The presented multi-period inventory planning models consider infinite capacity and therefore they can be solved efficiently. However, in the real world, capacities are often limited, and in addition, setup costs are not independent of the production sequences. This makes the problem much more difficult: it is proven to be NP-hard. The exact solution of such problems—even with efficient specialized algorithms—is achievable only on relatively small instances [9]. Therefore numerous approximation algorithms and heuristics are applied, which provide quasi-optimal solutions for some special cases. One possible future work is to combine our model with these solution concepts.

So far, we have considered only the one-period case of the asymmetric information. Naturally, we will continue the research in the case of longer horizons. We would also like to improve our agent-based simulation system so that it supports more complex analysis of the problems.

Acknowledgments

This work has been supported by the VITAL NKFP grant No. 2/010/2004, the OTKA grant No. T046509 and the EU AC/DC IST 031520 project.

References

[1] Biswas, S. and Narahari, Y. Object Oriented Modeling and Decision Support for Supply Chains. *Europen Journal of Operational Research*, 153:704–726, 2004.

[2] Cachon, G. P. The Allocation of Inventory Risk in a Supply Chain: Push, Pull, and Advance-Purchase Discount Contracts. *Management Science*, 50(2):222–238, 2004.

[3] Chikán, A. The New Role of Inventories in Business: Real World Changes and Research Consequences. *International Journal of Production Economics*, in print, 2007.

[4] Egri P. and Váncza, J. Cooperative Planning in the Supply Network – A Multiagent Organization Model. In Pechoucek, M., Petta, P. and Varga, L. Zs. editors *Multi-Agent Systems and Applications IV*, Springer LNAI 3690, pages 346–356, 2005.

[5] Egri P. and Váncza J. A Logistics Framework for Coordinating Supply Chains on Unstable Markets. *Proc. of* 3^{rd} *Conference on Digital Enterprise Technology*, 2006.

[6] Egri P. and Váncza J. Incentives for Cooperative Planning in Focal Supply Networks. *Proc. of* 6^{th} *International Workshop on Emergent Synthesis*, pages 17–24, 2006.

[7] Goodwin, P. Providing Support for Decisions Based on Time Series Information Under Conditions of Asymmetric Loss. *European Journal of Operational Research*, 163:388–402, 2005.

[8] Hopp, W. J. and Spearman, M. L. *Factory Physics – Foundations of Manufacturing Management.* Irwin, 1996.

[9] Kovács A., Brown, K. N. and Tarim, A. Generating Efficient Models of Capacitated Lot-sizing and Scheduling Problems with Sequence-dependent Setups. *International Journal of Production Economics*, accepted, 2006.

[10] Li, X. and Wang, Q. Coordination Mechanisms of Supply Chain Systems. *European Journal of Operational Research*, 179(1):1–16, 2007.

[11] Luck, M., McBurney, P., Shehory, O. and Willmott, S. Agent Technology Roadmap – A Roadmap for Agent Based Computing. `http://www.agentlink.org`, 2005.

[12] Marik, V. and McFarlane, D. Industrial Adoption of Agent-Based Technologies. *IEEE Intelligent Systems*, 20(1):27-35, 2005.

[13] Monostori L., Váncza J. and Kumara, S. R. T. Agent-Based Systems for Manufacturing. *Annals of the CIRP*, 55(2):697-720, 2006.

[14] Repast – Recursive Porus Agent Simulation Toolkit. `http://repast.sourceforge.net/`

[15] Shen, W., Hao, Q., Yoon, H. J. and Norrie, D. H. Applications of Agent-Based Systems in Intelligent Manufacturing: An Updated Review. *Advanced Engineering Informatics*, 20(4):415-431, 2006.

[16] Simchi-Levi, D., Kaminsky, Ph. and Simchi-Levi, E. *Designing and Managing the Supply Chain: Concepts, Strategies, and Cases.* McGraw-Hill, 2000.

[17] Stadtler, H. Supply Chain Management and Advanced Planning – Basics, Overview and Challeges. *European Journal of Operational Research*, 163:575–588, 2005.

[18] Strijbosch, L. W. G. and Moors, J. J. A. Modified Normal Demand Distributions in (R, S)-inventory Control. *European Journal of Operational Research*, 172:201–210, 2006.

[19] Tirole, J. *The Theory of Industrial Organization.* MIT Press, 1988.

[20] Váncza, J. and Egri, P. Coordinating Supply Networks in Customized Mass Production – A Contract-based Approach. *Annals of the CIRP*, 55(1):489–492, 2006.

[21] van den Heuvel, W. and Wagelmans, A. P. M. A Comparison of Methods for Lot-sizing In a Rolling Horizon Environment. *Operations Research Letters*, 33:486–496, 2005.

[22] Vernadat, F. B. *Enterprise Modelling And Integration: Principles and Applications.* Chapman & Hall, 1996.

[23] Wagner, H. M. and Whitin, T. M. Dynamic Version of the Economic Lot Size Model. *Management Science*, 5:89–96, 1958.

[24] Wooldridge, M. Jennings, N.R. and Kinny, D. The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, 2000.

[25] Zaidat, A., Vincent, L. and Boucher, X. Role Based and Processes Based Modelling Approach for Organization Network. *Proc. of Virtual Enterprises and Collaborative Networks*, pages 35–42, 2004.

Acta Cybernetica 18 (2007) 239–255.

Synthesising Robust Schedules for Minimum Disruption Repair Using Linear Programming

Dávid Hanák*, Nagarajan Kandasamy†

Abstract

An off-line scheduling algorithm considers resource, precedence, and synchronisation requirements of a task graph, and generates a schedule guaranteeing its timing requirements. This schedule must, however, be executed in a dynamic and unpredictable operating environment where resources may fail and tasks may execute longer than expected. To accommodate such execution uncertainties, this paper addresses the synthesis of robust task schedules using a slack-based approach and proposes a solution using integer linear programming (ILP). Earlier we formulated a time slot based ILP model whose solutions maximise the temporal flexibility of the overall task schedule. In this paper, we propose an improved, interval based model, compare it to the former, and evaluate both on a set of random scenarios using two public domain ILP solvers and a proprietary SAT/ILP mixed solver.

Keywords: scheduling, integer linear programming, robustness, slack, resource failure

1 Introduction

Scheduling plays a crucial role in manufacturing and service industries where companies must sequence their activities (or tasks) appropriately to meet customer deadlines. An off-line scheduling strategy considers resource, precedence, and synchronisation requirements of tasks, and generates a static schedule satisfying task timing constraints [2]. This schedule executes in a dynamic and unpredictable operating environment where critical resources may fail, tasks may execute longer than expected, or certain new tasks may need urgent processing. Consequently, the task schedule must accommodate such execution uncertainties.

*Department of Computer Science and Information Theory, Budapest University of Technology and Economics (BME SZIT), Hungary, E-mail: dhanak@cs.bme.hu

*Computer and Automation Research Institute of the Hungarian Academy of Sciences (MTA SZTAKI), E-mail: dhanak@sztaki.hu

†Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, U.S.A., E-mail: kandasamy@cbis.ece.drexel.edu

In this paper we address the synthesis of robust task schedules using a slack-based approach. In [11] we developed a method to construct schedules where individual tasks retain some temporal flexibility in the form of slack while satisfying their timing requirements. As opposed to *reactive* methods [7,16,19], which recover from the disruption as it happens, our method was *proactive,* i.e. it constructed a schedule that could absorb some disruptions without the need for rescheduling. In previously proposed proactive methods, like [4] and [9], a given amount of slack had been added to the tasks to accommodate expected repair times prior to scheduling. This had resulted in an increased make-span of the entire schedule. On the other hand, our method assumed that the tasks had explicit deadline and resource requirements. The goal was then to maximise the slack of each task such that the resulting schedule satisfied all the temporal constraints and the flexibility was maximal according to a given cost function. We proposed an Integer Linear Programming (ILP) model of the scheduling problem, and evaluated it using two different ILP solvers. In this paper, we advance our method by introducing an improved ILP model that performs better on larger problems. We also get a third kind of ILP solver involved in the evaluation of the new model, and compare the evaluation results with those of the first model.

We begin the discussion in Sect. 2 with a brief introduction of the background and preliminary assumptions. Section 3 summarises the first, slot based ILP model formulated in [11] and introduces the second, interval based model. We also show how the new model can incorporate additional forms of temporal constraints. Section 4 evaluates the performance of both models and compares the evaluation metrics. We conclude the paper with a discussion of future work in Sect. 5.

2 Preliminaries

This section introduces the application domain, then goes on to discuss the task model, sources of slack in a task schedule, and the slack distribution algorithm.

2.1 Application Domain

A research group of the *Institute for Software Integrated Systems (ISIS)* at Vanderbilt University in Nashville had been participating in the *Autonomous Negotiating Teams (ANT)* project [1,17] of the Defense Advanced Research Projects Agency (DARPA) for several years, in cooperation with the *Information Sciences Institute (ISI)* at the University of Southern California in Los Angeles. The deliverable result of the research was the prototype of a software tool to aid the scheduling of flight missions and regular maintenance of Harrier aircraft in a U.S. Marine Corps squadron.

A considerable part of the effort was the development of the core scheduling engine for maintenance tasks, which has been solved by implementing a finite domain constraint scheduler in Mozart-Oz [18]. This approach relies on a group of *propagators* which act independently on a shared set of variables with finite in-

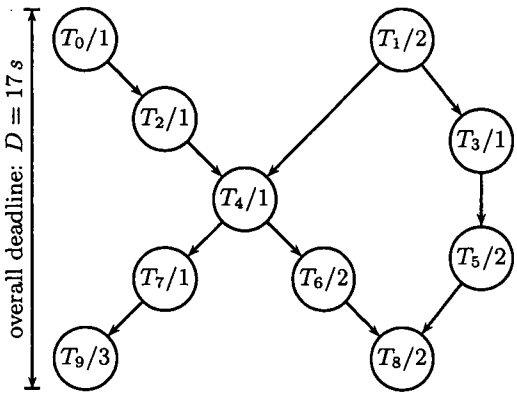


Figure 1: An example task graph G with an overall deadline of 17 seconds

teger domains [20], which, however, due to the independence of the propagators, can sometimes lead to inconveniently deep but solutionless subtrees in the search tree of the entire problem. This in turn can produce an unpredictable behaviour, where the scheduling engine produces a solution in a matter of seconds for one problem, but enters into a unacceptably long and fruitless search for another. This problem is a current issue in the field of constraint programming and it has been addressed by various research efforts [3, 15]. Our response to the problem was the attempt introduced in this paper to replace the domain of finite domain constraint programming with *integer linear programming,* which is perhaps more reliable in this respect. A further advantage of the approach is that it gives us an *anytime algorithm* [5] "for free".

2.2 Modelling Assumptions

Figure 1 shows a directed acyclic graph G modelling task interaction. Tasks are non-preemptive and have resource, precedence, and synchronisation requirements. The graph comprises vertices and edges representing tasks and precedence constraints, respectively. Each vertex is labelled T_i/c_i, where T_i is a task and c_i its estimated execution time in appropriate time units (seconds in this example). We denote the precedence constraint between tasks T_i and T_j in the graph by $T_i \rightarrow T_j$. Tasks without predecessors are called entry tasks and tasks having no successors are called exit tasks. We also assume that each task T_i requires a set of resources $\mathcal{R}_i = \{R_m\}$ for its execution where R_m denotes a resource of type m. Also, for each resource R_m, its available capacity at a given t point in time is given by $\mathrm{cap}(R_m, t)$.
Scheduling is a mapping of tasks to resources such that the specified precedence and deadline constraints are satisfied. The desired result is a feasible schedule specifying the *start times* (also referred to as *release times*) for each task T_i. It is also necessary to introduce some slack in this schedule to improve its robustness to execution uncertainties. In many cases, the necessary slack may be obtained by

appropriately dividing up the entire available *time frame* (i.e. the interval between the overall start and deadline time, $0\,s$ resp. $17\,s$ in the example) among the tasks.

Assume that tasks T_0 and T_1 start at $0\,s$, and that G must meet a deadline of 17 seconds, i.e. T_8 and T_9 must finish before $17\,s$. Note, however, that the longest path $T_0T_2T_4T_6T_8$ through G is only $7\,s$ long. This implies that a slack of $17 - 7 = 10\,s$ can be distributed to tasks along that path to retain some temporal flexibility during their scheduling. We now discuss a method aimed at distributing G's overall slack among tasks such that the slack added to each intermediate task is maximised. This process results in a *scheduling range* $[r_i, d_i)$ for each T_i where r_i and d_i denote the earliest release time and latest deadline, respectively.

2.3 Slack Distribution

Initially, only entry and exit tasks having no predecessors and successors, respectively, have their release times and deadlines fixed. In the slack distribution problem, the overall graph time frame must be distributed over each intermediate task such that all tasks can be feasibly scheduled on their respective resources. Slack distribution is NP-complete and various heuristics have been proposed to solve it. We use the approach proposed in [6] to maximise the slack added to each task in graph G while still satisfying its deadline D. The heuristic is simple, and for general task graphs, its performance compares favourably with other heuristics [12].

As part of the slack distribution, entry and exit tasks in the graph are first assigned release times and deadlines respectively. A path $path_q$ through G comprises one or more tasks $\{T_i\}$; the slack available for distribution to these tasks is:

$$slack_q = D_q - \sum_{i\,:\,T_i \in path_q} c_i \qquad (1)$$

where D_q is the length of the time frame of $path_q$ (i.e. the difference between the deadline and the release time of the path) and c_i is the execution time of task T_i along this path. The distribution heuristic in [6] maximises the minimum slack added to each T_i along $path_q$ by dividing $slack_q$ equally among tasks. During each iteration through G, a *non-extensible* path $path_q$ is chosen such that $slack_q/n_q$ is minimal, where n_q denotes the number of tasks along $path_q$. Then the corresponding slack is added to each task along that path. The deadlines (release times) of the predecessors (successors) of tasks belonging to $path_q$ are updated. Tasks along $path_q$ are then removed from the original graph, and the above process is repeated until all tasks are assigned release times and deadlines.

The graph in Fig. 1 is used to illustrate the above procedure. First, we select the path $T_0T_2T_4T_6T_8$ shown in boldface in Fig. 2(a); the total execution time of tasks along this path is $7\,s$, and as per the heuristic, a slack of $(17 - 7)/5 = 2\,s$ is distributed to each task. Once their release times and deadlines are fixed, these tasks are removed from the graph. Then path $T_1T_3T_5$ and finally path T_7T_9 is chosen, as shown in Figs. 2(b) and 2(c), respectively. In the former case, a slack of $\lfloor (13 - 5)/3 \rfloor = 2\,s$ is added to each task. Our algorithm leaves any remaining slack ($2\,s$ in Fig. 2(b)) unexploited, although it could be distributed to tasks with longer execution times to allocate the *relative* slack more evenly.

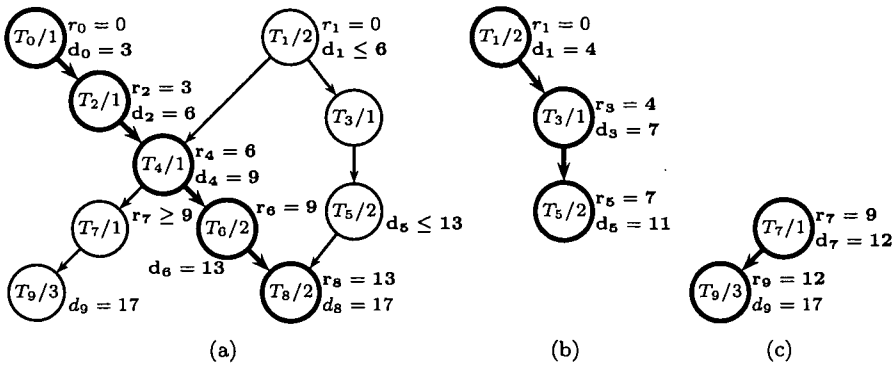


Figure 2: Steps corresponding to the the deadline assignment algorithm in [6]; the selected paths are shown as bold edges

3 ILP Model Formulation

In this section we first describe the part of the scheduling problem that remains after the slack distribution finishes, and present how it can be formulated as an ILP model. Two different formulations are shown, one where a contiguous sequence of uniform length *time slots* is assigned to each task, and another where one of a set of possible *predetermined intervals* is chosen for each task.

3.1 Interval Allocation

Once tasks are assigned deadlines, each T_i has a scheduling range given by $[r_i, d_i)$. However, to generate a feasible mapping of tasks to a limited number of resources, these scheduling ranges must be reduced appropriately to account for resource contention during task execution; we adapt concepts from interval scheduling [8] to solve this problem.

The scheduling range for T_i is first decomposed into a number of possibly overlapping intervals $\{I_{ij}\}$. Each I_{ij}, corresponding to the j^{th} possible scheduling interval for T_i is such that:

$$I_{ij} = [r_{ij}, d_{ij}) \text{ where } r_i \leq r_{ij} \leq d_i - c_i \text{ and } r_i + c_i \leq d_{ij} \leq d_i. \tag{2}$$

I_{ij} is also assigned a weight

$$w_{ij} = \frac{d_{ij} - r_{ij} - c_i}{d_{ij} - r_{ij}} \tag{3}$$

denoting the scheduling flexibility within that interval in terms of available relative slack.

Robust schedule generation can now be formulated as an *interval selection* problem where exactly one scheduling interval for each task must be selected such

that (i) at any point in the schedule, the overlapping task intervals do not consume more than the number of available resources and (ii) the sum of the interval weights is maximised.

3.2 Slot Based Approach

In [11], we proposed an ILP model using uniform length *time slots*, where the solution of a problem is an assignment of a contiguous set of slots to each task, such that the task can be executed in those slots without violating any of the constraints. The interval selection described above appears in this model only indirectly, since an interval is implicitly determined by the set of contiguous slots selected for the task. The model is shown on Fig. 3, and can be explained as follows. A boolean variable x_{it} corresponds to each task-slot pair, such that a specific task is scheduled to run in a particular slot if and only if the corresponding variable is assigned the value 1 in the solution. It is also necessary to introduce a set of auxiliary variables y_{ik}, such that y_{ik} is 1 if and only if the total number of slots (viz. the length of the scheduling interval) assigned to task T_i is exactly k. These values are used in the objective function to mask out the predetermined interval length weights.[1] The constraints ensure that resource capacities are not exceeded (4), that the intervals selected by the slot variables are contiguous to ensure non-preemptive execution (5), that tasks are not executed outside their scheduling ranges (6), and that the generated interval lengths are long enough to accommodate the tasks (7). Equations (8) and (9) describe the connection between the appropriate x_{it} and y_{ik} variables.

The major weakness of this approach is hidden in (5), the constraint which ensures that the scheduling intervals assigned to the tasks are contiguous. This involves moving a simple convolution window over the entire scheduling range and ensuring that there is not more than one 0-to-1 transition. Unfortunately, this is a nonlinear requirement, thus the number of ILP inequalities it can be expressed with is quadratic in the average size of the scheduling ranges, totalling approximately $\sum_i (d_i - r_i)^2$. Another difficult and inefficient detail of the model is the cost function, which cannot be expressed directly in terms of the slot variables, but requires the introduction of a large number of auxiliary variables, precisely $\sum_i (d_i - r_i - c_i)$ many.

3.3 Interval Selectors

To circumvent the shortcomings of the slot based model, the number of equations was reduced at the cost of increasing the number of variables, in the hope that the ILP solvers can cope better with the latter than the former. In the new model, a boolean variable was assigned to each selectable scheduling interval of each task defined in (2). A solution is an assignment of 0/1 values to these variables, such

[1] Weights must be calculated in advance and "hardwired" into the model, because as long as they are *nonlinear* in terms of interval length, they cannot be expressed explicitly in a *linear* model.

Index sets and constant parameters:

$$
\begin{aligned}
L &:= \{i \mid i \text{ is the index of task } T_i\} \\
\mathcal{R} &:= \{R_1, R_2, \ldots\} && \text{set of cumulative resources} \\
\mathcal{R}_i &:= \{R_{i1}, R_{i2}, \ldots\} \subseteq \mathcal{R},\ i \in L && \text{set of resources used by task } T_i \\
K_i &:= \{c_i, c_i + 1, \ldots, (d_i - r_i)\},\ i \in L && \text{interval lengths for } T_i \\
w_{ik} &:= (k - c_i)/k,\ i \in L,\ k \in K_i && \text{weight of interval of length } k \text{ for } T_i
\end{aligned}
$$

Variables:

$$
x_{it} = \begin{cases} 1 & \text{if interval for } T_i \text{ occupies slot } t \\ 0 & \text{otherwise} \end{cases} \qquad i \in L,\ -1^* \leq t \leq D
$$

$$
y_{ik} = \begin{cases} 1 & \text{if interval of length } k \text{ is selected for } T_i \\ 0 & \text{otherwise} \end{cases} \qquad i \in L,\ k \in K_i
$$

Maximise $\sum_{i \in L} \sum_{k \in K_i} y_{ik} w_{ik}$ subject to the following constraints:

Resource availability:

$$
\forall R_m \in \mathcal{R}, \forall\, 0 \leq t \leq D : \sum_{i \in \{i \mid R_m \in \mathcal{R}_i\}} x_{it} \leq \mathrm{cap}(R_m, t) \tag{4}
$$

Interval contiguity:

$$
\forall i \in L, \forall t \in \{r_i - 1, \ldots, d_i - 4\}, \forall l \in \{t + 2, \ldots, d_i - 2\} :
$$

$$
x_{it+1} - x_{it} + x_{il+1} - x_{il} < 2 \tag{5}
$$

Interval length:

$$
\forall i \in L, \forall t \in \{-1^* \ldots, r_i - 1, d_i, \ldots, D\} : x_{it} = 0 \tag{6}
$$

$$
\forall i \in L : \sum_{r_i \leq t \leq d_i} x_{it} \geq c_i \tag{7}
$$

Variable consistency:

$$
\forall i \in L \ : \sum_{r_i \leq t \leq d_i} x_{it} - \sum_{k \in K_i} k y_{ik} = 0 \tag{8}
$$

$$
\forall i \in L \ : \sum_{k \in K_i} y_{ik} = 1 \tag{9}
$$

*For technical reasons, the index t of x_{it} can take -1 as its value. Since a task cannot start at time -1, $x_{i,-1} = 0$ by definition.

Figure 3: The slot based ILP model

Index sets and constant parameters:

L : i task indices $\quad Z_i$: j interval indices for T_i, $i \in L$

$\mathcal{R}$: cumulative resources $\quad \mathcal{R}_i \subseteq \mathcal{R}$: resources used by T_i, $i \in L$

$I_{ij} := [r_{ij}, d_{ij}),\ i \in L,\ j \in Z_i$ $\quad j^{\text{th}}$ possible scheduling interval for T_i

$w_{ij} := \dfrac{d_{ij} - r_{ij} - c_i}{d_{ij} - r_{ij}},\ i \in L,\ j \in Z_i$ $\quad$ weight corresponding to interval I_{ij}

Variables:

$$x_{ij} = \begin{cases} 1 \text{ if interval } I_{ij} \text{ is selected for task } T_i \\ 0 \text{ otherwise} \end{cases} \qquad i \in L,\ j \in Z_i$$

Maximise $\displaystyle\sum_{i \in L} \sum_{j \in Z_i} x_{ij} w_{ij}$ subject to the following constraints:

Resource availability:

$$\forall R_m \in \mathcal{R}, \forall 0 \leq t \leq D \ : \sum_{\substack{i \in \{i \mid R_m \in \mathcal{R}_i\} \\ j \in \{j \mid t \in I_{ij}\}}} \sum x_{ij} \leq \text{cap}(R_m, t) \tag{10}$$

Singular interval selection:

$$\forall i \in L : \sum_{j \in Z_i} x_{ij} = 1 \tag{11}$$

Figure 4: The interval based ILP model

that exactly one out of all the variables belonging to a task is assigned the value 1, in addition to satisfying all the resource and temporal constraints. The complete model is shown in Fig. 4.

The constraints defined by (10) ensure that the time dependent resource capacities are never exceeded. For each particular resource and time slot, we select all the possible scheduling intervals containing this slot of all the tasks using the given resource. The sum of these 0/1 values equals the actual resource usage of any specific solution, therefore it must not be greater than the corresponding capacity. Constraint (11) encodes the requirement that exactly one interval is chosen for each task.

3.4 Jobs as Groups of Tasks

In addition to maintenance tasks, the application domain also operates with the concept of *jobs*. Regular aircraft maintenance consists of independent jobs (e.g. 56-DSI, an inspection scheduled about every 56 days), which are built up from smaller, interrelated tasks (e.g. remove wings to give access to the engine). When

Index sets and constant parameters:

M : k job indices $\quad N_k$: l job span indices for J_k, $k \in M$

H_{kl} := $[s_{kl}, e_{kl})$, $k \in M$, $l \in N_k$ $\quad l^{\text{th}}$ possible job span for job J_k

Variables:

$$y_{kl} = \begin{cases} 1 & \text{if job span } H_{kl} \text{ is selected for job } J_k \\ 0 & \text{otherwise} \end{cases} \qquad k \in M,\ l \in N_k$$

Consistent & singular interval selection (replacing (11)):

$$\forall k \in M, \forall l \in N_k, \forall i \in \{i \mid T_i \in J_k\} \ : \sum_{j \in \{j \mid I_{ij} \subseteq H_{kl}\}} x_{ij} = y_{kl} \tag{12}$$

$$\forall k \in M \ : \sum_{l \in N_k} y_{kl} = 1 \tag{13}$$

Figure 5: Extending the interval model with jobs

this is translated into a graph, each job constitutes a connected subgraph of the entire task graph, and slack distribution can be performed on these subgraphs independently. In addition, maintenance can only be performed while the aircraft is on the ground, and a job cannot be interrupted with a flight mission (for example because the aircraft is not yet completely reassembled). Assuming that mission planning always precedes maintenance scheduling, the interval based model must be extended to avoid scheduling jobs during and across missions.

Let us introduce the notion of jobs, denoted by J_k, which are sets of T_i tasks, and the concept of *job spans*, those time intervals within which jobs can be performed. A job span is denoted by $H_{kl} = [s_{kl}, e_{kl})$, and either *all* or *none* of the tasks constituting the job must be completed in it.

The overall algorithm is then modified as follows. Slack distribution described in Sect. 2.3 is executed for each job (i.e. a connected graph) and *for each job span separately,* assigning a separate scheduling range to each task of the job within each span. Then each of these scheduling ranges is used to generate possible scheduling intervals according to (2), with the added notational complexity that now there is more than one scheduling range per task. The remaining problem is again a singular interval selection for each task, but this time we must also ensure that for all tasks of a single job we select intervals from the same job span.

The extension of the model is shown on Fig. 5 (only new or modified elements are listed). For each job J_k, it introduces a second set of boolean variables, y_{kl}, one for each job span, which is 1 if and only if the corresponding span contains all tasks of the J_k in the resulting schedule.

Equation (11) is replaced with two new equations. Constraint (12) encodes the requirement that all the tasks of a job must be executed in the same job span. The

equation specifies that a particular job span is selected ($y_{kl} = 1$) if and only if all the tasks constituting the job have exactly one interval selected within that job span. Finally, (13) ensures that exactly one job span is selected for each job.

An obvious special case of the extended model is when there is exactly one job, containing all the tasks, and there is exactly one job span. Then the extended model behaves exactly like the simple interval based model on Fig. 4.

3.5 Efficiency Considerations

To avoid the explosion of the number of variables as the number of tasks and the size of the scheduling ranges ($d_i - r_i$) increase, we decided to limit the number of possible scheduling intervals per task. The algorithm we created to determine the intervals is as follows. For each task:

1. The number of intervals per job span is set to be proportional to the size of the span.
2. For each span, generate a set of possible starting points, at which all eventually created intervals will start. The number of points is chosen by keeping two goals in mind:
   - the number of intervals is not less than two per point;
   - the distance between neighbouring points is not less than the execution time of the task.

   The points themselves are distributed evenly within the job span. These rules help to maintain a healthy balance between the number of choices in the starting point and the length of the scheduling interval. And even though the rules are arbitrary, measurement results summarised in Sect. 4.3 indicate that these limitations do not effect the quality of the solutions significantly: the maximal objective values returned by the solvers using the latter model are not worse than those running on the former.
3. For each starting point, generate the required number of intervals (number intervals in the span divided by the number of points), in gradually increasing length from the minimally required up to a globally fixed multiple of the execution time.

Note that due to rounding and integer division, the number of actually generated intervals may not reach the limit. In our current solution, this remainder capacity is unexploited.

Figure 6 shows a robust schedule for the task graph displayed in Fig. 1, generated using an ILP solver on the defined model. Here we assumed that each task uses exactly one of the two available resources. The intervals corresponding to tasks T_5 and T_7 are shorter than the scheduling ranges in the solution, therefore they are shown in bold. The dotted sections are indicating the subintervals removed to satisfy the constraints.

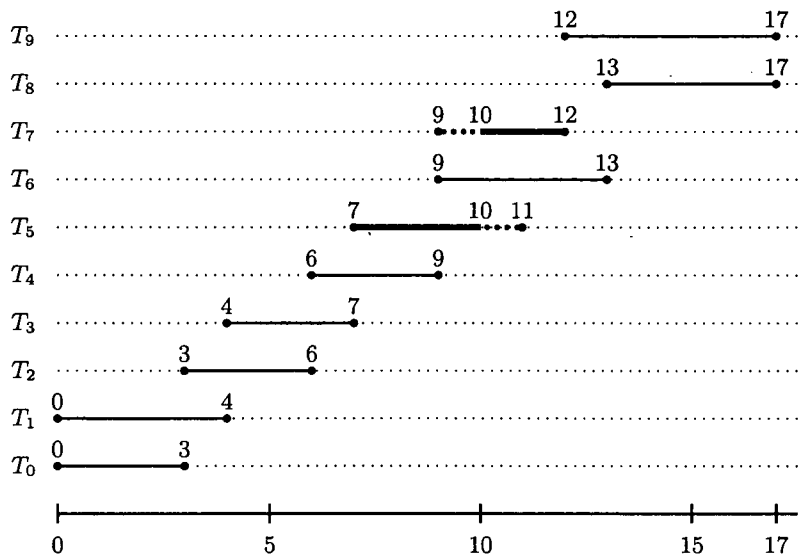


Figure 6: The robust schedule generated with the ILP model

4 Performance Evaluation

In [11], two ILP solvers were tested: a pure ILP called LP_SOLVE[2], a freely available generic linear programming solver [13], and a specialised 0-1 ILP solver targeting pseudo-boolean optimisation problems called PBS [14], which can also handle SAT formulæ. In order to use PBS, the integer constraints in the ILP model were converted to their appropriate pseudo-boolean and conjunctive normal forms. Now we added a second pure ILP solver called GLPK [10] to the list[3], which is also available under a free license.

We have tested the interval based model with real life data take from the software tool mentioned in Sect. 2.1, and verified that the ILP solvers were able to create valid schedules for them in an acceptable time frame. However, the slot based model was not elaborate enough (i.e. no notion of jobs) to handle these scenarios, and also these data represented only a small number of points in the entire problem space. In order to be able to perform a more thorough and systematic evaluation of the models, we turned to random problem generation. Here, our goal was to make the generated problems similar in structure to real life scenarios, so that the evaluation of the former would give us some feel for the performance of the solvers on the latter as well.

[2]We used version 2.0 of LP_SOLVE in our experiments. Since then, newer versions of the software have been released, at the time of writing this article the newest is version 5.5, which might (or might not) perform better on the test data sets.

[3]GLPK version 4.4 was used in the tests. The most recent version at this time is 4.13.

4.1 Expectations

Since the new model does not include constraints which could be encoded as clauses of a SAT formula, we expected a drop in the performance of PBS when moving from the slot based to the interval based model. On the other hand, since the number of constraints has decreased, we hoped that the pure ILP solver will tackle the problem better than before. As for the new GLPK solver, we did not have any previous experience to begin with.

4.2 Problem Generation

The random task graphs used in our experiments were obtained as follows. To generate a *directed acyclic graph* (DAG) with a specific number of tasks, a certain number of *layers* are filled by randomly distributing a number of independent tasks to each layer.[4] Next, we randomly link the edges between tasks in different layers. Finally, tasks are assigned execution times uniformly distributed between [2, 5] secs. A set of resource types $\mathcal{R} = \{R_m\}$, each with a specific capacity is also generated. In our experiments, these resources are distributed uniformly among tasks such that each task is allocated exactly one resource of a certain type. The original resource capacity can also be increased (decreased) as needed. Finally, the graph deadline D is set to $(1 + slack) \cdot p_{max}$, where p_{max} denotes the longest path length through the graph and *slack* is a user-specified value.

4.3 Analysis of Measurement Results

The slot based model has been evaluated with two solvers, LP_SOLVE and PBS. Table 1 summarises the performance of the two solvers given four resource types, each with a capacity of three. The experiments were performed on a 3.2 GHz Pentium 4 processor wit 1 GB of RAM. Graph deadlines are derived using $slack = 1.0$. The table shows the first solution (value of the objective function in the ILP model) returned by both solvers as well as the time taken to do so. (The resolution of the timer was 15 seconds in these experiments.) The solvers were then allowed to improve on their initial solutions up to a time-out period of five minutes and the best solution returned by the solvers after that period is also shown. If a problem is shown to be infeasible by a solver (i.e. it can prove that there is no solution), this fact is denoted by 'Inf.' in the appropriate cell, while a solver time-out without returning any solution is denoted by '—'.

For small numbers of intervals, the solutions returned by LP_SOLVE are superior to PBS at the cost of greater time overhead. For larger numbers of intervals, however, LP_SOLVE was unable to return a solution within the time-out period whereas PBS returned the first solution very quickly.

Now let us turn to the interval based model. It has been evaluated with both solvers used earlier, as well as GLPK, a second pure ILP solver. The results are

[4] The number of tasks per layer is chosen randomly from a specified range, and the number of layers is implicitly determined by the total number of tasks and the number of tasks chosen for each layer.

Table 1: Results of the slot based model with 4 resources of capacity 3, *slack* = 1.0

Tasks	Scheduling intervals	LP_SOLVE			PBS		
		First solution	Time (secs.)	Best solution	First solution	Time (secs.)	Best solution
~25	892	12.52	< 15	13.02	2.35	< 15	10.66
~50	2628	19.41	75	21.26	13.37	< 15	18.92
~75	2639	18.53	135	20.77	13.44	< 15	17.97
~100	3091	—	—	—	14.12	< 15	21.83
~150	7326	—	—	—	30.64	15	33.03

Table 2: Results of the interval based model with 4 resources of cap. 3, *slack* = 1.0

Tasks	Sched. ints	LP_SOLVE			PBS			GLPK		
		First sol.	Time	Best sol.	First sol.	Time	Best sol.	First sol.	Time	Best sol.
~25	892	13.02	< 5	13.02	11.22	< 5	11.22	13.02	< 5	13.02
~50	2628	28.00	< 5	28.00	13.82	< 5	16.42	27.99	< 5	28.00
~100	3091	32.81	< 5	32.81	17.02	10	17.02	32.73	< 5	32.81
~150	7326	63.4	25	63.4	32.06	15	32.06	63.41	< 5	63.43
~200	9167	72.77	25	72.82	—	—	—	72.72	10	72.82
~500	17613	157.78	275	157.78	—	—	—	157.78	15	157.78

summarised in Table 2. The test parameters and conditions were chosen to be identical to the previous tests (in fact, the very same task graphs were used) in order to make comparison possible, only in this case larger problems have been tested as well, since the speedup of the solvers using the new model permitted this increase in size. The time resolution has also been refined to 5 seconds.

The results clearly show that the interval based model suits the taste of the integer linear solvers much better. For all but the smallest problem, the solutions returned by LP_SOLVE are better than with the slot based model, and with much quicker response times. This solver has also been able to cope with problems containing 100–150 tasks, which earlier caused a time-out, and even with problems of 200–500 tasks, which were not even attempted. It is also worth pointing out that the quality of the first and the best solutions differ only minimally (if at all). The results of the GLPK solver are very similar to those of LP_SOLVE, only with smaller run times. (In fact, it would be interesting to observe how GLPK behaves with even larger problem sizes.) On the other hand, the performance of PBS is clearly poorer. Even though the quality of the first solutions is better than before for smaller problems, the quality of the best solutions has diminished. This change for the worse could be explained by the fact that while the slot based model contained a number of SAT encodable constraints, the interval based model does not, and we believe that

Table 3: Effect of *slack* values on solver performance

T#	*slack* = 0.5				*slack* = 0.8				*slack* = 1.0			
	int#	obj	LPS	GLPK	int#	obj	LPS	GLPK	int#	obj	LPS	GLPK
~25	436	9.48	0.03	0.13	747	12.52	0.04	0.23	892	13.02	0.05	0.25
~50	1338	19.66	0.11	0.34	1961	~ 24.7	0.18	0.53	2628	28	0.27	0.88
~100	1384	Inf.	0.12	0.28	2277	Inf.	0.31	0.48	3091	32.81	3.72	1.19
~150	3638	33.2	> 300	2.01	5839	56.65	> 300	3.76	7326	~ 63.4	> 300	4.3
~200	4234	Inf.	0.74	1.07	6886	58.79	60.99	2.7	9167	72.82	> 300	12.9
~500	8582	Inf.	1.99	2.38	14257	~ 126.3	> 300	30.32	17613	157.78	> 300	17.06

the main strength of PBS, which gave it an edge over LP_SOLVE, lies in the SAT solver core. Since it has not been able to exploit this feature with the interval based model, its performance has degraded.

In [11] we presented a table which emphasised the effect of increasing *slack* values on solver performance. It showed that when *slack* was increased, a larger number of possible scheduling intervals was generated, which in turn caused a larger search space and thus more time-outs. On the other hand, the robustness of the schedules improved where the solvers finished in time. Table 3 shows a similar data set for the interval based model, including results for the two ILP solvers. Since the objective values returned by LP_SOLVE and GLPK were always very close (and *almost* always equal), the table includes a single set of objective values, i.e. those of the best solutions found by both solvers. (A $\sim$ sign denotes where there was a minor difference in the values.) Two further columns per *slack* value show the *total run time* of the two solvers (i.e. the time required to be able to tell that the found solution is indeed the best).

It is interesting to see that there is a jump in the time values of LP_SOLVE, where it was not able to finish within the time limit any more. Nonetheless, it always found a solution[5], which was not even worse than the best solution found by GLPK. However, the time results of GLPK are convincing, it appears that GLPK scales well with the problem size.

5 Conclusions

This paper has addressed the problem of generating robust task schedules under explicit deadline constraints and proposed a new ILP-based solution. In addition to an earlier model of ours, we formulated a second ILP model whose solution maximises the temporal flexibility of the overall task schedule. This model was solved using three integer solvers LP_SOLVE, PBS and GLPK that use widely varying solution techniques. Our experiments show that while LP_SOLVE provides superior solutions for the smallest problems, it is outperformed by GLPK both in

[5]ILP searches can be considered *anytime algorithms* for practical purposes, knowing that they use the *branch-and-bound* algorithm, and assuming that the search tree is interspersed with solutions, which is apparently true our case.

speed and scalability. The SAT based PBS solver finished poorly in our tests. We believe this is because the strength of this solver lies in the SAT solver core, but our ILP model did not contain SAT encodable constraints.

5.1 Future Work

The tests clearly showed that even with the new model, the performance of the solvers degrades drastically as the problem size increases. To face the issue of scalability, we are experimenting with generating solutions in multiple passes, using a technique we call *rolling horizon*. First, scheduling ranges are determined as usual. The idea is then to generate a robust schedule for a relatively short period of the entire planning horizon in each pass, giving greater flexibility to the tasks scheduled at the end of this period, i.e. with the interval weights defined in (3) being modified to be monotonous in r_{ij}. Then in the next pass, the scheduling intervals selected for the trailing part (i.e. part of the output of the first pass) are used as new, reduced scheduling ranges (i.e. as input of the second pass). The second pass will then finalise these tasks by selecting a subinterval of these reduced ranges. This way each pass has the ability to slightly modify the decisions made in the previous pass near the "seams" without breaking any of the already satisfied temporal constraints.

The advantage of this "divide and conquer" approach is that it could help to keep the complexity of the problem linear in the number of tasks. When the ILP models are applied to the entire problem, the increase in run times is steeper than linear as shown in Sect. 4.3. By using a rolling horizon, however, the number of tasks per pass can be kept constant.

Acknowledgements

The work presented here has been completed at the *Institute for Software Integrated Systems*[6], Vanderbilt University in Nashville, TN in 2004. The authors would like to thank the institute, especially *Chris van Buskirk, Gabor Karsai* and *Himanshu Neema* for providing the context, application domain and the necessary resources for the research, and also for giving important ideas, scrutinising our claims and explanations.

Special thanks to *Péter Szeredi* at the *Department of Computer Science and Information Theory*, Budapest University of Technology and Economics for his valuable comments on this paper.

References

[1] Autonomous Negotiating Teams (ANT) Projects. `http://www.dsic-web.net/ito/programs/ant/projects.html`, 2001.

[6] ISIS Web address: `http://www.isis.vanderbilt.edu`

[2] Brucker, Peter. *Scheduling Algorithms.* Springer Verlag, Berlin, fourth edition, 2004.

[3] Bruynooghe, M. Enhancing a search algorithm to perform intelligent backtracking. *Theory and Practice of Logic Programming*, 4(03):371–380, 2004.

[4] Davenport, A.J., Gefflot, C., and Beck, J.C. Slack-based techniques for robust schedules. In *Proc. of the Sixth European Conf. on Planning (ECP-2001)*, Toledo, Spain, September 2001.

[5] Dean, T. and Boddy, M. An analysis of time-dependent planning. *Proceeding of Seventh National Conference on Artificial Intelligence AAAI*, 92:49–54, 1988.

[6] Di Natale, Marco and Stankovic, John A. Dynamic end-to-end guarantees in distributed real-time systems. In *Proc. of IEEE Real-Time Systems Symp.*, pages 216–227, San Juan, Puerto Rico, December 1994.

[7] Drummond, Mark, Bresina, John, and Swanson, Keith. Just-in-case scheduling. In *Proc. of the Twelfth Conf. on Artificial Intelligence (AAAI)*, pages 1098–1104, 1994.

[8] Erlebach, Thomas and Spieksma, Frits C. R. Interval selection: applications, algorithms, and lower bounds. *J. Algorithms*, 46(1):27–53, 2003.

[9] Gao, H. Building robust schedules using temporal protection—an empirical study of constraint-based scheduling under machine failure uncertainity. Master's thesis, Univ. of Toronto, Toronto, Canada, 1995.

[10] GLPK (GNU Linear Programming Toolkit). `http://www.gnu.org/software/glpk/glpk.html`, 2006.

[11] Kandasamy, Nagarajan, Hanak, David, Neema, Himanshu, van Buskirk, Chris, and Karsai, Gabor. Synthesis of robust task schedules for minimum disruption repair. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics (SMC'04)*, pages 5056–5061, The Hague, The Netherlands, October 2004.

[12] Kao, B. and Garcia-Molina, H. Deadline assignment in a distributed soft real-time system. 8(12):1268–1274, December 1997.

[13] LP_SOLVE: A Mixed Integer Linear Programming (MILP) solver. `http://tech.groups.yahoo.com/group/lp_solve`, 2006.

[14] Markov, Igor L., Sakallah, Karem A., Ramani, Arathi, and Aloul, Fadi A. Generic ILP versus specialized 0-1 ILP. In *Proc. IEEE Int. Conf. on Computer-Aided Design (ICCAD'02)*, pages 450–457, San Jose, California, November 2002.

[15] Müller, Tobias. *Constraint Propagation in Mozart.* Doctoral dissertation, Universität des Saarlandes, Naturwissenschaftlich-Technische Fakultät I, Fachrichtung Informatik, Saarbrücken, Germany, 2001.

[16] Smith, Stephen. OPIS: A methodology and architecture for reactive scheduling. In Zweben, M. and Fox, M., editors, *Intelligent Scheduling.* Morgan Kaufmann, 1994.

[17] The MICANTS Project: Model Integrated Computing and Autonomous Negotiating Teams for Autonomous Logistics. `http://www.isis.vanderbilt.edu/Projects/micants/micants.htm`, 2004.

[18] The Mozart Programming System. `http://www.mozart-oz.org`, 2006.

[19] Tsukada, T. K. and Shin, K. G. PRIAM: Polite rescheduler for intelligent automated manufacturing. 12(2):235–245, April 1996.

[20] Würtz, Jörg. Constraint-based scheduling in Oz. In Zimmermann, U., Derigs, U., Gaul, W., Möhrig, R., and Schuster, K.-P., editors, *Operations Research Proceedings 1996*, pages 218–223. Springer-Verlag, Berlin, Heidelberg, New York, 1997. Selected Papers of the Symposium on Operations Research (SOR 96), Braunschweig, Germany, September 3–6, 1996.

Acta Cybernetica 18 (2007) 257–274.

Adding XMP support to Firefox

Péter Jeszenszky*

Abstract

XMP (Extensible Metadata Platform) is an RDF-based framework of Adobe Systems Incorporated that supports the embedding of metadata in application files. If it becomes widely used on the web, it will provide a rich source of metadata to semantic web applications, too.

This paper presents a solution to add a unique feature to the popular Firefox web browser, the capability to extract XMP metadata from web resources. The solution is based on the Piggy Bank Firefox extension that turns Firefox into a "semantic web browser".

Keywords: Semantic Web, XMP, Firefox, browser extension, Piggy Bank

1 Introduction

The Semantic Web is a vision that aims at ensuring that web content is machine processable. Many researchers believe that it will be the next logical step in the evolution of the web. If the vision comes true, that will enable us to implement more intelligent information services than at present.

Computer processing of web content is an extremely difficult task because most of the currently available web content is intended for human consumption. Although CSS (Cascading Style Sheets) makes it possible to separate content and presentation, and is now a very popular and widely used technique, the task of locating relevant information in an arbitrary web page is still hopelessly difficult, and should require true AI capabilities.

Although the semantic web is a vision and the current web is far from "being semantic", the underlying standards and technologies, that are still experimental in many cases, have enormous potential.

One of the cornerstones of the Semantic Web is RDF (Resource Description Framework) [23], a flexible and simple framework to represent knowledge on the web. RDF provides a data model to represent metadata about resources in a machine-processable form. A resource may be anything that can be identified by an URI (Uniform Resource Identifier).[1] Resources are described by statements that

*University of Debrecen, Faculty of Informatics, E-mail: jeszy@inf.unideb.hu

[1]These URIs are not used to retrieve the content of the identified resources. This means that URIs may be assigned even to physical objects.

are ordered triplets of the form subject-predicate-object, in which the subject is the described resource itself, the predicate is a property, and the object is a property value. The meaning of such a triple is that the resource has a property, which has a particular value. The standard also defines an XML syntax (RDF/XML) to represent RDF metadata.

In order to ensure that web content is machine processable, data must be available in RDF. That assumes the existence of machine readable web pages written in RDF. The extensive availability of RDF metadata is the main problem. Although certain semantic web applications, for example FOAF [9] are based on the use of such machine readable web pages, it is not likely that millions of users will publish RDF data on the web.

XHTML 2.0 [26] provides an elegant and easy-to-use annotation mechanism, called metadata attributes module to embed RDF in XHTML in such a way that does not put additional burden on web page authors. However, XHTML 2.0 is a work in progress, and it is unknown when it will become a stable standard and supported widely. Furthermore XHTML 2.0 is not backward-compatible with previous XHTML versions.

A radically different approach to obtain RDF metadata is based on the use of RDFizers. The term "RDFizer" was suggested in the SIMILE [24] project. Actually, *RDFizers* is the code-name of one of the SIMILE subprojects [20]. It is a collection of tools to transform data from various sources to RDF. A list of the currently available converters is maintained at the web page of the RDFizers project. These tools are either developed by participants of the SIMILE project or other contributors. For example, they can convert BibTEX files, mailbox files, MS Outlook files or Java bytecode to RDF. The result of a conversion must be a set of RDF triples that reflect the semantics of the underlying data. In the case of highly structured data the conversion is often natural, otherwise human intervention is required. RDFizers try to return as much information as possible with no human intervention, and at the same time keep it potentially useful for the processing applications.

Piggy Bank [19] is a unique Firefox [8] extension that uses a similar technique to obtain RDF data from web pages. (This is discussed later in Section 4.)

XMP [1] is an RDF-based framework that allows metadata embedding in application files providing a rich source of RDF metadata for semantic web applications.

This paper presents a solution based on Piggy Bank that offers a new web browser feature, i.e. the capability to extract embedded XMP metadata from the resources that are accessible from the current web page.

In Section 2 a brief overview of XMP is presented. Next, Section 3 reviews how XMP is supported for the time being. Section 4 introduces Piggy Bank that forms the basis for this work. Section 5 presents the new browser feature together with the implementation details. Finally, Section 6 discusses other alternative uses of the solution presented, showing that it may be utilized as a more general tool to obtain metadata about arbitrary resources.

2 XMP

2.1 What is XMP?

XMP is an RDF-based metadata framework developed by Adobe Systems Incorporated, that provides the following:

- a data model,
- a storage model,
- metadata schemas,
- rules that describe how to embed XMP metadata in various application files.

These together constitute XMP.

The data model makes it possible to associate properties with resources in order to describe them, similarly to the data model of RDF. A resource may be a file, or a portion of it, that may be meaningful to a processing application in itself, and that is also a distinct logical component of the file structure. (For example an image imported into a PDF file may be a resource. On the other hand, a range of pages in a PDF document can not be considered as a resource, since the PDF file format does not recognize such a structure.)

The storage model provides an implementation of the data model, it uses a subset of the RDF/XML [22] syntax to represent XMP metadata. Metadata describing a particular resource are serialized as RDF/XML and may be embedded in the resource itself in an XMP packet. The packet is a wrapper around the RDF/XML data that is surrounded by easily recognizable delimiters. An example XMP packet is shown in figure 1. The XMP specification [3] provides for the way of embedding of these packets in many common file formats, such as GIF, JPEG, PDF, PNG, PostScript, TIFF and others.

XMP schemas are sets of predefined metadata elements that can be used by various applications to characterize a wide range of resources, such as electronic documents, digital images, audio and video files.[2] Each schema is identified by a unique namespace and typically contains related metadata terms that can be used to describe resources of a particular kind, or to describe resources from a particular viewpoint. Several standard metadata schemas are provided as part of the XMP specification. For example, the Camera Raw schema defines properties that may be associated with image files produced by a digital camera.

2.2 Why is XMP important?

XMP has obvious advantages:

- It provides a standard and file-format-independent way to annotate digital images and other resources.

[2]Currently XMP does not use RDF schemas or ontologies to define properties. Standard schemas are provided in the form of tables in the specification.

- Embedded XMP metadata will be available to virtually every application. Even an application with no knowledge of the file format may scan a file for embedded XMP packets. (However, this is not recommended, as discussed later.)
- Metadata is shipped together with the embedding resource and thus cannot be lost.

It is not an exaggeration to say that XMP opens up exciting new possibilities for digital photography and image editing applications. If it will be widely used on the web, a rich source of metadata will be available to semantic web applications that may be utilized effectively.

```
<?xpacket begin="" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/">
  <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about=""
        xmlns:dc="http://purl.org/dc/elements/1.1/">
      <dc:format>application/pdf</dc:format>
      <dc:title>
        <rdf:Alt>
          <rdf:li xml:lang="x-default">
            Test File
          </rdf:li>
        </rdf:Alt>
      </dc:title>
    </rdf:Description>
    <rdf:Description rdf:about=""
        xmlns:xap="http://ns.adobe.com/xap/1.0/">
      <xap:CreateDate>2006-12-10T10:00:00Z</xap:CreateDate>
      <xap:CreatorTool>
        pdfeTeX 3.141592-1.21a-2.2 (Web2C 7.5.4)
      </xap:CreatorTool>
    </rdf:Description>
    ...
  </rdf:RDF>
</x:xmpmeta>
<?xpacket end="r"?>
```

Figure 1: An XMP packet fragment.

3 Current XMP support

3.1 Commercial software

XMP is a flagship product of Adobe Systems Incorporated. Their software products, for example Adobe Acrobat, Adobe FrameMaker, Adobe GoLive, Adobe InDesign and Adobe Photoshop support XMP by default. To view or edit XMP metadata in an Adobe application the user must select the "File Info" dialog in the "File" menu. However, the author has experienced problems with Adobe Photoshop CS3 in the case of the PNG file format: XMP metadata can not be added to PNG files. This does not make any sense, since PNG is an open format.

3.2 Open source software

In respect of open source software, the situation is critical. Popular open source image and photo editing applications (for example GIMP) do not support XMP for the time being. Similarly, none of the currently available open source office applications (e.g. OpenOffice.org) can embed XMP metadata in the documents they produce.

The good news is that forthcoming versions of GIMP [10] will provide XMP support, making a significant step forward.[3]

Some of the few XMP-aware open source applications are mentioned here:

- The xmpincl [27] LaTeX package allows the embedding of an XMP packet in a PDF file that is produced by pdfLaTeX. The user must supply XMP metadata as an RDF/XML document. Producing such an RDF/XML document manually may be an extremely difficult task even for an experienced TeX user.

- PdfLicenseManager [18] is a Java application that focuses on licensing metadata. As the name suggests it supports only PDF files, and can display, insert and update Creative Commons licensing information that is stored in an embedded XMP packet.

- ExifTool [7] is a comprehensive set of Perl modules for reading and writing metadata in image, audio and video files. It runs on multiple platforms, and supports not only a wide range of file formats, but XMP also. A command line interface is available to the end-users. It is a great tool, although in the case of many formats it provides only read access to XMP metadata.

3.3 Developer support

Adobe XMP Toolkit (XMP SDK) [2] is Adobe's official XMP library that allows developers to add XMP functionality to their application programs. It is intended to

[3]The next version will be released as GIMP version 2.4 and it is still under development. According to a post to the GIMP developer mailing list it is likely that the metadata editor feature with XMP support will be available only in version 2.6.

be cross-platform and runs on multiple operating systems, namely on UNIX/Linux, Windows and Macintosh.

The SDK currently consists of two distinct components:

- The XMPCore component provides an implementation of the XMP data model and allows applications to parse, manipulate and serialize XMP. It must be emphasized that XMPCore reads and writes RDF/XML only.
- The XMPFile component makes it possible to locate XMP packets in application files. The detected XMP metadata is returned to the processing application and then may be manipulated using XMPCore. XMPFile also allows applications to add XMP metadata to files and offers metadata update facility. It supports a rich set of file formats, including AVI, JPEG, MP3, MPEG, PostScript, TIFF and WAV.

The SDK is distributed in source code form. Both the XMPCore and XMPFile components are implemented in C++. A Java version of the XMPCore component is also provided. Note that the Java support offered by the SDK is partial only, since the XMPFile part is not implemented in Java.

Another major deficiency is that building the full SDK on UNIX/Linux systems is not supported, the XMPFiles part is compilable only on Windows and Macintosh. This platform dependency together with the lack of file format support in the case of Java are major problems that may prevent XMP becoming more widely used and popular.

The XMP SDK is open source software. Previous versions were distributed under a license called ADOBE SYSTEMS INCORPORATED – OPEN SOURCE LICENSE. However, this license is not included in the list of open source licenses maintained by the Open Source Initiative [17] and its compatibility with the GNU GPL license is questionable. This nonstandard license did not allow the SDK to be used in many open source projects. The issue generated debate in the developer community.

As an alternative to XMP SDK the project exempi [6] has been created by a developer. It is a C/C++ XMP library that is distributed under the GNU LGPL license. Unfortunately, it is still under development and does not support file formats at the moment. According to the web site of the project, version 2.0 will be a port of Adobe's XMP SDK to UNIX/Linux systems. JempBox [14] is another open source Java library that provides roughly the functionality of the XMPCore component of Adobe's XMP SDK, but offers no file format support.

Recently, Adobe has changed the licensing policy and the latest version of the SDK is now available under the terms of the widely used and accepted BSD License.

4 Piggy Bank

4.1 What is Piggy Bank?

Piggy Bank is a Firefox extension that turns the popular Firefox browser into a "semantic web browser". It means that:

- It employs techniques and solutions providing a novel browsing experience.
- It utilizes semantic web technologies.

Piggy Bank is developed as an open source software in the SIMILE project, that is a collaboration of W3C, MIT Libraries and MIT Computer Science and Artificial Intelligence Laboratory (MIT CSAIL).

4.2 What is it good for?

Piggy Bank can extract "pure" information from a web page. Then the following options are available:

- Collected information can be saved and stored locally.
- Collected information can be uploaded onto communal information repositories called semantic banks. Semantic banks enable information to be shared with other people.
- Relevant information can be filtered out of the collected information flexibly.
- Novel display methods are available to visualize the collected information. For example, it is possible to display collected information items on a geographic map or on a timeline. The graph structure of collected information can be examined as well.

4.3 How does it work?

If Piggy Bank can extract "pure" information from the current web page, it is indicated by a small "data-coin" icon on the status bar of Firefox. Clicking on the icon will instruct Piggy Bank to collect the available "pure" information. Actually, RDF data will be harvested during the processing of the web page.

Piggy Bank can extract RDF from a web page in the following two cases:

- If the HEAD section of the HTML document contains LINK elements to RDF data, that is either in RDF/XML or in N3 format.
- The URL of the web page matches the URL pattern of an active screen scraper.

A screen scraper is a special JavaScript code that can transform the content of particular web pages to RDF. A screen scraper has an URL pattern. Web pages having an URL that matches that pattern can be processed by the scraper. Typically, the HTML structure of the web pages offered by a given web site is hardwired into a screen scraper that will enable it to extract all relevant information from a page. A screen scraper not only can process the current web page, but it may also load and process related web pages, or invoke a web service to obtain auxiliary information to enrich the content.

Screen scrapers must be installed in Piggy Bank in order to operate. A few screen scrapers are available at the Piggy Bank web site to collect pure information from popular web sites, such as Flickr or the ACM Portal. Another option is to write an own scraper. Solvent [25] is another Firefox extension that is also developed in the SIMILE project. It is a great tool that makes it easier to develop screen scrapers.

Piggy bank uses Longwell [15] to display the collected information, which is web-based faceted RDF browser.

5 XMP in Firefox

5.1 The main idea

Since the author is an enthusiastic fan of both Firefox and metadata standards including XMP, it was a fairly obvious, and at the same time a very desirable goal for him to add XMP support to his favourite web browser.

A Firefox extension is presented here that provides the following functionality:

- All the embedded XMP metadata can be extracted from the images and the resources that are linked to current web page with one click for browsing.
- XMP metadata can be extracted for browsing from individual images and links, too.

To the best of our knowledge none of the currently available web browsers offer such a function, thus it is a completely new browser feature. It is probable that such a functionality should also play an important role in the popularization of XMP.

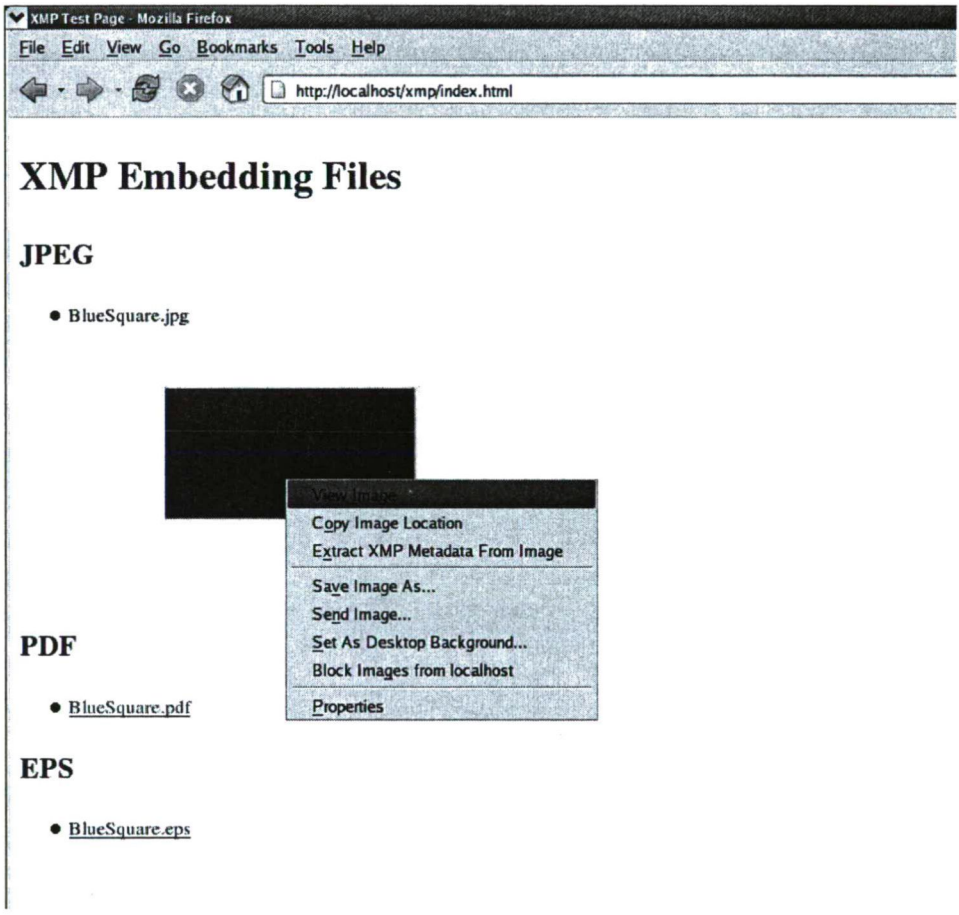


Figure 2: Right clicking on an image in Firefox. Select "Extract XMP Metadata From Image" to view embedded XMP metadata.

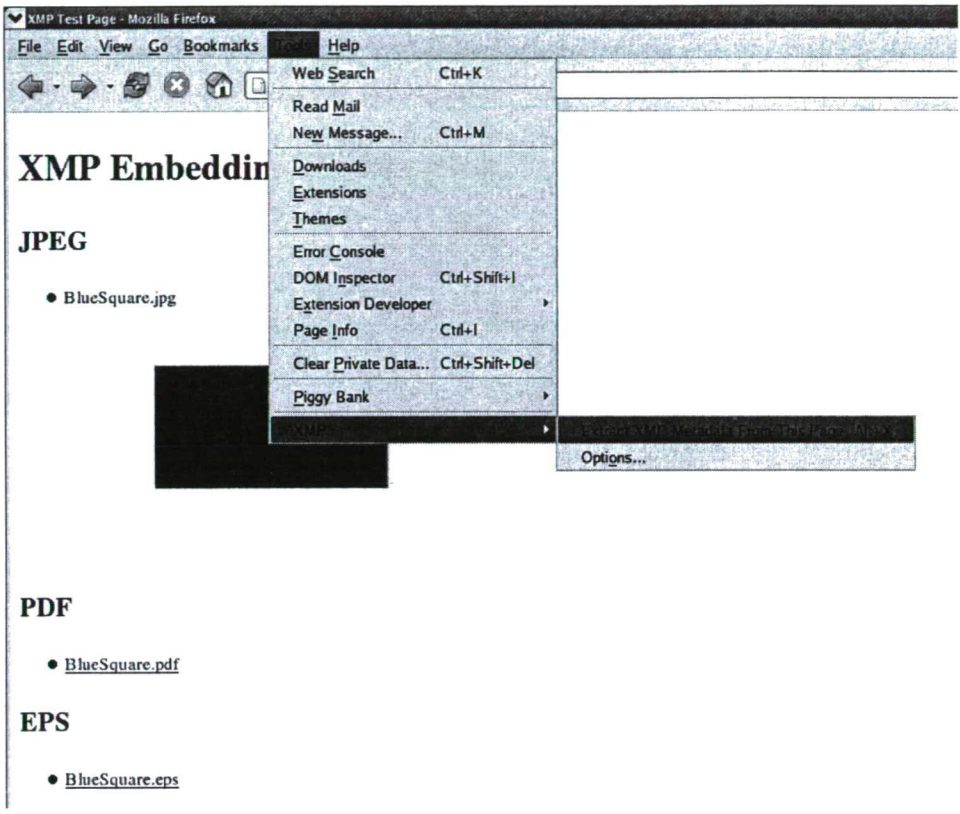


Figure 3: The XMP menu in Firefox.

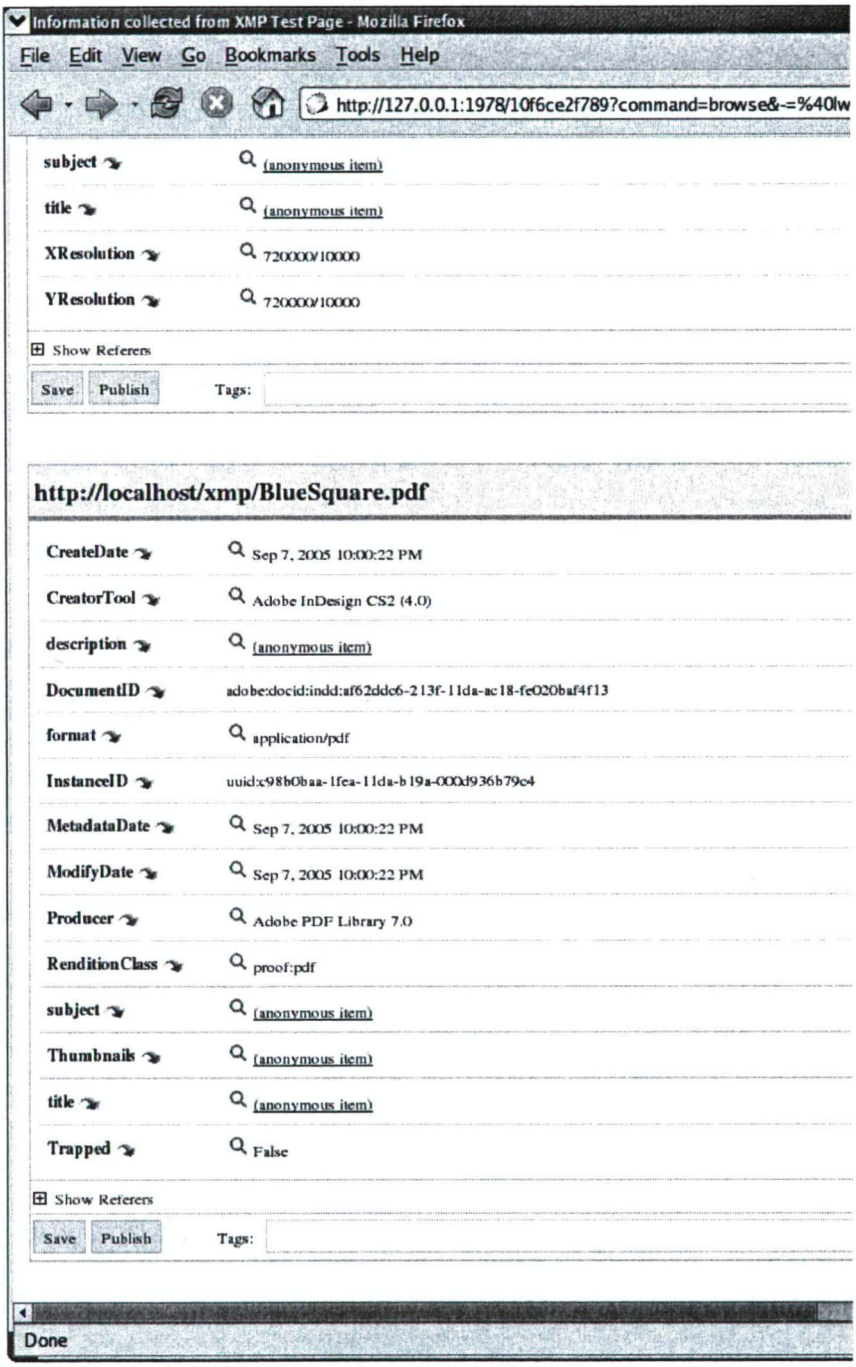


Figure 4: Browsing the extracted XMP metadata in Firefox.

5.2 Implementation

As the above functionality fits well into Piggy Bank, it has been chosen to serve as a basis for the implementation. The author has extended Piggy Bank with XMP support.

The implementation consists of two distinct parts:

- XMP extraction functionality that has been developed independently of the browser, as discussed below.
- An user interface part that integrates XMP functionality with the web browser.

The user interface of Piggy Bank is written in XUL [28]. XUL (XML User Interface Language) is a cross-platform user interface language developed by the Mozilla [16] project, that is used primarily in Mozilla applications and web browsers that are also based upon on the Gecko layout engine. The user interface of extensions, and also the entire user interface of Firefox itself is written in XUL. XUL makes use of many standards and techniques, including CSS, JavaScript and RDF. The user interface part of our implementation is also written in XUL.

Only few additional user interface elements were necessary to be added to the extension. Notably an XMP menu (containing the "Extract XMP Metadata From This Page" and "XMP Options" items) has been added to the "Tools" menu (see figure 3), an XMP options dialog has been created (see figure 6) and two appropriate items ("Extract XMP Metadata From Image", "Extract XMP Metadata From Link") have been added to the context popup menu that appears when the user right clicks on an image or a link (see figure 2). Figure 4 shows the final result (extracted XMP metadata) in the browser.

Currently, XMP extraction functionality is available to Piggy Bank as a RESTful web service. The web service accepts an URL in a HTTP [11] GET request and extracts XMP metadata that is returned as an RDF/XML document. This is a convenient solution which completely separates XMP logic from Piggy Bank code. Since screen scrapers may also utilize web services, this approach fits well with Piggy Bank, too. A great advantage of the solution is that it does not require XMP software to be installed on the client side, implementation details are hidden behind the web service.

The XMP extractor web service operates as follows:

1. The web service gets an URL in a HTTP GET request. XMP metadata will be extracted from the resource identified by the URL.
2. The web service initiates a HTTP HEAD request using the URL to query the MIME type and the length of the resource.
    - If the resource is not found or the XMP extractor web service does not support its format determined by the MIME type, then an appropriate response is returned, indicating the error.

- It is also an error if the content length of the resource exceeds a configurable limit.

3. The web service begins to retrieve the content of the resource and scans it for embedded XMP metadata.
   - If the XMP packet is found, it is postprocessed then returned as an RDF/XML document.
   - If XMP metadata is not found, an appropriate response is returned.

A schematic view of the above scenario is also shown in figure 5.

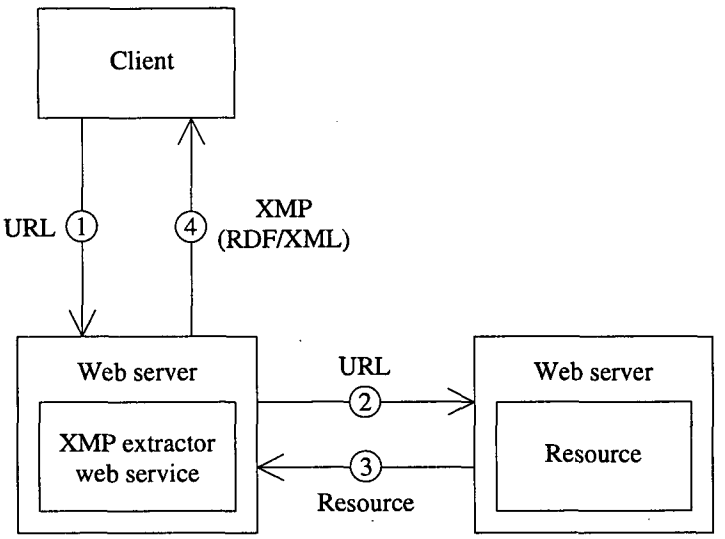


Figure 5: XMP extractor web service.

The web service is implemented in Java using the JAX-WS API [12, 13] and deployed to Apache Tomcat [4]. Java was a natural choice, as it is the "native language" of the author. Moreover, it is not tied to any particular vendor's operating system or platform.

To extract XMP metadata the author did not make use of Adobe's XMP SDK at all. The main reason lies in the previously mentioned licensing problems of the SDK that had existed when this work was done originally and also in portability problems. Since then problems have been solved only partially. Although the license has been changed, the XMPFiles part of the SDK that can perform XMP extraction is still not available on UNIX/Linux systems, as mentioned earlier in subsection 3.3. The author found this platform dependency annoying, as he prefers platform-independent solutions (he also prefers Linux).

Although there is a platform-independent Java version of the XMPCore component, it may be used only to parse the extracted XMP metadata. However, that is

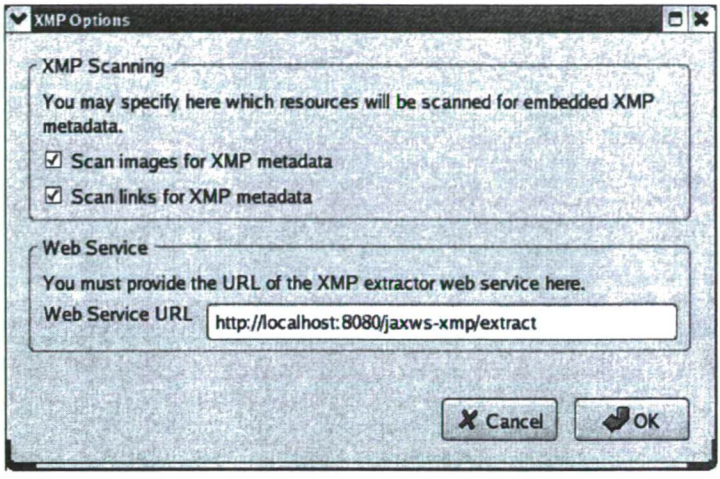


Figure 6: The XMP Options dialog in Firefox.

unnecessary. If an XMP packet is available in RDF/XML, it can be fed into Piggy Bank directly after a postprocessing step is applied on it.

As a workaround, the web service utilizes a proprietary Java class library that has been developed by the author to extract XMP metadata from files (see figure 7).

The ExtractorFactory class and the Extractor interface makes up the core of the API. In the following, a software module that is used to extract XMP metadata from a file is called an extractor. Our implementation uses format specific extractors, as does the XMPFile component of Adobe's XMP SDK also.

Even an application that is not aware of the file format may scan a file for XMP metadata, since the task is to detect special XMP packet delimiter character sequences. However, this is not recommended, and should be considered only as a last resort, because:

- XMP packet delimiters may be encoded as UTF-8, UTF-16 or UTF-32, and the byte order may also vary between big-endian and little-endian. This implies that old and tried pattern matching algorithms can not be used directly. (It is unknown which byte sequence will represent a specific character.)

- XMP metadata may be stored in the embedding file in compressed form. In that case, an application that is not aware of the file format will not be able to detect XMP packets.

- If there are multiple packets, the application will not be able to determine which one is the main. For example, the incremental save feature of the PDF file format allows applications to update XMP metadata without overwriting, keeping the obsolete XMP packet.

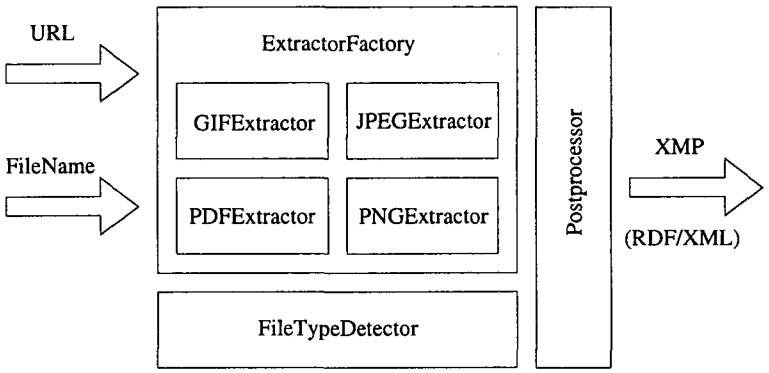


Figure 7: Java classes for XMP extraction.

- The XMP specification defines the way of embedding in the case of many popular file formats. Knowing the file format may help to locate XMP metadata more efficiently, without scanning the entire file for special delimiters. (The file format may determine possible locations.) This is very important, especially in the case of large files.

Thus the only reliable and efficient way to locate XMP metadata in a file is to use a processing application that recognizes the file format.

The ExtractorFactory class manages the available extractors. Currently the GIF, JPEG, PDF and PNG file formats are supported. (The development of a PostScript extractor is already in progress.) Each of these extractors is "smart" enough to know where to look for XMP metadata and so it does not scan the entire file.

Additional extractors can be plugged-in as well. When an URL or a file name is passed to the ExtractorFactory class, it returns an extractor object that will be able to perform XMP extraction. The ExtractorFactory class tries to determine the file format, then selects the appropriate extractor class.

The previously mentioned postprocessing step is not explained in detail because of its low-level technical nature. The result of XMP extraction is an XMP packet, which in fact is an XML document fragment enclosed by special delimiters. During postprocessing the XMP packet must be modified in order to get an appropriate and meaningful result, that may be processed by an RDF application. For example, lines 1, 2, 25 and 26 in figure 1 must be removed, otherwise traditional (non XMP-aware) RDF parsers may not be able to process the input as valid RDF/XML.

6 Generalization

As already mentioned in Section 1, RDFizers are tools to transform various data sources to RDF, thus making them available to semantic web applications. It should be noted that the XMP extractor web service is actually an RDFizing web

service. It is obvious that the web service may be replaced by any other RDFizing web service that accepts an URL and transforms the identified resource to RDF. The result is viewable in the browser just like XMP metadata.

The author is currently experimenting with an RDFizing web service that may be regarded as a generalization of the original one. It will continue to provide XMP extraction functionality but will be able to handle a much broader range of resources. In the new web service the ExtractorFactory class is replaced by a class named RDFizerFactory that maintains a list of available RDFizers. Each RDFizer handles a specific file format and transforms resources of that type to RDF. The web service accepts an URL, determines the MIME type of the resource, then selects the appropriate RDFizer to transform it to RDF.

The list of available RDFizers utilized by the web service contains the original XMP extractors and a few additional RDFizers that are also developed by the author [21]. Namely, an RDFizer that handles the RPM package file format used by many Linux distributions, and another one that handles torrent files used by the BitTorrent P2P file sharing system. Using the RDFizer web service the user may explore RPM package information (for example description, required dependencies, list of provided files) or torrent file metadata in the browser, by simply right clicking on a link that points to an RPM package or a torrent file.

The developers of Piggy Bank may be working on something similar. When the author mentioned the XMP extractor web service to the Piggy Bank developers on a mailing list they said that they also had a web service called Babel [5] that is now publicly available and that can convert between various formats, including RDF. At the moment just a few formats are supported, but we think it would be easy to extend it to support many other RDFizers also.

Although Piggy Bank collects information from web pages utilizing scrapers that may also use an RDFizer web service, the currently available scrapers do not exploit this opportunity. These scrapers return RDF by processing only the text of web pages, locating relevant information. The novelty of the work presented in this paper is that it demonstrates that Piggy Bank may also be used to obtain RDF from non-textual resources (XMP embedded in binary files).

7 Conclusions

The paper introduces a new feature that works with the Firefox web browser, the ability to display XMP metadata embedded in resources that are accessible from the current web page.

The implementation is based on the Piggy Bank Firefox browser extension, that turns Firefox into a "semantic web browser", and also employs a new web service.

The web service accepts an URL and scans the identified resource for embedded XMP metadata that is returned. Using the presented browser extension, images and hyperlinks can be processed by the web service with a single mouse click, and returned XMP metadata is viewable in Piggy Bank. The novelty of the presented work is that it uses Piggy Bank to obtain RDF from non-textual resources (for

example, from images on a web page).

Although work presented in this paper may appear to be a small step towards the semantic web, it has resulted in a quite general tool that can be used right now, and may be of interest to a wider audience.

References

[1] Adobe XMP
`http://www.adobe.com/products/xmp/`

[2] Adobe XMP SDK
`http://www.adobe.com/devnet/xmp/`

[3] Adobe XMP Specification
`http://www.adobe.com/devnet/xmp/pdfs/xmp_specification.pdf`

[4] Apache Tomcat
`http://tomcat.apache.org/`

[5] Babel
`http://simile.mit.edu/babel/`

[6] exempi
`http://www.figuiere.net/hub/blog/?Exempi`

[7] ExifTool
`http://www.sno.phy.queensu.ca/~phil/exiftool/`

[8] Firefox
`http://www.mozilla.com/firefox/`

[9] FOAF
`http://www.foaf-project.org/`

[10] GIMP – The GNU Image Manipulation Program
`http://www.gimp.org/`

[11] Hypertext Transfer Protocol – HTTP/1.1
`http://www.ietf.org/rfc/rfc2616.txt`

[12] Java API for XML Web Services (JAX-WS)
`http://java.sun.com/webservices/jaxws/`

[13] JAX-WS Reference Implementation
`https://jax-ws.dev.java.net/`

[14] JempBox – XMP Compatible Java Library
`http://www.jempbox.org/`

[15] Longwell
http://simile.mit.edu/longwell/

[16] Mozilla
http://www.mozilla.org/

[17] Open Source Initiative
http://www.opensource.org/

[18] PdfLicenseManager
http://media.polito.it/masala/plm2_index.html

[19] Piggy Bank
http://simile.mit.edu/Piggy_Bank

[20] RDFizers
http://simile.mit.edu/RDFizers/

[21] RDFizers by Peter Jeszenszky
http://www.inf.unideb.hu/~jeszy/rdfizers/

[22] RDF/XML Syntax Specification
http://www.w3.org/TR/rdf-syntax-grammar/

[23] Resource Description Framework (RDF)
http://www.w3.org/RDF/

[24] SIMILE Project
http://simile.mit.edu/

[25] Solvent
http://simile.mit.edu/solvent/

[26] XHTML 2.0
http://www.w3.org/TR/xhtml2/

[27] The xmpincl macro package
http://www.ctan.org/tex-archive/macros/latex/contrib/xmpincl/

[28] XML User Interface Language (XUL)
http://www.mozilla.org/projects/xul/

Conference on Hungarian Computational Linguistics

Guest Editor:

Zoltán Alexin

Department of Software Engineering
University of Szeged
Szeged, Hungary
alexin@inf.u-szeged.hu

Preface

This issue of *Acta Cybernetica* contains seven papers whose preliminary versions appeared in Hungarian language in the proceedings of the 4th Conference on Hungarian Computational Linguistics.

The conference was held in Szeged on December 8–9, 2006 and its aim was to provide a forum for researchers working on Hungarian computational linguistics and speech processing, see `http://www.inf.u-szeged.hu/mszny2006/`.

After the conference, the authors were invited to submit completed versions of their papers to *Acta Cybernetica*. All submitted papers were then subjected to the normal refereeing process of the journal. Altogether seven manuscripts were submitted, all of them have been finally accepted, three after major revision. We thank the authors and the referees for their help in the preparation of this issue.

The 5th Conference on Hungarian Computational Linguistics was held also in Szeged on December 6–7, 2007.

Zoltán Alexin

Guest Editor

Acta Cybernetica 18 (2007) 279–291.

Topic and language specific internet search engine*

Domonkos Tikk[‡] György Biró[‡] Ferenc P. Szidarovszky[†§]
Zsolt T. Kardkovács[‡] and Gábor Lemák[¶]

Abstract

In this paper we present the result of our project that aims to build a categorization-based topic-oriented Internet search engine. Particularly, we focus on the economic related electronic materials available on the Internet in Hungarian. We present our search service that harvests, stores and makes searchable the publicly available contents of the subject domain. The paper describes the search facilities and the structure of the implemented system with special emphasis on intelligent search algorithms and document processing methods.

Keywords: Internet search engine, Web crawlers, Document processing, Text categorization

1 Introduction

In the past 5 years the percentage of home internet users increased from 6% to 30%, and there are about 1.5 million people today who use the internet at least 1 hour a month. The increasing number of internet surfers brought on the expansion of content provision as well, but in addition to the traditional business-based content provision — thanks to the technical development and support — user driven content providing plays an increasingly important role. This expansion generated an increasing demand for search services, the development of which was also stimulated by the dynamic increase of the domestic online advertising market. Furthermore the companies and the academic domain realized the scientific challenge in internet searching, and from the year 2000 research workshops — mainly financed by NKFP

*This work was supported by Program GVOP, Grant no. GVOP-3.1.1.-2004-05-0130/3.0. Domonkos Tikk was partly supported by the János Bolyai Research Scholarship of the Hungarian Academy of Science. Zsolt Kardkovács was partially supported by Mobile Innovation Centre.

[†]Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, H-1117 Budapest, Magyar Tudósok krt. 2., Hungary E-mail: {tikk,szidarovszky,kardkovacs}@tmit.bme.hu

[‡]Textminer Ltd., H-1029 Budapest, Gyulai P. u. 37., Hungary E-mail: gbiro@gmail.com

[§]Szidarovszky Ltd., H-1392 Budapest, P.O. Box 283., Hungary E-mail: ferenc.szidarovszky@szidarovszky.com

[¶]GKI Economic Research Co., H-1092 Ráday u. 42–44., Hungary E-mail: lemakg@gki.hu

and IKTA programs — started to develop new search algorithms and -intelligences. In the past 6 years both the academic and business domains attempted to develop new search methods that tried to improve search speed and efficiency through automatic text processing. In most cases, the utilization of these initiatives didn't take place, which motivated us to establish an R&D project that also targets the commercial utilization.

The goal of our project is to create a topic-oriented internet search service prototype that applies semantic-based technologies and novel visualization methods. We selected the topic of economy as the focus of the prototype. The service enables the user to search the largest available economic content collection, and also allows to view and download the documents if the consent of the content provider or digital right holder is given.

With the topic of economy we focused on a compact but thriving segment of the market of internet search services with a diverse user group. The economic contents in broader sense may equally interest the average users (small individual investors, lay users), corporate leaders, business consultants, and decision-makers, as well as users from academia — lecturers, researchers, students.

The project intends to provide such a search service that satisfies users information need more accurately as current state-of-the-art general internet search applications, and with a new way of visualization that may shorten the time of the search. The know-how created in the project offers the opportunity to set-up similar topic oriented search services for other thematic areas and languages as well.

This paper is organized as follows. First, Section 2 describes the designed search functionalities of the system, then Section 3 presents the structure and main components of the system. Section 4 investigates some important features of the system, including the document processing flow, and the web harvesting module. Section 5 presents the results of related projects. Finally, Section 6 concludes the paper.

2 Supported search functionalities

When defining the functionalities of the system, our goal was to provide to the users:

- advanced search possibility,
- enhanced support for browsing the search results visually,
- adaptive search refinement option.

One possible way to improve search efficiency is to enable the user to define the topic of the search. This option helps the search service to capture the meaning of the user query and to get a more accurate idea about their information need — e.g. when processing polysemous queries. This may also decrease the result set's size and simultaneously the number of irrelevant results. The topic-oriented navigation

and search result browsing can be achieved by thematic organization of the searchable content. The availability of a hierarchical category system with appropriate coverage (topic taxonomy) of the broad topic is a prerequisite for this purpose. The general worldwide search services also provide such a search option (see e.g. Google Directory, Yahoo Directory, the former Zeal search of the Looksmart group[1]), but the significance of this search option is much larger when it concerns a search service on a unilingual and thematically limited document collection. In the case of general search engines, it is much more difficult to create and maintain a fully detailed topic taxonomy with required coverage, and it is also a challenging task to sufficiently fill up the taxonomy with quality content. The focused search topic our project alleviates the difficulties that arise in the maintenance of a dynamically changing topic taxonomy of diverse contents.

Eventually, the user may search for similar documents starting from a sample one — possibly created by him- or herself. General search engines do not support queries longer than a certain limit, therefore if the sample document is not indexed by the search engine — that is quite likely in the case of an own document — these engines are unable to properly execute this task.

The results of a former search can also be a starting point for the search refinement. But, it is far from trivial for the average user — even after having gone through some elements of the search list — how the query should be effectively expanded or modified. This activity can be efficiently supported by offering candidates keywords taken from the search result set.

Based on the previous considerations, we set the following search functionalities:

- keyword based search with taxonomy support,
- similar document search based on user's sample,
- browsing in fixed topic taxonomy,
- query refinement based on keywords from the search result set.

3 The structure of the system

The system of the search service comprises four main components that are depicted on Figure 1. The main components are

- web crawler,
- natural language processing module
- indexer and categorizer module
- graphical user interface.

Next we describe the task and operation of each component.

[1] `http://www.google.com/dirhp`, `http://search.yahoo.com/dir`, `http://www.zeal.com`

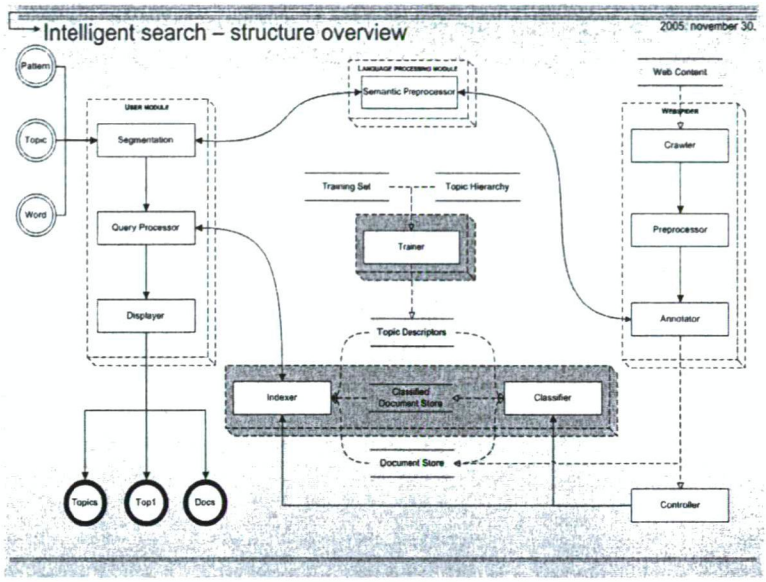


Figure 1: Structure overview of the system

3.1 Web crawler

The web crawler harvests the contents of selected topic-related web pages, and converts them to the prescribed XML format used by the system. Currently, the crawler collects contents from about 50 sources, which are mostly national economic content providers, but also includes relevant pages from some leading portals and regional news providers too.

3.2 Natural language processing module

This module converts the documents with different formats coming from different sources. The document processing flow has been defined to be able to perform different text representation techniques (such as word or character n-gram based), and to integrate various external natural language processing tools.

3.3 Indexer and categorizer module

The indexer engine creates and maintains the index catalog that serves as the basis of answering queries. The categorizer engine administrates and determines the category information to support the taxonomy based search of the system. The categorizer component performs supervised machine learning, i.e. it learns the significant words/expressions of each category of the taxonomy by means of sample training documents. This feature is implemented by integrating HITEC hierarchical text categorizer engine [5, 9]. On the one hand, HITEC's engine is

applied to create category profiles consisting of weighted words/expressions, and on the other hand, HITEC predicts category labels of unlabelled documents being harvested from the web or coming from other sources. On Figure 1 we illustrated by a gray background that these were externally developed modules of the system.

3.4 Graphical user interface (GUI)

The GUI provides the search functionalities to the users, forwards queries towards the search engine and displays and visualizes the results.

3.5 Creation of the taxonomy and training environment

In order to guarantee the efficiency of the internet search service and the quality of the search, it is essential to have a taxonomy that simultaneously represents the selected topic with appropriate details, and vaguely covers topics weakly related to the main theme. We used the subject hierarchy of the Central Library of the Corvinus University Budapest (CUB) as a starting basis of the economic taxonomy used in our system. The graph of the original subject taxonomy included different relation types between subject headings (broader/narrower, used/unused term, related concept), and included undirected cycles. Due to these inconveniences, it couldn't be applied directly for our task that requires an acyclic directed taxonomy based on broader/narrower term relations. The modification of the taxonomy was done by the librarians of CUB in cooperation and with guidance of our stuff. As a result, we got a taxonomy that includes all the subject-headings of the library's original subject taxonomy as categories. During the build-up of the taxonomy some new, connecting taxonomy nodes have also been created. The final taxonomy has 2397 nodes starting from 16 top level categories. Based on the results of preliminary tests, this structure has been further modified by merging rare categories and hence decreasing their number to 1383.

To complete the training environment of the system we needed sample training documents that sufficiently represent the categories of the taxonomy. For this task, we obviously used the subject taxonomy of CUB, since doing so we gained numerous training samples, particularly those electronically available documents that were indexed and stored by the Central Library of CUB. This initial document set has been expanded first by acquiring the electronic versions of already annotated documents, and second, by annotating other electronic documents of the topic.

4 Operation of the system

4.1 Document processing flow

We differentiate two document types in the system: training and harvested documents. The only difference is the present/absent of the category label: training documents have category labels, while harvested ones don't. The user queries are handled analogously as harvested documents in the processing flow except that they

are not stored. The original format of documents can vary. The system processes HTML, PDF, DOC, RTF and plain text files, and converts them to the internal XML representation format. The internal XML format is capable of representing documents at any processing stage of the document processing flow, and though being particularly optimized for text mining tasks, it also can be easily converted to any standard XML document scheme (such as, e.g., NewsML, TEI). At the creation of the DTD the main points were that

1. it should be able to code the required textual and meta-information;
2. the storage capacity to store the XML format of documents should be minimal.

This latter point is crucial in both keeping the storage need of the document corpus as low as possible, and reducing time and memory requirements of the document processing algorithms. The first requirement is guaranteed with a relatively flexible structure definition, while the latter one was obtained by short XML element names and by minimizing the set of required attributes. The values of XML elements are designed to store the textual information of the documents, while the additional meta information (e.g. grammatical features) is stored in attribute values.

The work flow of document processing is presented on Figure 2. One can observe that the document processing flow is identical for each document type (training, harvested, query). After XML conversion the module *Merger/Splitter* unifies the character encodings of the documents. The *Text Extractor* component employs various natural language processing tools, such as:

- *Stemmer*: The system offers two alternatives for this task. First, it includes an implementation of timid stemmer algorithm [4], second, it can employ the stemmer of the *HunMorph* package [10]. The XML format is able to store different grammatical parsing alternatives of a given word (see *g[rammar]* element), such as e.g. various word stems. This information is stored in the stem attribute of g.
- *POS tagger*: This approach also exploits a function of the HunMorph package. The usual implementation of an index catalog is word stem based. This solution merges homonym words. In order to alleviate this deficiency the system stores *[stem, pos-tag]* information in the index catalog. The part-of-speech of the words are stored in the *pos* attribute of *g* element.
- *Word filter*: This component is necessary for the query refinement. When offering keywords for this search feature, the function or stop-words should be disregarded. The filter works based on a stop-word list and patterns. The process sets the *sw* attribute of the filtered word's *g* element to true. These words are used in the index catalog; therefore they cannot be eliminated from the text.
- *Sentence segmenter*: This module segments the text into sentences. Its output is used when displaying the most relevant context of a document. It is a rule-based module: when finding a candidate sentence separator, the matching

rules determine whether it is a valid sentence separator or not. The rules are assigned signed weight. If the analyzed text fragment of the sentence separator matches several rules the final decision is taken by the aggregation of matching rules. An abbreviation dictionary is also used at the process. The detected sentences are labeled with *s[entence]* element.

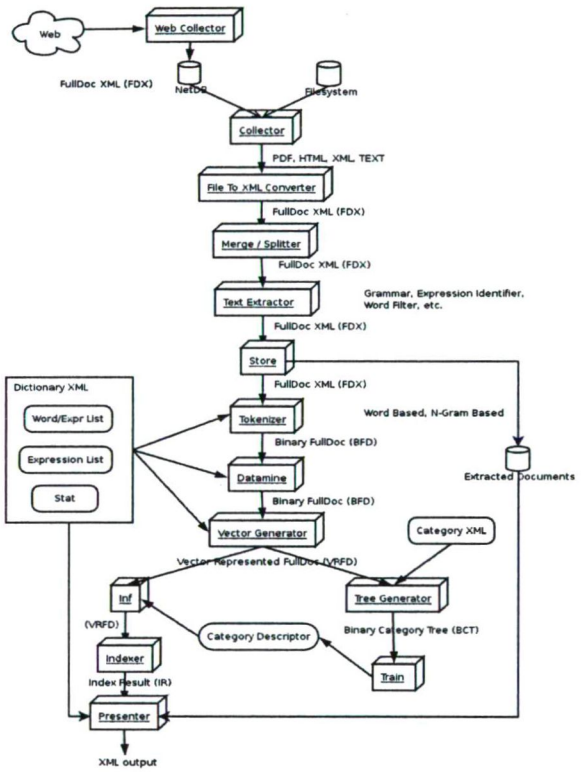


Figure 2: The document processing flow of the system

One can easily code the results of arbitrary natural language processing tools with the internal XML format, e.g. the output of a morphological parser, or named entity recognizer [8]. We will investigate the effectiveness of the integration of such external tools in terms of improving search efficiency.

Each document is stored in three versions coming from different processing stages. In addition to the original format, we save the raw XML file, and final fully processed XML format for each document. This is performed by the *Store* module, which also assigns the category information to the document, if available. The URLs of the different versions are specified in the appropriate attribute of the *document* element, while the category information is encoded into the *mc* (main category) and *sc* (secondary category) fields.

After this stage documents are converted into numerical representation, which is done by the *Tokenizer* method. The *Datamine* module searches frequent word sequences in the tokenized version, and creates new tokens from them. Finally the inner representation form is generated by the Vector Generator module. This creates two vectors: one for indexing — this also includes stop words; and one for categorization and keyword-determination — where stop-words are excluded. The former one applies TF-IDF weighting scheme, while the latter one uses entropy-weighting [3].

Finally, training documents are applied to train the classifier (*Train*), the category information of other documents are inducted by means of the classifier model (*Inf*). The documents are indexed in the next step. The *Presenter* module displays the required category and keyword information towards the user interface, and attaches the matching document for the query. Here the tokenized versions are converted back to text format.

4.2 Web harvesting

The task of the web crawler is to keep track of, archive and annotate the contents of selected web pages. The harvesting has two main functions:

1. Monitoring of selected web-sites, downloading new information (briefly: harvesting).

2. Conversion and structural annotation of downloaded documents.

The selected web pages have to be visited regularly. The harvesting is performed by a so-called daemon — termed as *Crawler* — that has four simple functions: start, harvest, stop and delay.

The *Crawler* starts the download process, where the downloaded content is typically a pre-specified, regularly visited URL — the main page of a portal, or an RSS-channel[2]. Having downloaded the raw content of the visited page, the Crawler analyzes and annotates its content, selects the documents to be retrieved, saves them to the document archive, and finally preprocess them before the next step of the document processing is started.

The specification of the *Crawler* does not contain topic specific details, therefore it is applicable for any topic domain with proper parameterization. Having said this, one should observe that there is no uniform algorithm to separate the relevant and irrelevant parts of the document. The automatic selection of the coherent and connected segments of a document with 100% accuracy cannot be guaranteed even with deep semantic analysis of the text. (On accuracy we mean that the selected segment includes all topic-related material from the downloaded text, and only that.)

However, the relevant text can be identified with about 90% accuracy by means of the analysis of some key factors, such as, the displayed and real title of the

[2]Real Simple Syndication; `http://blogs.law.harvard.edu/tech/rss`

document, and the mutual relation of the following 5 fields: date, author, title, abstract, text body.

In practice, the annotation of different portal sites is performed by means of a limited set of auxiliary software, called plug-ins. These programs convert the input HTML text into the required XML format defined by *fulldoc* schema using the structural characteristics of the harvested site.

Since the characteristics of harvested sites can change in time, the *Crawler* has to be prepared for such structural changes that are originated from, e.g., the modification of the portal engine, or the revision of the web site. In such cases, the harvested documents are likely to become invalid. Therefore all XML output of the *Crawler* is parsed syntactically before archiving. It often happens that the output created by an outdated plug-in misses some relevant fields (e.g. title or text body), and this can be detected by parsing. The structural changes of the harvested portals can thus be detected with high probability. The syntax parsing check might only fail to detect a problem with the input, if the structure of the portal does not change, but the topic of the content is altered. Such modifications cannot be detected automatically in the current version of the *Crawler* without continuously re-retrieving articles from the portal.

4.3 Graphical user interface

The visualization of the search results was an important factor at the design of the system. The result documents are displayed in two alternative ways: the document map and the document list. The document map is a visual display form (see Figure 3), which places the result documents on a circle, where the distance from the origin represents the similarity of the document to the query. Documents of different categories are marked with different colors. The content of a document can be viewed by clicking on the document icon. The document list (see Figure 4.) is a traditional form of displaying the search results. Here, by default, the query refinement tab are also displayed. This tab can also be switched on at the document map view. The taxonomy of the service can be applied to perform category based filtering of the result (also at the initial search). The service will be made publicly available at the beginning of 2008.

5 Similar initiatives

The goal of the *Information & Knowledge Fusion* project (IKTA3-181/2000) was the analysis, design, and establishment of a new intelligent knowledge warehousing environment, capable of efficient information- and knowledge management on specific vertical application domains [2, 1]. The project developed a knowledge-based information retrieval system for the financial sector that is based on various data sources, and generates reports according to the needs of the field of application in a structured format.

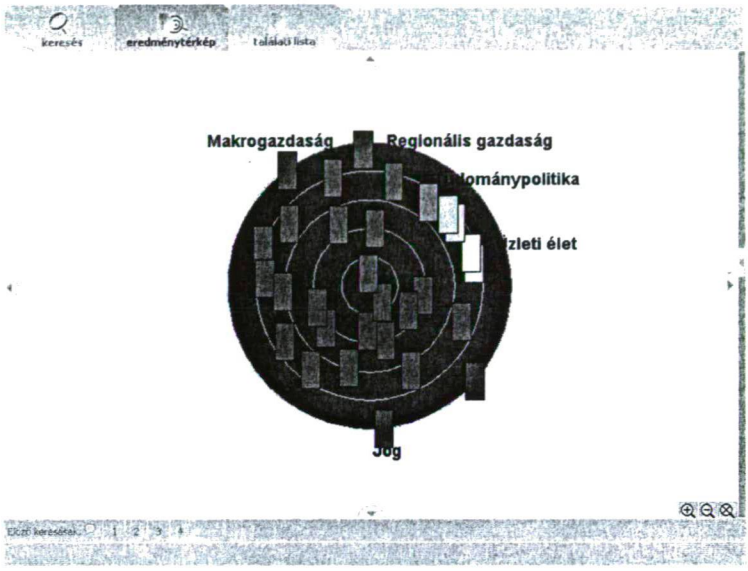


Figure 3: The document map view

The project, named *In The Web of Words* (NKFP 0019/2002) aimed the establishment of a complex internet based search/query application, introducing new search methods in both the deep web search (online accessible contents hidden from traditional search engines, such as databases or not directly linked contents), and image search [6, 7]. Supporting image search, a visual thesaurus has been developed that serves as a text based image-content category system used in characterizing and indexing image contents. In the deep web search, the system allows search queries in the form of natural language sentences in the Hungarian language.

The *Semantically organized lexical network and internet content research* (IKTA5-123/02) project intended to produce an internet content search technology based on a semantically organized and interpreted lexical network. The project tried to reach its goal by researching the possible connections of meaning centers (the basic unit of the lexical network that is a structure of natural language designators — words, phrases, etc. — organized around a common meaning), and by building the appropriate connecting elements. The semantically organized, communicative lexical network — assembled by the linking of meaning centers — is developed in a way that it can efficiently operate in applications based on language-technology (like natural language processing systems, interpreted information-search in electronic texts and structured text bodies, content monitoring, machinery translation, context- and style sensitive spell checker).

The *Knowledge based Hungarian semantic search system* project (IKTA 00148/2002) was led by the National Traumatology and Emergency Institute (NTEI). In addition to the statistical control of data, the project includes de-

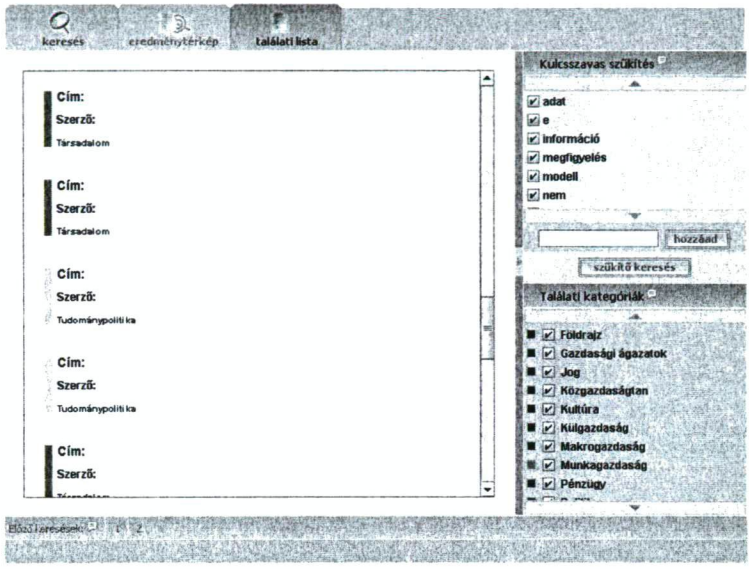


Figure 4: The document list view

termination of extensional relations based on meaning coherency (clustering) and logical connections. The developed technology recommends the use of "knowledge retrieval" by means of machine learning and processes based on neural network, together with classical data mining methods (drilling-up, drilling-down etc.). The test of the system executed at the NTEI, because the necessary medical ontology was available there, and at them it is vitally important to find the relevant document as fast as possible. The developed knowledge based search technology can be used widely as a search engine for libraries, archives, medical-, legal- or corporate data- and knowledge-bases or any commercial applications where the goal-oriented searching has an important role.

The *WebKat*[3], developed by Neumann-ház in 2002 in the framework of a national R&D project, is a topic map based solution for searching Hungarian contents. The topic map supports the search by the visualization of its structure with internal relations. This search engine retrieves documents from a dedicated a separate database, so the search is not performed directly on the Internet.

Polymeta[4] is a general purpose meta-search engine, that enables the user to search the Internet by using simultaneously several selected search engines. An aggregated result page is generated from the various result sets. On this page, single hits are displayed in order of importance. A "table of contents" is also created, where hits are clustered and displayed as directories. This allows the user to capture the various meanings and associations related to the query and the

[3]`http://www.webkat.hu`
[4]`http://polymeta.hu/`

results.

The new initiative called *Vipkereső* cannot be reached in full functionality by this time, but according to recent news, it will be a full text web search engine, offering image-, blog- and news search options as well.

6 Summary

This paper presents the functions and structure of a topic oriented semantic-based Internet search engine developed in the framework of a Hungarian R&D project. The prototype of the system performs intelligent search on Hungarian economic related content. The paper describes in detail the main components of the systems, the document processing flow, the document harvesting solutions, and also describes the graphical user interface.

References

[1] Cs. Dezsényi et. al. Tudásalap információk kinyerése: az IKF projekt. *Tudományos és Műszaki Tájékoztatás*, 2004. `http://www.neumann-haz.hu/tei/publikaciok/2004/biro_ref_ikf_hu.html`.

[2] Dezsényi, Cs., Varga, P., Mészáros, T., Stratusz, Gy., and Dobrowiecki, T. Ontológia-alapú tudástárház rendszerek. `http://nws.iif.hu/ncd2003/docs/ahu/AHU-118.htm`.

[3] Salton, G. and Buckley, C. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1998.

[4] Tikk, D., Biró, A. Törcsvári Gy., and Bánsághi, Z. Szótövező eljárások hatása magyar szövegek automatikus kategorizálásánál. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 430–434, Szeged, Hungary, 2005.

[5] Tikk, D., Biró, Gy., and Yang, J. D. Experiments with a hierarchical text categorization method on WIPO patent collections. In Attok-Okine, N. O. and Ayyub, B. M., editors, *Applied Research in Uncertainty Modelling and Analysis*, number 20 in Int. Series in Intelligent Technologies, pages 283–302. Springer, 2005.

[6] Tikk, D., Kardkovács, Zs. T., Andriska, Z., Magyar, G., Babarczy, A., and Szakadát, I. Natural language question processing for hungarian deep web searcher. In Elmenreich, W., Haidinger, W., and Tenreiro Machado, J. A., editors, *Proc. of the IEEE Int. Conf. on Computational Cybernetics (ICCC'04)*, pages 303–308, Vienna, Austria, 2004.

[7] Tikk, D., Kardkovács, Zs. T., and Magyar, G. Searching the deep web: the WOW project. In *Proc. of the 15th Int. Conf. on Intelligent Systems Development (ISD'06)*, Budapest, Hungary, 2006. To appear.

[8] Tikk, D., Szidarovszky, F. P., Kardkovács, Zs. T., and Magyar, G. Magyar nyelvű kérdő mondat elemző szoftver. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 455–459, Szeged, Hungary, 2005.

[9] Tikk, D., Yang, J. D., and Bang, S. L. Hierarchical text categorization using fuzzy relational thesaurus. *Kybernetika*, 39(5):583–600, 2003.

[10] Trón, V., Halácsy, P., Rebrus, P., Rung, A., Simon, E., and Vajda, P. Morphdb.hu: magyar morfológiai nyelvtan és szótári adatbázis. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 169–179, Szeged, Hungary, 2005.

Acta Cybernetica 18 (2007) 293–301.

Hungarian named entity recognition with a maximum entropy approach

Dániel Varga* and Eszter Simon†

Abstract

In the analysis of natural language text a key step is *named entity recognition,* finding all complex noun phrases that denote persons, organizations, locations, and other entities designated by a name. In this paper we introduce the **hunner** open source language-independent named entity recognition system, and present results for Hungarian. When the input to **hunner** is already morphologically analyzed, we apply the system together with the **hunpos** morphological disambiguator, but **hunner** is also capable of working on raw (morphologically unanalyzed) text.

Keywords: natural language processing, computational linguistics, named entity recognition

1 Introduction

In the machine analysis of natural language documents we often seek to answer questions in terms similar to those used by humans: *who* is this document about, *where* is the action taking place, *how much money* is involved, and so on. By *named entity recognition* (NER) we mean an algorithm that takes natural language text (typically, in document-sized chunks rather than word by word or sentence by sentence) as input, and identifies all persons, locations, organizations and similar entities that are designated by a name. The name can be a single word (proper noun) such as *Budapest* or a complex phrase such as *Budapesti Műszaki és Gazdaságtudományi Egyetem Média Oktató és Kutató Központ.* Even when the eventual goal is more remote (e.g. machine translation, information extraction, or information retrieval), NER is a useful intermediate stage of processing.

The trivial algorithm that identifies those phrases as named entities which are written capitalized works well for English, Hungarian, and many other languages that capitalize proper names, but of course it fails both for languages like German

*Budapesti Műszaki és Gazdaságtudományi Egyetem – Média Oktató és Kutató Központ, E-mail: daniel@mokk.bme.hu

†Budapesti Műszaki és Gazdaságtudományi Egyetem – Kognitív Tudományi Tanszék, E-mail: esimon@cogsci.bme.hu

where capitalization conventions are different, for languages like Arabic or Chinese that do not have a separate upper- and lowercase, and for text that lacks casing (e.g. the output of speech recognition). Also, because the trivial algorithm is prone to false positives sentence-initially and to false negatives in text written in anything but the most carefully edited prose, it is important to develop methods that are less error prone.

The NER task, as posed by the MUC and CoNLL competitions [8], is to identify disjoint chunks of the input token sequence as named entities and to annotate these chunks with a small set on named entity categories such as PERson, ORGanization, LOCation, and MISC (all other named entities). The evaluation of an automatic NER algorithm is through comparing its output to a manual annotation. Typically, the algorithm itself learns its parameters from a manually annotated corpus (supervised learning).

For major languages, many dozens of papers were published on NER algorithms. Almost every currently known supervised machine learning technique was used. Some examples: BBN IdentiFinder [7], and Zhou and Su [14] apply Hidden Markov Modeling. Borthwick [2] [1], and Chieu and Ng [3] apply the maximum entropy approach. Sekine et al. [9] use decision trees. There are not too many language-dependent components of these, and other similar systems. Still, for Hungarian, we are only aware of one quantitative study of a NER system which is based on machine learning methods: Szarvas et al. [11] published results on their NER system based on C4.5 decision trees with Boosting. Their system achieved state-of-the-art accuracy for English, and for Hungarian it reached an accuracy of 94.77% CoNLL F-score on the Szeged NER Corpus [10]. (See Subsection 5.1. for the definition of CoNLL F-score.)

In Section 2 we describe the corpus, and in the Section 3 the external resources we used. We detail the architecture and implementation of our system in Section 4, and describe the methodology and the results of our experiments in the concluding Section 5.

2 The corpus

For the training and evaluation of our system, we used the Szeged NER Corpus [10]. At the time of writing, this is the only named entity annotated Hungarian corpus with a size suitable for supervised learning.

The Szeged NER Corpus is a more than 220 thousand token subset of the Szeged Corpus [4], manually annotated for named entities. A distinct characteristics of the Szeged NER text is its thematic homogenity: it only contains various subgenres of business news. This means that organization names are very highly represented. This category dominates the others in freqency.

The annotation of the corpus follows the tagset and annotation conventions of CoNLL [8]. This means that we used the following tagset: person names (`PER`), organization names (`ORG`), location names (`LOC`) and miscellaneous other named entities (`MISC`). In the Szeged NER Corpus, the `MISC` category mostly contains

brand names and financial acronyms.

An example sentence from the corpus: „[Sam DiPiazza]$_{PER}$, a [PWC]$_{ORG}$ vezérigazgatója szerint az [EU]$_{ORG}$ által már kötelezővé tett úgynevezett [GAS]$_{MISC}$ az eddiginél nagyobb betekintést ad az adott cég pénzügyeibe." (According to [Sam DiPiazza]$_{PER}$, chief executive of [PWC]$_{ORG}$, the so-called [GAS]$_{MISC}$, already enforced by the [EU]$_{ORG}$, gives more insights into the finances of the given firm.)

3 Gazetteers

Though 'gazetteer' originally means geographical directory, in the context of the NER task the phrase is simply used as a list of names. We assembled various gazetteers to be incorporated into our system.

- Hungarian and common non-Hungarian last names
- Hungarian and common non-Hungarian first names
- names of Hungarian cities
- country names in Hungarian
- Hungarian street names
- Hungarian organization names
- international organization names
- suffices for company names
- suffices for street names
- financial acronyms

In the first six cases, our source was an aggregated version of a Hungarian phone book, and a web database. The lists of Hungarian organization names and street names were cleaned of suffices with automatic methods. The common suffices (e.g. Inc., Ltd. for organizations, Street, Sq. for places, or in Hungarian 'Kft.', 'Rt.', 'utca', 'tér', respectively) were extracted, and moved into separate lists. The international organization list was kindly provided to us by György Szarvas and Richárd Farkas.

There was just one case when analyzing the development corpus lead to the inclusion of a new dictionary: the lexicon of financial acronyms. The development corpus contained several stock market index names (DAX, Libor, Nasdaq), which were sometimes marked as `ORG` instead of `MISC` by the algorithm. To solve this problem, we extracted such stock market terms from a web-based financial knowledge base. We note that using this lexicon did not improve the performance on the test corpus, and even decreased it slightly. The reason for this is that most of

these terms occured in the development part of the corpus, i.e. the split between the test and the development corpus was not a random one (see Subsection 5.1 for more details).

The gazetteers incorporated into our final system (except the case of financial terms) were finalized before the inspection of the train and development corpora. During the tuning of the system to the development corpus, we have found serious cases of over- and undergeneration in the gazetteers. Since correcting these errors did not improve accuracy significantly, we reverted to the original, uncorrected, automatically collected versions of the gazetteers especially as using these results in a cleaner methodology (less manual labor).

Similarly to the source code of the system, we publish the gazetteers under a free document license.

4 Architecture

Our system roughly follows the architecture described by [1] and [3], incorporating some ideas introduced by [6].

When building a supervised machine learning system, a major step is feature extraction, that is, collecting information from the raw data that can be relevant for the classification task. In the case of the NER task, an obvious approach to feature extraction is to collect such information from the neighbourhood of the inspected token. Some possible examples of features are capitalization, part of speech, occurrence in some dictionary. The task of the supervised machine learning algorithm is then to find in this large amount of information regularities that are relevant to the classification task.

We used maximum entropy learning as our supervised machine learning approach. During classification, the output of the maximum entropy algorithm was post-processed in a so-called smoothing phase we will describe in Subsection 4.4.

4.1 Feature extraction

Most of our features deal with very easily computable syntactic properties of tokens. On the other hand, we exploited the fact that we have the hunpos morphological disambiguator [5] at our disposal.

The feature set was composed manually. Below is the complete list of the features used by our system:

1. Is some neighbourhood of the token contained in a gazetteer? If yes, is the token at beginning, ending or middle position of the phrase? (To deal with morphology, when determining the matching of multi-word phrases, we treated the last word of the phrase differently: matching on a suitably chosen prefix was enough. This corresponds to the way Hungarian multi-word phrases are inflected.)

2. Sentence start, sentence end position.

3. Boolean valued syntactic properties of the word form: upper case, all upper case, contains capitalized letter after noncapitalized (e.g. *iPod*), is a number, contains a number, contains a dash, contains a period.

4. String-valued surface properties of the word form: the word form itself, the five-letter prefix, and most notably all consecutive three-letter character sequences (trigrams) of the word form. Note that we use gazetteers crafted beforehand, but in practice, these features have an effect similar to gazetteers directly extracted from the train corpus. Similar application of character n-grams for named entity recognition was first proposed by Klein et al. [6].

5. Information provided by the hunpos morphological disambiguator: part of speech (NOUN, ART, NUM, ADJ, VERB, etc.). The lemma of the token. Is the word form recognized by hunpos? Is the identified lemma differently capitalized than the token itself?

Example: The *Gyula* token, in a sentence starting position gets the following features: Built-in Boolean features: `sentencestart caps`. Character n-gram features: `tri.Gyu tri.yul tri.ula prefix.Gyula`. Gazetteer features: `firstname.lone city.lone familyname.lone corp.start`. (`.lone` here means that the gazetteer contains the token itself, as opposed to the case of `corp.start`, which means that the `corp` gazetteer contains a phrase starting with this token.) Morphological features: `postag.noun lemma.Gyula`.

The system collects these data for each individual token of a sentence. To incorporate context, we simply add the features of neighbouring tokens, recording the relative positions of the tokens. For example, if a token gets the feature `caps.pre2`, it means that the token two positions before is capitalized. A parameter of our system is the size of the context window for a given feature. For simplicity's sake, we didn't optimize this parameter for each feature separately. According to our experiments, in the case of character trigrams and prefixes, using context radius 3 (that is, a 7-token interval) leads to optimal results. In the case of the rest of the features, context radius 5 (11 tokens) was used.

4.2 Tag sets

The NER task in its original form deals with the classification of unknown contiguous token sequences, and it is not immediately obvious how to phrase this as a token classification task. Roughly following [3], we chose the following solution: Every token must be classified into one of 17 different classes: { `O`, `LOC.single`, `LOC.start`, `LOC.middle`, `LOC.end`, `ORG.single`, ..., `MISC.end` }. There are two major advantages of this approach: First, the machine learner can more easily recognize correlations that are specific to the start or end of the NE. Second, the tag set has implicit built-in consistency requirements: e.g. `*.start` can not follow `*.middle`. As we will see in Subsection 4.4, we can use this fact to improve the output of the machine learner in a post-processing step.

4.3 Maximum entropy method

We used the maximum entropy method as our classification methodology. This was already successfully used in the weighted morphological analyzer component of the hunpos morphological disambiguator. We also experimented with C4.5 decision trees and support vector machines, but the maximum entropy method gave consistently superior results.

We have chosen Zhang Le's [13] maximum entropy implementation because of its high performance. This implementation uses the L-BFGS algorithm [15] for model building.

On our data sets, the L-BFGS iterative learning algorithm starts to converge after approximately 100 iterations. The accuracy of the model fluctuates wildly and randomly before this iteration number is reached. Our published numbers are based on 300 iterations. On the other hand, because of the relatively high (approx. 1 hour) running time of 300 iterations, some of our elementary feature engineering decisions were based on measurements with lower (30 or 100) iterations. This may have led to suboptimal decisions.

We note that using prefixes, character trigrams and very wide context windows led to a very high total number of features. On the 200,000 token corpus, 250,000 different kinds of features occur, for a total of 10 million feature instances. The maximum entropy approach is capable of dealing with such a high number of features without the feature selection phase needed by some other machine learning methods.

4.4 Smoothing

One important characteristics of maximum entropy learning is that during classification it can emit a full probability distribution on tags, instead of just a single tag with maximum likelihood. The advantage of this is that we can override local decisions if they prove to be inconsistent with each other. The method described below is common in the machine learning literature [3]. First we query the maximum entopy algorithm for tag emission probabilities for each token. We then define transition probabilities between tags as follows: Transition probabilities for illegal transitions (e.g. `ORG.start` after `LOC.start`) are set to be zero. Every legal transition (e.g. `ORG.start` after `LOC.end`) is set to be equiprobable. Treating the maximum entropy algorithm's outputs as independent distributions, we can apply the Viterbi algorithm to calculate the tag sequence that maximizes the joint transition-emission probability for a whole sentence. This tag sequence will be necessarily well-formed. According to our measurements, this parameterless post-processing step improves the system's F-score by approximately 0.5% in typical measurement setups.

5 Measurements

5.1 Methodology

To measure the accuracy of a machine learning algorithm, its output has to be compared to a gold standard test dataset. The standard method to quantify the similarity between two named entity labelings is the CoNLL F-score. According to this, a gold standard named entity is correctly labeled if the automatic labeling gives the same start- and end-position, and the same named entity class. Based on this, precision and recall values can be calculated for the corpus, and the F-score is, as usual, the harmonic mean of these two values.

We started the early development of our system with an ad hoc train-test split of the Szeged NER Corpus. But it quickly became apparent that if we intend our results to be comparable to the only existing quantitative study on Hungarian named entity recognition, then we will have to switch to the train-development-test split used by [11]. Szarvas et al. were kind to provide this split, and from this point, we followed standard methodology: We optimized the parameters of the system guided by the F-score on the development corpus, and only measured the F-score on the test corpus once, when this optimization was finished.

5.2 Results

The system described above reached an F-score of 96.35 on the development corpus, and 95.06 on the test corpus. This is a minor improvement on the numbers published by [11] (see Table 1). But we have to note that the [11] system was optimized in parellel for English and Hungarian. Our system obviously needs further work to give state-of-the-art results for several languages.

Table 1: Results

NE-type	devel	test	Szarvas et al devel	Szarvas et al test
LOC	92.06	96.36		95.07
MISC	93.58	85.12		85.96
ORG	97.62	96.20		95.84
PER	97.44	94.94		94.67
Global	**96.35**	**95.06**	**96.20**	**94.77**

We measured the effect of each major subsets of the features. As we noted, the global accuracy score of the system was 95.06 on the test corpus. Removing just the built-in Boolean features (caps, dash, etc.) decreased this score to 92.37. Removing just the character n-gram features decreased the score to 90.04. The gazetteer and morphological features had significantly less effect: removing these decreased the score to 94.69 and 94.70, respectively. Note that these two sets of

features were exactly the ones that required external resources. Removing both led to a resourceless system without seriously affecting the score: the resourceless system had an accuracy of 94.73.

6 Acknowledgements

We thank György Szarvas and Richárd Farkas for sharing their data sets and Péter Halácsy for help in the implementation of the algorithm.

References

[1] Borthwick, A.. *A Maximum Entropy Approach to Named Entity Recognition.* PhD thesis, New York University, 1999.

[2] Borthwick,A., Sterling, J., Agichtein, E., Grishman, R.. NYU: Description of the MENE Named Entity System as Used in MUC-7. In: *Proceedings of MUC-7*, 1998.

[3] Chieu, H. L., Ng, H. T.. Named Entity Recognition with a Maximum Entropy Approach. In: *Proceedings of CoNLL-2003*, pp. 160-163. Edmonton, Canada, 2003.

[4] Csendes, D., Csirik, J., Gyimóthy, T.. The Szeged Corpus: A POS tagged and Syntactically Annotated Hungarian Natural Language Corpus. in: *Proceedings of TSD 2004*, vol. 3206 (2004) 41-49.

[5] Halácsy, P., Kornai, A., Varga, D.. Morfológiai egyértelműsítés maximum entrópia módszerrel (Morphological disambiguation with maximum entropy method). In: *Proc. 3rd Hungarian Computational Linguistics Conf.*, 2005, Szegedi Tudományegyetem.

[6] Klein, D., Smarr, J., Nguyen, H., Manning, Ch. D.. Named Entity Recognition with Character-Level Models. In: *Proceedings the Seventh Conference on Natural Language Learning*, 2003.

[7] Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., and the Annotation Group (BBN Technologies). BBN: Description of the SIFT System as Used for MUC-7. In: *Proceedings of MUC-7*, 1998.

[8] Sang, E. F. T. K., De Meulder, F.. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: *Proceedings of CoNLL-2003*, 2003. http://www.cnts.ua.ac.be/conll2003/ner/.

[9] Sekine, S., Grishman, R., Shinou, H.. A decision tree method for finding and classifying names in japanese texts. In: *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Canada, 1998, 14(4):365-393, 1999.

[10] Szarvas, Gy., Farkas, R., Felföldi, L., Kocsor, A., Csirik, J.. A highly accurate Named Entity corpus for Hungarian. In: *Proceedings of International Conference on Language Resources and Evaluation*, 2006.

[11] Szarvas, Gy., Farkas, R., Kocsor, A.. A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: *Proceedings of Discovery Science 2006*, DS2006, LNAI 4265 pp. 267-278, 2006. Springer-Verlag, 2006.

[12] Trón, V., Gyepesi, Gy., Halácsy, P., Kornai, A., Németh, L., Varga, D.. Hunmorph: open source word analysis. In: *Proceeding of the ACL 2005 Workshop on Software*, 2005.

[13] Le, Zh.. Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

[14] Zhou, G. D., Su, J.. Named Entity Recognition using an HMM-based Chunk Tagger. In: *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, pp. 473-480, July 2000.

[15] Zhu, C., Byrd, R. H., Lu, P., Nocedal, J.. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. In: *ACM Transactions on Mathematical Software (TOMS)*, 1997.

Acta Cybernetica **18** (2007) 303–314.

Similarity Based Smoothing In Language Modeling

Zoltán Szamonek* and István Biró†

Abstract

In this paper, we improve our previously proposed Similarity Based Smoothing (SBS) algorithm. The idea of the SBS is to map words or part of sentences to an Euclidean space, and approximate the language model in that space. The bottleneck of the original algorithm was to train a regularized logistic regression model, which was incapable to deal with real world data. We replace the logistic regression by regularized maximum entropy estimation and a Gaussian mixture approach to model the language in the Euclidean space, showing other possibilities to use the main idea of SBS. We show that the regularized maximum entropy model is flexible enough to handle conditional probability density estimation, thus enable parallel computation tasks with significantly decreased iteration steps. The experimental results demonstrate the success of our method, we achieve 14% improvement on a real world corpus.

Keywords: language modeling, word similarity, maximum entropy, SVD

1 Introduction

Data Sparseness in Language Modeling. A central problem in statistical language modeling is data sparseness, *i.e.* we do not have enough data to accurately estimate the probability distribution of words. Even in the context of bigrams, training an efficient model requires huge training corpus, not to mention training higher order distributions. This phenomenon is also known as curse of dimensionality. Probability density estimation thus requires some form of approximation techniques.

Smoothing. Generally, smoothing methods reserve some mass in the probability model for unseen events, and then assign that mass to these events as a function of their marginal frequencies. A great deal of techniques have been proposed for smoothing of n-gram models, including discounting, recursively backing-off to lower order n-grams, linearly interpolating n-grams of different orders. An excellent survey of these and some other techniques can be found in [6].

*Eötvös Loránd University, Pázmány P. sétány 1/c, Budapest 1117, E-mail: zszami@elte.hu

†Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende u. 13-17, Budapest 1111, E-mail: ibiro@sztaki.hu

Clustering models make use of smoothing ideas, but also attempt to make use of similarities between words. In the class-based n-gram estimation [5], a greedy algorithm searches the space of clustering to find one that increases the data likelihood the most. The method is extended to handle trigram [17], where a heuristic randomization was used to speed-up the clustering process with only a slight loss in performance.

These early works rely on short histories (such as bigram or trigram) and therefore tend to produce syntactically-oriented clusters. Another approach proposed by [2] exploits larger contexts (*i.e.* documents) between words and results in clusters that are semantically oriented. The idea is to exploit that documents can be thought of as a semantically homogeneous set of sentences. The algorithm forms a word-document matrix given the training data, performs singular-value decomposition (*SVD*) on the resulting matrix, and then clusters over the projections according to a well-chosen measure.

Similarity Information. Clustering is a special case of constructing similarity information over the words (or word-forms) of a language. [8] proposed to use a kernel-based smoothing technique where the kernel function is built using similarity information derived from word co-occurrence distributions. [16] investigated a Markov chain model that exploits *a priori* similarity information, whose stationary distribution was used for solving prepositional phrase attachment problems.

Our contribution. In our previous work [4], we presented a special case of the general Similarity Based Smoothing (SBS) method. We showed, that the similarity based methods improve language models' performance, and tested the approach on bigram estimates. Regularized logistic regression struggled with computational difficulties when estimating the parameters on real world data, although SBS performed well on test cases with lower dimensions, suggesting that the approach may introduce significant improvement to language models.

In this paper, we show that a more general and robust estimation method, the regularized maximum entropy, is able to handle real world problems as well. Our framework is also flexible enough to handle conditional probability density estimation, thus enable parallel computation tasks with significantly decreased iteration steps. The experimental results demonstrate 14% improvement on a real world corpus, which proves the efficiency of the approach.

2 Language modeling based on similarity information

The goal of language modeling is to provide an accurate estimation for $P(w|h)$, where $w \in \mathcal{V}$ is a word from the vocabulary, and $h \in \mathcal{V}^*$ is some history. Restricting the history to one word, we arrive at the well known bigram model (first order Markov model). For the sake of simplicity in this paper we will deal only with bigrams (the methods proposed can be easily extended to higher order n-grams). In what follows the notation $p(y|x)$ will be used to denote the bigram probabilities.

Given appropriate basis functions, $\{\varphi_i\}$, the log-conditional probabilities of the bigrams can be written in the form

$$\log p(y|x) \propto \sum_{i \in \mathcal{I}} \alpha_i \varphi_i(x, y). \tag{1}$$

Clearly, the simplest choice is to let $\mathcal{I} = \mathcal{V} \times \mathcal{V}$ and use the so-called Euclidean basis functions, $\varphi_i(x, y) = \varphi_{(i_1,i_2)}(x, y) = \delta(i_1, x)\delta(i_2, y)$, where $\delta(\cdot, \cdot)$ is the Kronecker's delta function. With this basis function set, fitting this model to some dataset is equivalent to maximum likelihood training. The main idea in the paper is that given some similarity information over $\mathcal{V}$ it might be possible to construct another basis function set that facilitates generalization to unseen cases. Since the basis functions that we will construct may form an overcomplete representation, we will resort to a form of regularization to prevent overfitting.

2.1 Incorporating similarity information

Let $\mathcal{V} = \{w_1, \ldots, w_{|\mathcal{V}|}\}$ be the words in the vocabulary. In order to estimate the $p(y|x)$ conditional probability distribution, where $(x, y) \in \mathcal{W}$, we used the similarity information hidden in the context. Assume that the similarity information between words is represented as an undirected weighted graph $G = (\mathcal{V}, E, W)$, where $E \subset \mathcal{V} \times \mathcal{V}$ are the edges and $W : E \to \mathbb{R}_0^+$ assigns non-negative weights to word-pairs. For mathematical convenience, we use W to denote the $|\mathcal{V}| \times |\mathcal{V}|$ weight matrix, where the $(i, j)^{\text{th}}$ entry of W is the weight of the edge $i \to j$. The idea is to construct basis functions that respect the similarity information in W. One way to accomplish this is to use spectral graph clustering which is known to yield basis functions that can be used to represent any square integrable function over G [7]. In fact, spectral graph clustering methods construct basis functions that are natural for constructing geodesically smooth global approximations of the target function. In other words, smoothness respects the edges of the graph. In our case this means that similar words (or word-pairs) will be assigned similar probabilities.

In the particular methods we have chosen, the basis functions are computed using the singular value decomposition of the matrix $P = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Here D is the diagonal *valency* matrix; its diagonal elements are defined by $d_i = \sum_j w_{ij}$. This operator is spectrally similar to the normalized graph Laplacian operator, $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$. In fact, elementary algebra yields that $I - L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = P$. The spectrum of the graph Laplacian is known to have a close relationship to global properties of the graph, such as "dimension", bottlenecks, clusters, mixing times of random walks, etc. The spectral decomposition method employed is motivated as follows: The smoothness of a function, $f : \mathcal{V} \to \mathbb{R}$ on the graph G can be measured by the Sobolev-norm $\|f\|_G = \sum_{v \in \mathcal{V}} |f(v)|^2 d_v + \sum_{(u,v) \in E} (f(u) - f(v))^2 w_{uv}$. The first term here controls the size of the function, whilst the second controls the gradient. Using this norm, the objective to exploit similarity information can be expressed as the desire to find a log-likelihood function whose G-norm is small. Now, the projection of a function f on the linear function space spanned by the top k eigenvectors of the Laplacian is the smoothest approximation to f with k basis

functions, in the sense of the G-norm. Hence, influenced by [10] we decomposed P using singular value decomposition: $P = USV^T$. For practical purposes, we truncated the SVD of P in such a way that the $\|\mathcal{P}_k\|_F = 0.9\|\mathcal{P}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm, and $\mathcal{P}_k = U_k S_k V_k^T$ is the truncated version of P where only the k column vectors of U and k row vectors of V corresponding to the k largest singular values of S are kept. For proper normalization, the singular vectors in U_k are multiplied by the square root of the respective singular values [9, 10]: $U'_k = U_k S_k^{1/2}$.

Now, the basis functions are obtained using the columns of U_k: Denoting these columns by $\varphi_1, \ldots, \varphi_k$, $\varphi_{(i_1,i_2)}(x,y) = \delta(i_1,y)\varphi_{i_2}(x)$, where $i_1, x, y \in \mathcal{V}$, $i_2 \in \{1,\ldots,k\}$. When the similarity information is unreliable we may add the Euclidean basis functions, $\varphi'_i(x,y) = \varphi_{(i_1,i_2)}(x,y) = \delta(i_1,x)\delta(i_2,y)$, to the set obtained. This way even when the automatically constructed basis functions are useless, the method still has a chance to recover. To handle unknown words, one may resort to Nyström's method [1]: In order to extend a singular vector φ_i to a new word, z, it suffices to know w_{xz} for all $x \in \mathcal{V}$ in the set of known words. In fact, $\sum_{x\in\mathcal{V}} \frac{w_{xz}}{\sqrt{d_x d_z}}\varphi_i(x)$ can be shown to be a good approximation.

In the experiments below we always used a single similarity graph, though the algorithm has the natural ability to use more graphs by simply merging the set of resulting basis functions. We may use all the basis functions, as well as all the products of two basis functions [15].

Algorithm 1 Similarity Based Smoothing (SBS)

Input: similarity information between words, training data
Output: estimated probability distribution

1. Build a similarity graph between words $G = (V, E, W)$
2. Calculate the normalized matrix of G: $P = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
3. Determine the SVD decomposition of $P = USV^T$
4. Calculate the mapping operator from the singular vectors of top singular values: $\Phi = U_k S_k^{\frac{1}{2}}$
5. Train the Estimation Model's weight parameters using the training data (Maximum entropy estimation, Logistic regression, Gauss estimation)

2.2 Logistic regression based learning

In our previous work [4], we implemented SBS using logistic regression with Laplacian and Gauss priors [14]. Logistic regression (LR) was already able to achieve good results, but unfortunately it was infeasible on real world data with large vocabularies.

Int this paper, we investigate joint maximum entropy models with regularization constraints which is a generalization of LR models, and provide better scaling properties. Our regularized maximum entropy implementation significantly outperformed the previous LR estimation in terms of computational time, which made real world experiments possible. We did the synthetic experiments for both cases, and it turned out, that using regularized maximum entropy estimation, in most of the cases yielded better model quality as well.

2.3 Maximum entropy model

Incorporating the similarity information, the basis function takes the form of $\varphi_{(i_1,i_2)}(x,y) = \delta(i_1,y)\varphi_{i_2}(x)$, substituting into (1) and rose to the exponent:

$$p_\lambda(x,y) = \frac{1}{Z_\lambda} e^{\left(\sum_{i_1,i_2} \lambda_{i_1,i_2}\delta(i_1,y)\varphi_{i_2}(x)\right)} = \frac{1}{Z_\lambda} e^{\left(\sum_{i_2} \lambda_{y,i_2}\varphi_{i_2}(x)\right)} = \frac{1}{Z_\lambda} e^{\left(\lambda_y^T \varphi(x)\right)} \quad (2)$$

where Z_λ is the normalizing factor, $\lambda = \left[\lambda_1, \ldots, \lambda_{|\mathcal{V}|}\right]^T$ is the matrix of weight parameters to be learned, λ_y contains the weight parameters of word y, and $\varphi(x)$ is the vector formed from the basis functions evaluated at x. We shall call $\varphi(x)$ the feature vector associated with x. From $p(x,y)$, it is straightforward to calculate the conditional probability $p(y|x)$.

The last term in (2) is a Gibbs distribution with parameter vector λ_y. According to [3], the maximum likelihood Gibbs distribution is equivalent to the maximum entropy distribution estimated from the training data. Assuming that the data sequence $\mathcal{D} = (v_1, \ldots, v_N)$, $v_i \in \mathcal{V}$ is generated by a first order Markov model where the transition probabilities can be modeled by (2), the data log-likelihood takes the following form:

$$L_{\mathrm{ML}}(\lambda|\mathcal{D}) = \sum_{j=2}^{N} \left[\lambda_{v_j}^T \varphi(v_{j-1}) - \log Z_\lambda\right] \quad (3)$$

The maximum likelihood (ML) estimate of the parameter vectors is the vector that maximizes this function. In order to prevent overtraining it is advisable to prefer smooth solutions, especially if the number of basis functions is big (in other words, feature vectors are high dimensional). One way to enforce smoothness is to introduce a prior distribution over the parameters λ_{ij}. Several choices are possible for such priors. In this work we studied the behavior of the method using the Laplacian-prior, $p(\lambda_{ij}) \propto \beta_{ij} \exp(-\beta_{ij}\,|\lambda_{ij}|)$. In this case, the training problem becomes the maximization of the following objective function:

$$L_{\mathrm{MAP}}(\lambda|\mathcal{D}) = \sum_{j=2}^{N} \left[\lambda_{v_j}^T \varphi(v_{j-1}) - \log Z_\lambda\right] + \sum_{i=1}^{|\mathcal{V}|} \sum_{j=1}^{k} \beta_{ij}\,|\lambda_{ij}| \quad (4)$$

i.e. an ℓ^1 penalty term is added to the ML objective function.

Applying priors to avoid overtraining is called as regularization in maximum entropy modeling. We implemented a state-of-the-art regularized maximum entropy [11] algorithm to find the solution for equation (4).

Computational speed-up. Instead of estimating $p(x, y)$ joint probability, we estimated $p(x|y)$. The reasons were two-fold, firstly in this case we were able to train these conditional models parallel, secondly because the number of parameters to be estimated reduced to k (dimension of feature vectors) from $|V| \cdot k$, the number of iterations reduced adequately in all $p(x|y)$ instance resulting in an overall decrease in running time.

2.4 Gauss based estimation

As another approach, it is obvious to try some simple models, when the words are mapped to an Euclidean space. One such model can simply fit a Gaussian mixture model on the training data, and use it to estimate (maybe only the missing) statistics.

$$P(y|x) = \frac{1}{Z_x} \sum_{x' \in \mathcal{V}} P'(y|x') \cdot e^{\left(-\frac{||\varphi(x') - \varphi(x)||}{\sigma^2}\right)}$$

We used the maximum likelihood estimate for P', and Z_x was a normalization factor such that $\sum_y P(y|x) = 1$. If the distance of words in the Euclidean space reflects the similarity between words (as in our case), this may lead to a good estimate.

3 Experimental Results

We compared the performance of our approach, SBS, on synthetic and real-world data to interpolated Kneser-Ney smoothing (IKN). IKN is considered as one of the best smoothing techniques [13]. As special cases of SBS, we present the results for the maximum entropy model (Maxent), the logistic regression estimation (Logreg) and the Gauss mixture approach (only Maxent is calculated for real world data).

In order to evaluate the methods we calculated the empirical cross-entropy of the trained model on held-out data, $w_1, \ldots, w_N$:

$$H_N(p, \hat{p}) = -\frac{1}{N} \sum_{i=1}^{N} \log_2 \hat{p}\,(w_i|w_{i-1})\,. \tag{5}$$

Here $\hat{p}(w_i|w_{i-1})$ is the probability assigned to the transition from w_{i-1} to w_i by the model and p denotes the true distribution. (For the synthetic datasets we have $w_i \sim p(\cdot|w_{i-1})$.) Since by assumption, the held-out data has the same distribution as the training data, by the Shannon-McMillan-Breiman theorem we know that (under mild assumptions) H_N is close to the cross-entropy of $\hat{p}$ with respect to the true distribution p of the data.

For each test, we calculated the cross entropies for the IKN estimate and the proposed similarity based smoothing techniques. In the case of synthetic datasets

we also estimated the entropy of the models by using the true transition probabilities in (5), the corresponding points on the graphs are labeled as 'Model'.

3.1 Tests on Synthetic Data

The model generating the synthetic data. In order to develop a good understanding of the possible advantages and limitations of the proposed method, we tested it in a controlled environment where data was generated using some designated bigram model. In order to make the test interesting, we decided to use "clustered" Markov models of the following form: the probability of observing y given x is computed by

$$P(y|x) = P(\, y \,|\, c(y)\,)\, P(\, c(y) \,|\, c(x)\,), \tag{6}$$

where $c(x)$ is the cluster of symbol (word) x ($c : \mathcal{V} \to \mathcal{C}$ with $|\mathcal{C}| < |\mathcal{V}|$). The idea is that the next observation depends on the past observation x only through its class, $c(x)$ and hence two past observations, x and x' yield to identical future distributions whenever $c(x) = c(x')$. Note that when $P(y|c) = \delta(c(y), c)$ then the model can be thought of as a Hidden Markov Model (HMM) with state transitions defined over $\mathcal{C}$, whilst when the observation probabilities are unconstrained then in general no HMM with $|\mathcal{C}|$ *states* is equivalent to this model. In any way, the model can be specified by two matrices, (A, B), with $A_{c_1,c_2} = P(c_2|c_1)$ ($A \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$) and $B_{y,c} = P(y|c)$ ($B \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{C}|}$).

For computing the matrix A, we start with a permutation matrix of size $|\mathcal{C}| \times |\mathcal{C}|$ and perturb it with random noise so that *(i)* all transition probabilities are nonzero and *(ii)* for each state the number of "significant" transitions lies between 1 and $\frac{|\mathcal{C}|}{4}$.

For computing B, we start from the idealized block-structured matrix with $B'_{yc} \propto \delta(c(y), c)$ and then perturb it according to

$$B_{yc} \propto B'_{yc} + \delta(1 + \gamma z'_{yc}),$$

where the elements of z'_{yc} are independent random variables uniformly distributed in $[-0.5, 0.5]$. If $\delta = 0$ then the block structure of the source is clearly identifiable based on the outputs: Given an observation y and knowing $\mathcal{C}$ we can infer with probability one that the hidden state is $c(y)$ and as noted before the model collapses to a hidden Markov model with $\mathcal{C}$ states. When δ is non-zero this structure is blurred. In the experiments we used $\delta = 0.1$ whilst we let γ change in $[0, 1]$. Note that $\gamma = 1$ roughly corresponds to a noise level of 5% in the observations.

Similarity information provided for SBS. We controlled how well the similarity information provided for SBS reflects the actual block structure of the data. The perfect similarity information assigns 0 to observations (words) in different clusters, whilst it assigns the same positive value, say 1, to observations in the same cluster. The corresponding matrix is denoted by S ($S \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$): $S_{xy} = \delta(c(x), c(y))$.

We then disturbed S by adding noise to it. For this a random $|\mathcal{V}| \times |\mathcal{V}|$ matrix (Z) was created whose entries were independent, uniformly distributed random variables taking values in $[0, 1]$. Given a noise parameter, $\epsilon \geq 0$, the perturbed matrix is computed by

$$S_\epsilon = S + \epsilon\,((-S \odot Z) + ((\mathbf{1} - S) \odot Z)).$$

Here $U \odot V$ denotes the component wise product of matrices U and V, $\mathbf{1}$ denotes the matrix with all of its entries being 1. In words, increasing ϵ reduces the inter-cluster similarity and increases between-cluster similarity. In the extreme case when $\epsilon = 1$ all similarity information is lost.

Training and test datasets. A training dataset normally contains $N_{\text{tr}} = 300$ observation sequences (except when we experiment by changing this parameter), each having a random length that was generated by drawing a random number L from a normal distribution with mean 11 and variance 6 and then setting the length to $\max(2, L)$. The test dataset (separate from the training dataset) had 5000 sequences which was generated using the same procedure. All measurements were repeated 100 times and the average values are presented.

3.2 Results

We performed a sensitivity analysis to test the effect of how the various parameters influence the results. In particular, we studied the performance of SBS as a function of the observation noise, γ, that masks the identifiability of the block structure, the noise in the similarity matrix (ϵ) that gradually decreases the quality of the similarity information available for SBS, the number of training sentences (N_{tr}) and the cluster structure. These parameters were changed one by one, whilst the others are kept at their default values which were $\gamma = 0.2$, $\epsilon = 0.1$, $N_{\text{tr}} = 300$. The default cluster structure was to have 6 clusters, each having $30, 20, 10, 5, 5$ and 5 words respectively (so that some clusters were bigger, some were smaller). Thus $|\mathcal{V}|$, was kept at 75.

Sensitivity to noise masking the block-structure. When $\gamma = 0$, the observations can be used directly to infer the underlying classes and the estimation problem is easier. When the block-structure masking noise is increased the problem becomes harder.

Figure 1 shows the results of the measurements. It can be observed that the proposed methods perform significantly better over the considered spectrum of γ than IKN. On the other hand, SBS Logreg and SBS Maxent were able to maintain its estimation accuracy for the whole range of γ, while the Gauss estimate had a slight loss of quality.

Robustness against the degradation of the similarity information. In the next set of experiments we investigated the sensitivity of the methods to the

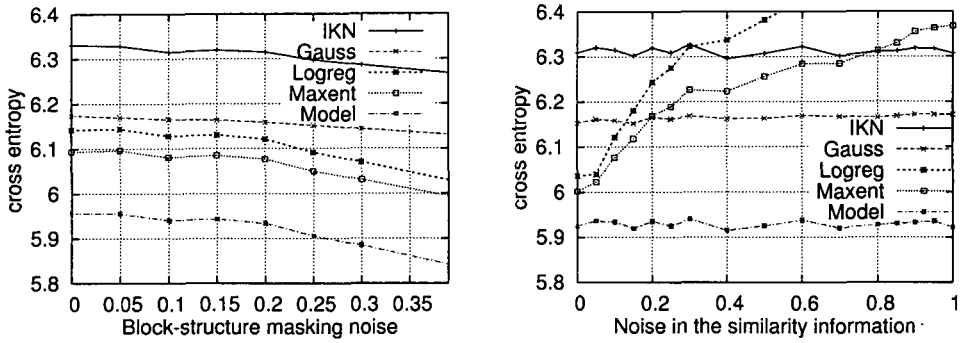


Figure 1: Left: The cross-entropy of models built with various algorithms as a function of the block-structure masking noise parameter (γ). Right: Cross-entropy of the noise parameter, ϵ, governing the quality of the similarity information.

quality of the similarity information. For this we gradually increased the similarity-information noise parameter, ϵ, from 0 to 1. As we have discussed, when $\epsilon = 0$ the similarity information is perfect, whilst when $\epsilon = 1$ all similarity information is lost.

As expected, when the similarity information gets weaker, the performance of SBS methods degrade and converge to that of the ML-estimated bigram model (cross entropy 9.8 due to lack of all bigrams in training data). It is notable that even when $\epsilon = 0.7$ (when the similarity information is already very poor) SBS Maxent performs as well as IKN. The reason is that although in these experiments we did not add the Euclidean basis functions to the set of basis functions, we can expect the spectrum of a high-noise similar matrix to be uniform, hence covering 90% of the spectrum will add almost all singular vectors to the basis and thus the model automatically retains its power. This effect may be the reason for the Gauss estimation, which seems to maintain it's performance during this experiment. Namely, as the noise gets bigger, the more features will be kept, and the Gauss estimate seems to handle this very well (the training of the Gauss estimate does not depend on the number of features as long as it contains the necessary information, while LR and Maxent had a linear growth in the number of parameters).

Investigation of the $\epsilon = 1$ case: Gauss estimation seems to result in a nearly unigram distribution: $P(y|x) = P_{Gauss}(y)$, which seems to be a very good model for this data. Due to the small corpus size (about 3000 bigrams for 5625 word pairs) and lack of similarity information, other models (including IKN) were trying to capture the small differences in counts, while Gauss tried to fit normal distributions randomly over the space resulting in nearly unigram distribution. However, we must note, that there is still a significant difference between the original model and the estimations.

Sparsity of training data. The main promise of SBS is that by using the similarity information it is capable of achieving better performance even when the available training data is limited. In order to validate this, we performed a set of experiments when the number of sentences in the training data was varied. The results are shown in Figure 2. All SBS models were able to outperform IKN for a wide range of values of N_{tr}, although for very small N_{tr} IKN seems to be better than Maxent and Logreg. We see, that Gauss performs very well at the beginning, but it does not converge quickly to the theoretical optimum 'Model'.

Cluster structure. We tested SBS with large variety of cluster setups, ranging from 1 to 15 clusters, and with vocabulary sizes between 7 and 110. The results are summarized on Figure 2.

It is clear that if the number of clusters is small or when all the words are in different clusters, then there is no significant structural information in the similarity. It is notable that in such cases SBS Logreg was still able to perform as well as IKN. On the other hand, if there is a significant hidden structure in the model, all SBS methods greatly outperform IKN. In this experiment, $H(p, \hat{p}_{\mathrm{ikn}})$ ranges 2.5 to 6.5.

Gauss seems to be able to perform well only with large vocabularies. Maybe this is due to the observation made at sparsity of training data experiments, that it does not really converge to the optimal model when more data is available.

clusters $\hat{p}_* =$	$H(p,\hat{p}_{\mathrm{ikn}}) - H(p,\hat{p}_*)$		
	$\hat{p}_{\mathrm{Gauss}}$	$\hat{p}_{\mathrm{Logreg}}$	$\hat{p}_{\mathrm{Maxent}}$
1,1,1,1,1,1,1	-0.126	0.000	-1.933
10	-1.713	0.017	0.017
10x2	-0.044	0.089	0.073
10x3	-0.016	0.178	0.160
10x5	0.097	0.254	0.266
10,5	-0.050	0.044	0.015
10,7,5	-0.100	0.102	0.087
10,7,5,3	-0.012	0.128	0.128
30,20,10	0.161	0.284	0.309
30,20,10,5x3	0.156	0.194	0.238
30,20,10,5x3,1x3	0.161	0.151	0.219
30,20,10,5x3,1x6	0.169	0.113	0.209
30,20,10,5x3,1x9	0.166	0.073	0.194

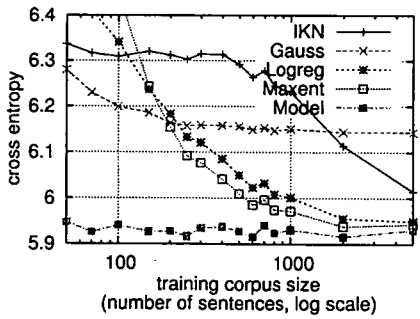


Figure 2: Left: The cross-entropy decrease of SBS as compared with IKN for a number of different cluster structures. Right: The cross-entropy of models built with various algorithms as a function of the number of sentences in the training data (N_{tr}).

3.3 Tests on Real Data

The SBS Maxent model was built using 55,000 sentences from the Wall Street Journal (WSJ) corpus. We tested the model on the Brown corpus, which contained 47,000 sentences.

We reduced the feature space by using only the most relevant features from Φ and we did not add the Euclidean basis functions to the features. With these constraints, the model is not in its most general form, but this relaxation reduced the computational requirements significantly. To compensate the lack of information, we use IKN to estimate high frequency bigrams, and leave the low frequency case ($C(x) < 2$ when estimating $P(y|x)$) for SBS where IKN doesn't perform well.

The similarity graph was constructed according to the number of word occurrences in the different senses of each word (WordNet [12] overview property). Weights for the similarity graph were constructed according to weight information in the overview.

Combined vocabulary size from WSJ and Brown contained nearly 70,000 words. We reduced this to 18,000 words by merging the infrequent ones, which were mainly numerical values and named entities.

The cross entropy using IKN trained on WSJ and tested on Brown corpus was 9.47, while using our approach for the sparse part of the IKN estimate, the cross entropy was only 9.27. This is a 14% improvement in terms of Perplexity.

This result shows, that using general similarity graph from WordNet we were able to improve the estimation even on a different corpus.

4 Conclusions

The improved Similarity Based Smoothing algorithm is now capable to work on real world data. We have found that regularized maximum entropy models provide a more flexible way of learning the parameters of the probability distribution to be estimated. We have also investigated that a simple Gaussian mixture model over the words might perform well, if relevant similarity information is present in the underlying data. We achieved 14% perplexity improvement over Interpolated Kneser-Ney smoothing on real world corpus using the maximum entropy SBS method.

Using WordNet, we created a similarity graph over words. This graph is an external information source, we built it independently from the training corpus, thus we expect this graph to be domain independent. Real world experiments showed that maximum entropy version of SBS was able to utilize this large graph and improved the bigram estimation.

The current work shows that the expectations with SBS are true. It is able to handle large dimension real world problems as well as synthetic and low dimension problems (*e.g.* POS tag bigrams).

We also showed that this kind of word representation allows us using robust mathematical models (Logistic Regression, Maxent, Gaussian Mixtures) to conserve the topology (*i.e.* distance of points in the Euclidean space maps well to distance of words in similarity), yielding promising results.

References

[1] Baker, C. T. H. *The Numerical Treatment of Integral Equations.* Clarendon Press, Oxford, 1977.

[2] Bellegarda, Butzberger, Chow, Zen-Lu, Coccardo, and Naik. A novel word clustering algorithm based on latent semantic analysis. In *ICASSP*, 1996.

[3] Berger, A. L., Pietra, S. A. Della, and Pietra, V. J. Della. A maximum entropy approach to natural language processing. In *Computational Linguistics*, 1996.

[4] Bíró, István, Szamonek, Zoltán, and Szepesvári, Csaba. Sequence prediction exploiting similary information. In *IJCAI*, 2007.

[5] Brown, P., de Souza, P., Mercer, R., Pietra, V.J, and Lai, J. Class based n-gram models of natural language. In *Computational linguistics*, 1992.

[6] Chen, Stanley F. and Goodman, Joshua. An empirical study of smoothing techniques for language modeling. In *ACL*, 1996.

[7] Chung, Fan. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics.* American Mathematical Society, 1997.

[8] Dagan, Ido, Lee, Lillian, and Pereira, Fernando C. N. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34:43, 1999.

[9] Dhillon, Inderjit S. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

[10] Ding, He, Zha, and Simon. Unsupervised learning: Self-aggregation in scaled principal component space. *LNCS*, 2002.

[11] Dudik, Phillips, and Schapire. Performance guarantees for regularized maximum entropy density estimation. In *COLT*, 2004.

[12] Fellbaum, Christaine, editor. *WordNet: An Electronic Lexical Database.* The MIT Press, Cambridge, Massachusetts, 1989.

[13] Goodman, J. A bit of progress in language modeling. Technical report, Microsoft Research, 2001.

[14] Krishnapuram, Carin, Figueiredo, and Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE*, 2005.

[15] Lin, Y. Tensor product space anova models.

[16] Toutanova, Manning, and Ng. Learning random walk models for inducing word dependency distributions. In *ICML*, 2004.

[17] Ueberla, J.P. An extended clustering algorithm for statistical language models. Technical report, Simon Fraser University, 1994.

Acta Cybernetica 18 (2007) 315–328.

Hungarian WordNet and representation of verbal event structure*

Judit Kuti‡ Károly Varasdi‡ Ágnes Gyarmati† and Péter Vajda†

Abstract

During the development of the Hungarian WordNet we found that the primarily hierarchical structure of the Princeton WordNet (PWN) had to be expanded in order that it could be used for a more accurate representation of relations among verbal meanings in Hungarian, as well. Treating verbs as eventualities we made some aspectual information explicitly available in the representation of verbs. Using the notion of *nucleus* introduced by Moens&Steedman we added new relations to the WordNet and classified verbal synsets according to aspectual characteristics. This enabled the representation of some psycholinguistically relevant pieces of information and a wider possible usage of the thus extended Hungarian verbal WordNet in the field of computational linguistics.

Keywords: WordNet, verb, event structure, event ontology, aspect

1 Introduction

In the present study we examine some specific problems related to Hungarian verbs which we have encountered when developing the Hungarian WordNet (HuWN), and show how our results relate to the Princeton WordNet, the standard database we relied on when building the HuWN. As wordnets[1] were originally designed to describe the hierarchical structure of nouns, it is nouns that constitute a preponderant

*The building of the Hungarian WordNet was carried out in the framework of the project "Building of the Hungarian WordNet" (GVOP – 2004 – 3.1.1.) since 2005. The project was a collaboration between the University of Szeged, MorphoLogic Ltd. and the Research Institute for Linguistics of the Hungarian Academy of Sciences. The Research Institute for Linguistics was involved in building the verbal part of the WordNet. It is here we would like to thank for the support of the project. As the present article largely relies on the common work carried out in cooperation with our colleagues Judit Cziczelszki, Anikó Nagy and Marianna Tóth, we would like to thank them for their help and contribution.

†Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest, E-mail: {kutij,varasdi,aagnes,vajda}@nytud.hu

[1] When talking about a specific WordNet of a given language, we refer to it with the widespread, trademark-like spelling, using capital *'W'* and *'N'*, while when referring to the database type as to a common noun we use minuscules.

part of existing wordnets and it is nominal relations that, to a large extent, served as examples for verbal relations, as well.[2] However, the choice of the two distinct names for equivalent relation types in the two respective parts of speech in PWN indicates already that a meaning representation framework for verbs cannot be solely designed on the basis of the existing grounds for a nominal hierarchy, not even in the case of a language like English, in which verbs as lexical units bear little or no information related to aspect or aktionsart.[3] In the case of languages in which this information is stored in the verb, e.g. thruough preverbs, the inapplicability of the nominal structure to the verbal network is even more obvious, when attempting to develop a lexical semantic network.

Examining the event structure of verbs provides help in approaching these questions. Accordingly, in what follows, we would like to show what ways of representing certain pieces of information that stem from the event structure of verbs and determine their semantic relations we have worked out – within the framework facilitated by wordnet as a genre.[4] We present some fundamental statements on event structure and aspectuality in general, on aspect and aktionsart of verbs with special respect to Hungarian, and an elementary event-structure called *nucleus* introduced by Moens&Steedman. The remaining parts of our study show that by using the notion of *nucleus* we acquire a means that enables us to

1. incorporate lexicalised meanings into WordNet more easily than was possible previously
2. represent psycholinguistically relevant information that were so far missing from the Hungarian WordNet
3. store information that prove to be useful for computational linguistic applications of the HuWN.

2 Eventualities and their aspectual properties

2.1 Logical implication between verbal meanings

It is necessary to examine in what way the relation of logical implication holds between verbs[5] since this is what both the relations *troponymy* and *hyponymy* are based on. Implications of a sentence are highly dependent on its aspect, illustrated by the following examples:

[2]Although the main relation used in the structuring of verbal synsets in PWN, *troponymy*, as introduced by Fellbaum (see [3]), is, in principle, different from the hypo-hypernymy relation used in the nominal part of the WordNet, the hierarchical relational structure adopted from the nominal part suggests nonetheless enough similarity between the two relations to call the equivalent verbal relation expressing super- and subordination in BalkaNet (see [8]) *hypo-hypernymy*.

[3]In the definition of these two terms we largely rely on [5].

[4]The slightly theoretical linguistically oriented nature of the paper is intended to be balanced by the usefulness of WordNet for computational linguistic applications.

[5]When talking about logical implications and aspectual properties of verbs, we should, in fact, be talking about verbal phrases, since verbs on their own are underspecified with respect to this kind of information, see [10].

1. *Mari éppen ment át az utca túloldalára, amikor megpillantotta Jánost.*
   'Mary was crossing the street when she saw John.'

2. *Mari átment az utca túloldalára, amikor megpillantotta Jánost.*
   'Mary crossed the street when she saw John.'

While sentence (1) does not imply that Mary actually crossed the street – she might have turned back to greet John, sentence (2) does imply that Mary did finish crossing the street (moreover, the pragmatical implicature suggesting that Mary crossed the street because she had noticed John, is also present).

The difference between the two main clauses in Hungarian is merely aspectual: the first is in progressive, while the second is in perfective aspect, each possessing a different logical potential.[6] It is, thus, obvious that the question concerning what implications the preverb and verb as a whole can take part in is not separable from its aspectual value in the sentence. Although in Hungarian the actual aspect of a sentence[7] is of course determined by many factors in the sentence besides the verb, its aspectual *potential* — as well as the sentences it can imply — is largely determined by the event structure of the verb.

In Hungarian some preverbs can bear information related to both aspect and aktionsart. This alone might make Hungarian seem to be similar to Slavic languages. However, on the one hand, Hungarian does not express aspect in as a predictable manner, as e.g. Russian, whose WordNet we could have used as a basis for the Hungarian one, if the two languages had had enough similarities. On the other hand, aspect and aktionsart in Hungarian are interwoven in a way that is unique among the languages that so far have been developed a wordnet for. The perfective aspect for example goes almost always hand in hand with one of the aktionsart-types that are present in Hungarian (Kiefer 2006:45, see [5]).

Furthermore, Hungarian has an extremely rich system of preverbs which can modify the meaning of the stem, making it inevitable, when dealing with Hungarian, to consider aspectual characteristics as much as possible within the given framework. As already mentioned, the basic verbal relation, hypo- and hypernymy, but troponymy just as well, were elaborated based on the pattern of nominal relations in the sense that the wordnet-methodology requires that semantic relations between morphemes hold through logical implications. While in the case of nouns one can show that N1 is a hyponym of N2 — by checking whether the pattern "it is true for each X that if X is an N1, then X is an N2" holds —, this is not possible for verbs, since one can only establish logical relations between propositions or the sentences expressing them, but the logical structure of sentences is determined by verbs together with their modifiers and complements. However, the verb-complement relation is highly asymmetrical: the logical potential of the sentence is determined by the verb; complements are only more or less passive participants.[8] As the PWN, which has served as a basis for HuWN, does not contain aspectual information

[6]The above phenomenon is known as the *imperfective paradox*.

[7]Aspect itself is considered a sentence semantic category (see [5]).

[8]In the case of verbs with direct object complements it is also the direct object that takes part in determining the aspect of the sentence. However, the impact of the direct object on the aspect

due to the lack of morphological marking of aspect in English, another way had to be found for representing typically occurring phenomena related to aspect in Hungarian.

A first framework of approaching aspect in general is provided by Zeno Vendler [9], who developed a type of event ontology in a way that would become useful for linguistic theory. This system was later elaborated by Emmon Bach [1], and worked out for computational linguistics by Marc Moens and Mark Steedman [7]. Drawing on Moens&Steedman's work we would like to suggest a way to structure aspectually related verb meanings in WordNet.

2.2 Aspectual classes according to Vendler and Bach

Vendler's classification of eventualities distinguishes between four aspectual classes according to the internal temporal structure of the event expressed by the verb. According to Vendler the four event types — with arguments and with context — differ in the aspects they may take: activities (e.g. *swim*) typically take the progressive aspect, accomplishments (*go out of the room*) take both the progressive and the perfective aspect, and achievements (*blow up*) take the perfective aspect. States take neither the progressive nor the perfective aspect. The classification as further developed and extended by Bach represents aspectual categories in a binary system, highlighting the existence of point expressions that are different from achievements (e.g. *click*). In Bach's terminology Vendler's accomplishments are called *protracted events*, achievements are called *culminations*, while *point* expressions are called *happenings*.

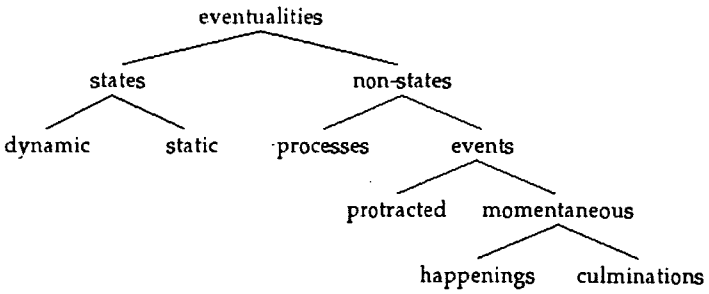


Figure 1: Classification of eventualities according to Bach

Vendler's four aspectual classes are also characterised by whether the interval of the event is divisible or not — i.e. whether the eventuality denoted by the verb holds for most of the sub-intervals, as well. Accordingly, *activities* and *states* may be considered homogeneous eventualities,[9] since they are expressed by predicates

can be relatively well predicted from the event structure of the verb and the properties of the object, so we do not have to specifically deal with this in the framework of the WordNet.

[9]We are using the term *eventuality* after Bach, see [1].

any sub-intervals of which may be described by the very same predicates. *Accomplishments* and *achievements* on the other hand are in this respect coherent units of different kinds of heterogeneous event-components. *Point* expressions are also taken to be non-complex eventualities. From the point of view of constructing the Hungarian verbal WordNet it is the representation of complex eventualities — *achievements* and *accomplishments* — in a way that does justice to their aspectual properties that is of interest to us. Their complexity might be interpreted with the help of the so-called *nucleus-structure* introduced by Moens&Steedman.

2.3 The event-nucleus of Moens&Steedman

Moens&Steedman introduce a classification of eventualities relying on Vendler's aspectual classes but further refining it. Their central notion is that of an event-nucleus, which might be called a tripartite structure or triad, as well. The reason for the latter name is that an idealised eventuality consists of potentially three components belonging together: *preparatory phase*, *telos/culmination* and *consequent state*.

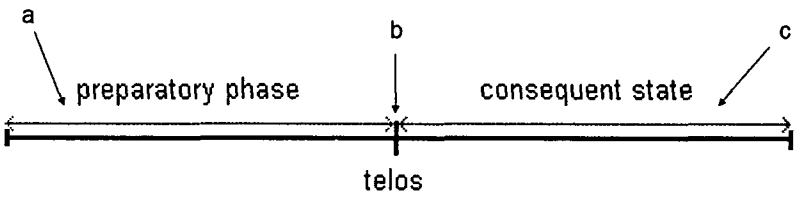


Figure 2: The event-nucleus of Moens&Steedman

One may also represent the triad as an ordered triple $< a, b, c >$ where a=preparatory phase, b=telos and c=consequent state. Moens&Steedman place this idealised event-unit beyond the level of linguistically manifested lexicalised meanings. The components of the event-nucleus are thus filled with meta-linguistic and not with lexicalised linguistic elements.[10]

meta-language level	$< a, b, c >$
linguistic level	lexicalized linguistic units (verbs)

Figure 3: The event-nucleus on meta-language and linguistic levels

Treating the three nucleus-components[11] as a unit might be justified as follows. Since we are examining eventualities from an aspectual point of view, the fact that when testing a lexicalised expression with linguistic tests sensitive to aspectual

[10]Since we may only refer to these with linguistic elements, we will use small capitals so that they can be held apart from italicised, linguistic elements.

[11]Here we are dealing with the event-components irrespective of whether they are lexicalised or not.

properties (in Hungarian the tests of the progressive and the perfective) the co-occurance of no more than the three components outlined above may be shown, must be considered relevant. We may, thus, acquire information about the aspectual properties of a verb expressing a certain eventuality by looking at which of the three event-components described above are conceptually present.

On the example of the eventuality lexicalised with the verbal phrase *go out of the room*: The existense of the first component can be tested by looking at whether the expression can be put into the progressive. An expression will be acceptable in the progressive if and only if the first component of its triad is conceptually present. The existence of the third component, which practically goes hand in hand with the presence of the second one,[12] can be tested by looking at whether the expression can be put into the perfective (see [7]). Due to certain characteristics of the Hungarian language the easiest way we can test whether certain components of the triad are conceptualised is by translating the Hungarian sentence into English and putting the translated equivalent into Present Perfect / Progressive.[13]

3. *János éppen ment ki az épületből, amikor találkoztam vele.*
   'János was going out of the building when I met him.'

4. *Mire Zsuzsa megérkezett, addigra János kiment az épületből.*
   By the time Sue arrived, John has gone out of the building.

As a result of the two tests we can see that the phrase *go out of the building* conceptualises all the three components of the triad:

<GOES TOWARD THE GATE, PASSES THE THRESHOLD, IS OUTSIDE>

Moens&Steedman elaborate the categories established by Vendler/Bach, by adding the factors of the presence or lack of the triad-components. In order to see how the classification according to the triad-components relates to the classification of Vendler/Bach, let us look at Table 1. This shows the classification of eventualities according to the factors taken into consideration by Moens&Steedman (+/− atomic and +/− consequent state), explicitly referring to the equivalents in Vendler's and Bach's system, where possible (in cases where the new terminology differs from the former one, we have indicated the latter in brackets).

[12]Although the second compontent may sometimes be called culmination point, this is not supposed to imply that this event-component necessarily has to take place within a pointlike short time. This can indeed be a longer period which is conceptualised as a point.

[13]Since this methodology may be surprising at first, some explanation is in order. In Hungarian — as opposed to English — there are no clear-cut and simple tests that are sensitive enough to the aspectual properties of a sentence (or verb phrase). Realizing that the impossibility of providing a usable test battery for Hungarian, we chose a detour, as it were, through a proxy in English. Benifiting from the situation that everybody in the WordNet developers' team spoke English on an advanced level and had learnt to be sensitive to certain aspectual features in English, we decided to rely on our tacit knowledge of the aspectual features we wanted to test. When translating a Hungarian sentence into the English Present Perfect or Progressive, one had to judge its aspectual acceptability irrespective of whether the translation was correct in any other respect. Obviously, this methodological shortcut should be backed by further research in second language acquisition to be of sound theoretical value, but we believe that used with sufficient care it provides a reliable tool when the tests in the object language prove too complicated for practical usage.

Table 1: Eventualities in the system of Moens&Steedman

	non-states		**states**
	atomic	extended	
+conseq	**culmination** (=ACHIEVEMENT) recognize, spot win the race	**culminated process** (=ACCOMPLISHMENT) build a house eat a sandwich	resemble understand love know
−conseq	**point** hiccup tap, wink	**process** run, swim, walk play the piano	

Theoretically 2^3 different potential aspectual types may be distinguished according to the conceptual presence of the nucleus-components, listed as follows.[14]

$< \oslash, \oslash, \oslash >$ $< \oslash, b, c >$ $< a, \oslash, \oslash >$
$< a, b, c >$ $< a, \oslash, c >$ $< \oslash, b, \oslash >$
$< a, b, \oslash >$ $< \oslash, \oslash, c >$

The coherence of the nucleus components is more than mere temporal sequentiality, it is what Moens&Steedman call *contingency* — "a term related, but not identical to a notion like causality" [7]. The mutual dependency among the three components of the nucleus means that none of them can be seen as *preparatory phase*, *culmination* or *consequent state* per se. An eventuality that, based on the above tests, seems to possess a preparatory phase, but lacks both culmination and consequent state (could be marked as $< a, \oslash, \oslash >$) cannot be seen as a preparatory process, as it does not precede anything. By analogy, an eventuality that, based on the above tests, seems to possess a consequent state but lacks a *culmination* (could be marked as $< \oslash, \oslash, c >$) cannot be seen as a consequent state, just like an eventuality with what seems to be a point of culmination, but lacking both preparatory phase and consequent state (could be marked as $< \oslash, b, \oslash >$) cannot be interpreted as a telos. In other words, a triad having a consequent state implies that the triad also has a culmination point. However, the three respective components seemingly appearing on their own may easily be interpreted as corresponding to the notion *process* and *state* as used by Vendler and to the Bachian *point* expression.

Although the three non-complex eventualities (process, point, state) are not discussed further by Moens&Steedman, we deal with them in HuWN, and follow the above convention of showing the aspectual information in an ordered triple. Accordingly, the above listed possible combinations of the nucleus-components, each standing for one possible aspectual verb-subtype, are illustrated with examples, as follows:

[14]The sign $\oslash$ refers to non-conceptualised components of the triad.

$< \varnothing, \varnothing, \varnothing >$	no example
$< a, b, c >$	*befelhősödik*
$< a, \varnothing, c >$	no example
$< \varnothing, b, c >$	*eltörik*
$< a, b, \varnothing >$	no example
$< a, \varnothing, \varnothing >$	*fut*
$< \varnothing, b, \varnothing >$	*kattan*
$< \varnothing, \varnothing, c >$	*szeret*

Three of the possible combinations are excluded based on epistemologic grounds: (i) A nucleus having no components at all cannot be discussed neither conceptually nor linguistically. An eventuality (ii) having a preparatory phase and a culmination point, as well as one (iii) having a preparatory phase and a consequent state cannot be lexicalised due to the coherence of the telos and the consequent state.

Besides the remaining five lexicalised possibilities of nucleus-component combinations we have, however, seen the need for marking a sixth possible aspectual type in HuWN. As mentioned above, in many cases linguistic tests in Hungarian are unreliable in the sense that they provide ambiguous results even for native speakers. For the sake of usability in Hungarian language technology applications we considered it necessary to explicitely mark those cases in HuWN where the Hungarian test for the progressive did not result in a clearly grammatical sentence, but the English equivalent did. One such example can be seen in (5):

5. *János éppen gyógyult meg, amikor huzatot kapott a füle és újra belázasodott.*
   John was getting better when his ear caught cold and he got fever again.

In cases like the above mentioned we decided to mark the first component of the nucleus "unmarked", designating this with an x: <x,b,c>

3 The notion of the nucleus in HuWN

As we have seen, the conceptual presence or absence of meta-language elements beyond the lexicalized expressions can be tested with the help of Moens & Steedman's nucleus structure. The number of components a verb conceptualizes compared to an idealized complex event unit provides information on the telicity or atelicity of a given eventuality. If the third component of a nucleus denoted by a given verb is expressed,[15] the eventuality is telic, if this component is not present, the eventuality is atelic.

3.1 Representing telicity in HuWN

From the six mentioned possible patterns whose lexicalisation the presence of the respective nucleus-components enables it is only complex eventualities that can be

[15] As mentioned in the previous section, the presence of the third component entails the presence of the second component.

telic. If we would like to get an overview of these complex eventualities from an aspectual point of view, the representation in ordered triples as introduced in 2.3, seems appropriate, as it can be seen in Table 2:

Table 2: Telicity of complex eventualities illustrated by the tripartite event structure of Moens&Steedman

Components of the triad	The metalinguistic name for the conceptualised components of the phrase lexicalising the triad	Telicity of the VP
<a, b, c>	*kimegy (to exit)*: <TO GO TOWARDS THE DOOR, TO STEP THROUGH THE DOOR, BEING OUTSIDE>	+consequent state→telic
< ∅, b, c>	*felrobban (blow up)*: < ∅, THE MOMENT OF BLOW-UP, BLOWN-UP STATE>	+consequent state→telic
<x, b, c>	*meggyógyul (get better)*: <X, CHANGING STATE FROM SICK TO HEALTHY, HEALTHY STATE>	+consequent state→telic

Of the simple eventualities, *processes* and *states* are usually considered atelic while *point* expressions (on their own, without context) are underspecified for this kind of information. When constructing a wordnet the question arises whether and how to represent meanings that should be synonyms according to the notion of synonymy in wordnet and yet differ aspectually. The notion of the nucleus helps us answer: aspectual differences can and should be represented in HuWN. If a meaning represented as a synset in the wordnet is transformed into a minimal proposition, one can determine whether the consequent state of the appropriate nucleus is present.[16] By encoding whether a meaning has a consequent state (and hence a telos), through assigning to it one of the six conceptualization patterns of the triad components, the telicity of the eventuality expressed by the verb will be made explicit. This information is stored in HuWN in a similar way as in the case of the information on verb frames: we indicate which of the three triad components is conceptualised in Hungarian on the level of the literals.

As already introduced, for the sake of uniformity and transparency we follow the convention of showing the aspectual information in an ordered triple even in the case of simple eventualities mentioned in 2.2 and 2.3. Accordingly, the ordered triple of the verb *fut* 'run' is $(< a, \emptyset, \emptyset >)$. This triple shows on the one hand that the eventuality expressed by the verb *fut* is atelic, and on the other that it is

[16]Transforming verbal meanings into minimal propositions is ensured in the WordNet by mapping all the possible verbal subcategorisation frames of a given literal onto its synset. Sometimes several verb frames are merged into one verb frame record with optional arguments. In this case verbs should be considered with the minimal number of obligatory arguments. E.g. the verb frame *eszik*, 'eat' contains an optional direct object, so the minimal predicate should be formed without an object, and that predicate is atelic.

a Vendlerian process, indicated by the preparatory phase being solely present.

3.2 Complex eventualities in HuWN

Besides the possibility of storing a minimal amount of aspectual information concerning the given literal in a verb synset, the relational structure of the wordnet and the nucleus taken as a single unit allow us to propose another extension to the verb synset structure. In the case of complex eventualities whose certain triad components are not only conceptually present, but are *lexicalised*, as well, the unity of these components can be represented. Although the structure of PWN is based on a hierarchical system, an alternative structure has already been accepted for adjectives in PWN. By analogy it should be possible to organise the verb synsets in a slightly modified way than nouns, as well. The tripartite structure described above may be mapped onto the system of wordnet in the form of relations. The meta-language level described by Moens&Steedman's nucleus structure can be mapped onto the level of lexicalised elements, represented by wordnet synsets. The connection of the two levels is shown in Figure 4.

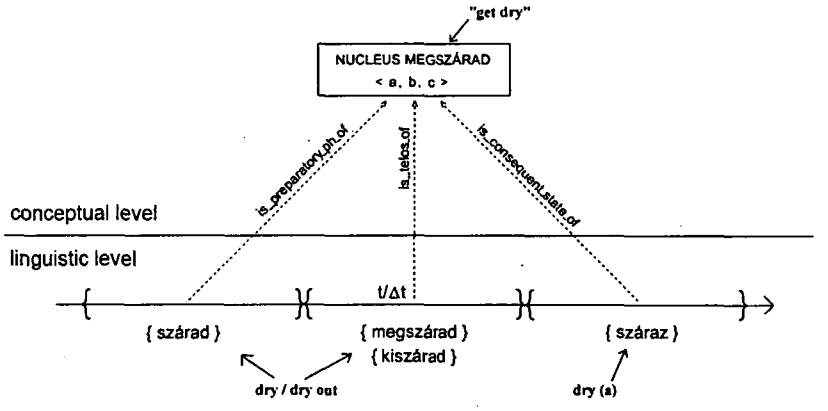


Figure 4: Applying the event-nucleus of Moens&Steedman to the synsets of WordNet

Artificial nodes introduced in HuWN (see [6]) are suitable for naming meta-language nuclei, e.g. the complex eventuality denoting the change of state from wet to dry, in the above example.[17]

The relational structure of the wordnet allows introducing three new relations according to the respective triad-components being related to the meta-language nucleus-unit, represented by an artificial node. These new relations point to the appropriate artificial node and they are called *is_preparatory_phase_of*, *is_telos_of* and *is_consequent_state_of*, respectively, based on the names of the different nucleus components.

[17] Artificial nodes are written with capital letters to distinguish them from natural language synsets.

Meanings that are lexicalized by a single verb in English but not in Hungarian can thus be distinguished: the same meaning might be present in Hungarian often as a verb with a preverb providing more aspectual information and as a verb without a preverb, more underspecified for aspectual information. In the above example, the Hungarian *szárad* and *megszárad* synsets are both equivalent to the English {dry:2}[18]. Without integrating the nucleus system into the wordnet the synset *megszárad* could be placed into HuWN only as a hyponym of *szárad*, considering all the originally available relations. However, this kind of storage would not distinguish the different implicational relation between the above mentioned two meanings, but would merge them into a hyponym-hypernym relation.[19] After having integrated the nucleus system into the wordnet, there is no need for an additional explicit relation between the components of a nucleus: they are already connected through the artificial node. Following the path of the relations *is_preparatory_phase_of* and *is_telos_of*, it is easy to determine that the synset *szárad* represents the preparatory phase of the nucleus whose another lexicalized component is *megszárad*, hence *megszárad* implies *szárad*, while the implication does not hold in the other direction[20].

As we have seen, verbs belonging to the same triad (often with and without a preverb respectively) can be placed more accurately in HuWN with the help of the new relations. Furthermore, the relation *is_consequent_state_of* is not restricted to verbs, the third component of the triad mentioned above is the adjective synset *száraz* ({dry:1}). This psycholinguistically relevant piece of information is present in HuWN but would be lost if we had strictly held onto the structure of PWN without the tools for representing triads.

3.2.1 Triads in HuWN

Given that the presence of all the three triad-components presupposes an eventuality with a preparatory phase, a telos and a consequent state, we set out from the assumption that a possible domain where the adaptation of the nucleus-structure to the wordnet would be likely to prove useful would be that of verbs denoting some kind of change (e.g. change of state). Accordingly, we have chosen the unique beginner synsets[21] {változik:1} ({change:1}) and {változtat:1} ({change:2}) — whose hyponyms alone make up at about the fourth of the verb synsets in the WordNet — to test the adaptation of the nucleus-structure to HuWN. When encoding the presence of the nucleus-components, the assumption that the hyponym trees of the above two unique beginner nodes would bear several synsets that lend themselves to being represented in a nucleus-structure proved to be right. Our results are

[18] *szárad* 'is drying' (v), *megszárad* 'get dry' (v), *száraz* 'dry' (a)

[19] By analogy to the nominal hypernymy relation, one way of conceiving of this relation between verbs would be basing it on selectional restrictions. E.g. the synsets {hervad, fonnyad} (fade, wither) and {rohad} (rot) would have such an ideal hypernymy relation, since the former selects plants as subject, while there is no such restriction on the subject of the latter one.

[20] See Section 2.3 for a discussion on the connection (called contingency by Moens&Steedman) between the components of a triad.

[21] Synsets with no hypernyms are called *unique beginners* in WordNet terminology.

shown in Table 3.

Table 3: Proportion of nuclei under the synsets {változik:1} and {változtat:1}

nucleus type	{változik:1}	{változtat:1}
< ∅, b, c> or <x, b, c>	24 %	14 %
<a, b, c>	30 %	42 %
Altogether	54 %	56 %

After examining around 150 direct hyponyms of the mentioned two unique beginner synsets we have found that in more than half of the cases the application of the nucleus-structure facilitated the positioning and representation of a meaning lexicalised in Hungarian in the network.

The conceptual similarity between the two unique beginner nodes could lead to the assumption that the triads associated with them have a parallel internal structure, too. However, at a closer examination one sees that the relations between the respective components of the nucleus differ as a result of the causative-inchoative alternation. The difference may be explained through the following example: The second and third component of the nucleus <SZÁRAD, MEGSZÁRAD, SZÁRAZ> are connected by a consequent state relation, since the consequent state of the event expressed by the verb *megszárad* is the dry state of its subject. The seemingly similar structure of the nucleus <SZÁRÍT, MEGSZÁRÍT, TÚLVAN A MEGSZÁRÍTÁS FOLYAMATÁN> hides a different relation between the second and the third component: the consequent state of the eventuality refers to the object of the verb *megszárít*, meaning that the synset {megszárít} has a *causes* relation to the adjective synset {száraz} (dry). Although the *causes* relation appears in the English WordNet between the nodes {dry:1} (causative) and {dry:2} (inchoative)[22], there is no relation that would connect the verb synset {dry:2} with the adjective {dry:1}. If the Hungarian WordNet was to follow the relation patterns of PWN, a *causes* relation would have to be encoded between both causative-inchoative verb-pairs (*szárít–szárad* and *megszárít–megszárad*). However, this would unnecessarily duplicate the number of relations encoded, while the relation between the causative verbs and the cause itself, expressed by the adjective synset, would still only be indirectly visible. This is why we propose to connect the metalinguistic nodes denoting the triads with a causes relation (i.e. *megszárít* –causes→ *megszárad*), which allows for interpreting the causes-relation between the verbs of the causative and the inchoative triad as well as between the verbs of the causative triad and the adjective synset expressing the consequent state of the inchoative triad.

[22]Unfortunately the encoding of these relations is not systematic in the PWN.

4 Possible applications

Besides the fact that one of the main tasks of a wordnet is to provide a uniform representation for the idiosyncratic properties of the lexical items, the extension of HuWN in the proposed way brings practical benefits, as well. As we have seen, it can be easily deduced from the triad whether a given verb is telic or atelic, perfect or progressive, respectively. A Hungarian-English MT system can be improved by using this information provided in the HuWN, e.g. in the area of matching the verb tenses in the source and the target language more appropriately. Since there are only two morphologically marked tenses in Hungarian (present and past), a rule-based MT system would select the same two tenses in the target language, simple present and simple past, respectively. Inaccurate translations would emerge inevitably. However, the above outlined information integrated into HuWN would improve the system. In Hungarian, for example, morphologically present tense forms of a telic verb have a future reference. The English equivalent of the Hungarian sentence *Felhívom Pétert* is not *I call Peter*, but *I will call Peter.* Similarly, progressive past tense verb forms should be matched with the past continuous form of the appropriate verb, instead of selecting the simple past form: the Hungarian *Péter az udvaron játszott* should be matched to the English *Peter was playing in the yard*, instead of the now expected *Peter played in the yard.* Aspectual information may be used in generating sentences, as well, whether it be translation from English to Hungarian, or some other tasks requiring generation.

These properties of verbs may be helpful in machine comprehension, as well. The knowledge of such idiosyncratic properties of verbs is an important component of the inner representation of a computer. Without this information, just by considering the temporal adverbials (possibly) present in the sentence, it is not possible to represent or reconstruct the temporal structure of a narrative accurately.

5 Conclusions

In the present paper we have tried to show on the example of the Hungarian WordNet in what ways the wordnet-structure as conceived of in PWN may be exploited and extended in order to represent some language-specific and part of speech specific phenomena of typologically different languages than English, as well. Although specifically implemented for solving a linguistic situation in the Hungarian language, the implementation of the nucleus-structure in the WordNet in the form of relations might prove to be useful for other languages with a rich morphology showing aspectual distinctions, as well. Later applications of the extended HuWN will hopefully prove the direction in which we tried to point with the above detailed representation of verbs to be useful.

References

[1] Bach, E. The Algebra of Events. *Linguistics and Philosophy*, 9: 5-16, 1986.

[2] Dowty, D. *Word Meaning and Montague Grammar.* D. Reidel, Dordrecht, W. Germany, 1979.

[3] Fellbaum, C. *WordNet An Electronic Lexical Database.* MIT Press, 1998.

[4] Kiefer, F. *Jelentéselmélet.* Corvina, Budapest, 2000.

[5] Kiefer, F. *Aspektus és akcióminőség. Különös tekintettel a magyar nyelvre.* Akadémiai Kiadó, Budapest, 2006.

[6] Kuti, J., Vajda, P. and Varasdi K. Javaslat a magyar igei WordNet kialakítására. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia*, 79–88, Szeged, 2005. Szegedi Tudományegyetem.

[7] Moens, M., Steedman, M. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28, 1998.

[8] Tufis, D. et al. BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. *Romanian Journal of Information Science&Technology*, 7(1-2):1–35, 2004.

[9] Vendler, Z. Verbs and Times. *Philosophical Review*, 66:143–160, 1957.

[10] Verkuyl, H. J. *On the compositional nature of the aspects* Foundations of Language, Supplementary Series, 15, Reidel, Dordrecht, 1972.

[11] Vossen, P. EuroWordNet General Document. Technical Report EuroWordNet (LE2-4003, LE4-8328), 2005.

Acta Cybernetica 18 (2007) 329–340.

How to Represent Meanings in an Ontology *

Miklós Szőts† Tamás Grőbler† and András Simonyi †

Abstract

We work on a method for giving a formal semantic representation of natural language texts. The semantic representation is generated in an ontology, on the basis of morphological and syntactic information. The task of the semantic analysis is to create instances in the ontology that contains the world model, i.e. to create those individuals and relations that correspond to the situation described by the text. The knowledge base of the semantic analyser is stored in an OWL ontology. This paper gives an overview of the system, and we discuss those questions of ontology design that require special attention in the context of meaning representation. We also present a software prototype that is based on the method and generates electronic medical records from free-form medical texts.

Keywords: ontology, semantics, natural language processing, electronic medical record

1 Introduction

The research presented here aims at giving a formal representation of the semantic content of natural language texts. A method for formal meaning representation can be put into use in many areas—we have already implemented a software prototype that generates formally structured medical records from free-form texts, but our most important long-term goal is to develop a semantic search engine (text mining tool).

Currently we handle only the descriptive function of language, i.e. we represent only the meaning of declarative sentences. Our approach is based on the idea that the meaning of a text (or text fragment) is the representation of the described situation in a world model.

*This work was supported by research grants GVOP-3.1.1-2004-05-0363/3.0 and NKFP-2/042/04 (MEO).

†Applied Logic Laboratory, E-mail: {szots,grobler,simonyi}@all.hu

The method uses a knowledge base consisting of the following modules:

- knowledge about the world, that is, the world model;
- linguistic knowledge,
- a mapping representing the connections between these components.

Both the world model and linguistic knowledge are represented in ontologies. Using these modules we are able to store and use the knowledge that is required for interpreting natural language texts: an algorithm can be formulated that tracks the referential connections between expressions of a given text and elements of the world model on the basis of the above mentioned mapping, and 'populates' the world model with the situations, objects, relations etc. that are described. These additions to the world model ontology are, in effect, formal representations, that can be collectively described as the meaning of the text in question.

The paper is organised as follows. Section 2 surveys issues connected with the world model, while Section 3 gives an overview of the linguistic knowledge base, its relation to the world model, and, most importantly, the method of using the three modules for generating semantic representations. The remaining four sections contain a description of the 'proof of concept' software prototype we have implemented, a short discussion of related work, and a summary of our plans for further development.

2 The world model

In order to be able to represent the meaning of texts about a given area of knowledge, the ontology has to be capable of representing those situations that typically occur in the texts to be analysed. Consequently, it has to contain both domain-specific concepts, and concepts corresponding to everyday words that connect the domain-specific expressions with each other. This requires, first of all, a satisfactory top-level ontology, which determines what is expressible in the system. In the following we discuss some of the most important issues that has to be faced when designing a top-level ontology. We raise these problems on a general level, but it is to be emphasised that the categories of a special domain ontology might be radically different from those of general purpose ontologies. For instance, it is totally unnecessary to include the common genus HUMAN of the concepts PATIENT and MEDICAL_STAFF_MEMBER in a medical ontology.

2.1 Eventualities

Eventualities (also known as occurrences or perdurants in the literature: they include both events and states) are of crucial importance for natural language processing, since they are those elements of reality (or our representation thereof) that are usually referred to by verbs. We have introduced the relation PARTICIPATES_IN between the concepts ENDURANT and EVENTUALITY (the former concept applies

to all physical or abstract objects that persist through time), and thematic roles are considered to be subrelations of it, determining those concepts whose instances can be e.g. the actor, the object etc. of an event—see [15].

2.2 Properties

How should we represent that 'the sky is blue' i.e. that 'the colour of the sky is blue'? Or, for that matter, how should we represent the sentence that 'on the 8th of November, 2006, the patient's blood pressure was 220/178 mmHg, measured on her left arm, when she was sitting'? Although the first example seems to be simple—even if there is a hidden time dependence—the second illustrates an obvious problem: we have to represent the fact that an instance of the concept PATIENT has a property (BLOOD_PRESSURE) with a given value ('the patient's blood pressure') which depends on various parameters (position, place of measurement, time). Obviously, different properties will depend on different parameters. We chose to solve this problem by reifying the relation 'has property,' and to connect the reified, individualised properties (also known as tropes in the philosophy literature) with the relations BEARER, HAS_VALUE, and relations corresponding to the parameters (IN_POSITION, HAS_PLACE_OF_MEASUREMENT, HAS_TIME in the example) to the relevant objects and values.

2.3 Time

Almost every domain's representation requires a representation of time, and we opted for a relatively simple treatment: a distinction is made between time intervals and time points, and the class of time points is mapped onto a linear scale. We introduced the HAS_START and HAS_END relations with the concepts TIME_INTERVAL as their domain and TIME_POINT as their range. Unfortunately, this simple picture is spoilt by the problem of granularity: time expressions, like *day*, or *month* can refer either to time intervals or time points, depending on the context. This problem can be resolved in a number of ways—we chose to take the referents of these expressions exclusively as time points that can be the beginning or end of certain time intervals. It is the task of the semantic analyser to find those time expressions that in fact refer to time points serving as endpoints of intervals.

2.4 Location

The representation of locations is essentially different from that of time points and intervals, since there is no unified, common sense coordinate system for them. Consequently, we have to use other concepts to determine locations. In medical contexts we encounter two, totally different ways of referring to places: certain medical concepts are connected to body parts (e.g. a liver tumour), while in other cases medical units (hospitals etc.) are the locations that are referred to. We represent these two ways of locating an object by two different relations.

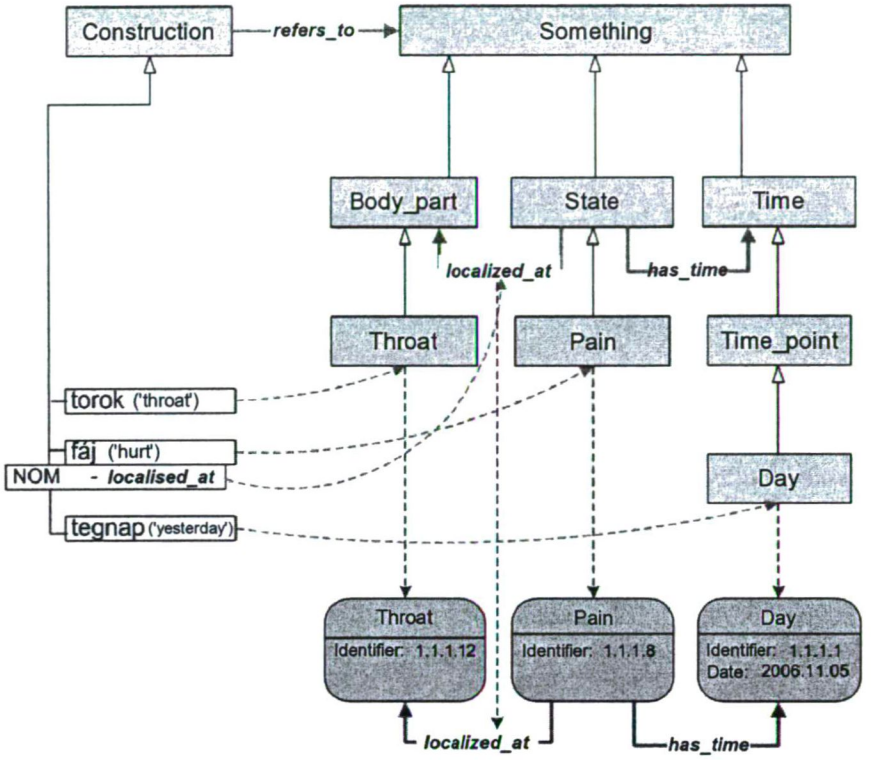


Figure 1: Ontology fragment and semantic representation corresponding to the sentence *Tegnap fájt a torka* '[She/He] had soar throat (her/his throat was hurting) yesterday.' Unlabelled continuous arrows indicate the generic relation. Concepts representing lexemes are connected by the REFERS_TO relation with the corresponding concepts of the world model. The meaning representation is given by the instances shown at the bottom.

3 The linguistic knowledge base and its connection with the world model

In accordance with the MEO model [14], our knowledge base consists of a conceptual layer (the world model) and a linguistic layer that contains the linguistic elements referring to the concepts. Presently, the linguistic component is no more than a lexicon, which lists lexemes (words, affixes, idioms) and their disambiguated versions. The meaning of lexemes are given by restrictions on the REFERS_TO relation (see Figure 1). In addition, a group of lexemes and morphological marks that figure in case-frames also refer directly to concepts or relations. During semantic analysis we create instances of the concepts that correspond to lexemes in the text, and connect them with relations that (might) hold between them according to the

world model. The generated instances can also have data type properties.

Of course, the real problem in meaning representation is not the satisfactory representation of word meanings, but the representation of the syntactic connections that organise words and morphemes into a meaningful sentence—that is, the representation of the head–dependent relation. This can be done in two ways, but the problem of ambiguity has to be faced in both cases.

1. **Representing the case-frame.** A head–dependent relation that is indicated by the grammatical case of the dependent holds between word instances—these syntactic relations correspond to thematic roles in the world model. The correspondence is not universal: for instance, the nominative usually corresponds to the agent or patient role, but in fact it can represent almost any role, e.g. in medical texts it frequently stands for location. Consequently, the mapping between the lexicon and the world model has to indicate the connection between grammatical cases and thematic roles on a case by case basis for each verb: we do this by introducing a relation for each grammatical case (e.g. the relation NOM for the nominative), which determines for every verb or verb phrase the thematic role that corresponds to the case in question. For instance, in the case of *fáj* 'hurt,' the nominative case corresponds to the location role of the PAIN concept, therefore the NOM relation will connect this word with the location thematic role.

2. **Adjuncts.** Most of the affixes that indicate adjuncts in Hungarian can be taken as corresponding to relations of the world model: e.g. the *-ban* 'in' affix can stand for location or time relations.

Naturally, there are expressions to which no referent can be connected in the world model, e.g. articles, negative particles etc. These expressions have to be handled by special rules attached to the syntactic analysis (see Section 5.4).

4 The medical record generating program

The purpose of the software prototype that we have developed is to convert Hungarian texts about medical encounters to a unified medical record representation format. The generated medical record has to contain the patient's identification information, the circumstances of her medical encounters, the reported complaints and symptoms, their properties, the time span of their presence, and possibly other pieces of relevant information.

It is impossible to determine the meaning of a text without relying on syntactic and morphological regularities. Nonetheless, a precise grammatical (especially syntactic) analysis also requires semantic information. In this respect, the ideal solution would be to carry out the morphological, syntactic and semantic analysis simultaneously. Analysers working along these lines already exist for the Hungarian language [1], and our long-term plans also include the implementation of this method. Until then, we consider morphological and syntactic analysis as provided by a preprocessor.

Grammatical preprocessing is performed by MorphoLogic Ltd's morphological and syntactic analyser [10]. The syntactic analysis produces a parse-tree in XML format. The analysis is head-driven: whenever it is possible, larger units, groups, and phrases refer to the terminal element that can be regarded as their head. Relevant morphological characteristics of terminal elements are also indicated, which helps in detecting the semantic connections to a great extent.

Text normalisation is especially important in the case of the medical documents to be processed in the project. We have to handle foreign (typically Latin) words that are characteristic of the subject, abbreviations and their various versions, numbers, and the noticeably frequent mistakes and typos resulting from fast note-taking. Currently, the system works with normalised input. In the next phase we also plan to implement text normalisation in cooperation with MorphoLogic.

To summarise, our software system consists of a grammatical preprocessor, a semantic analyser that generates the semantic representation and a medical record generator that collects those pieces of information from the semantic representation that correspond to fields on the medical record. The grammatical preprocessor is endowed with its own dictionary and grammatical database, while the database of the semantic analyser is the ontology discussed above. The ontology is stored in OWL format, which is a standard, description logic-based ontology language maintained by the W3C consortium [6]. We edit the ontology with the Protégé ontology editor, and our Java code interacts with it using the Jena Semantic Web Framework.

5 Our results

Representing meanings in an ontology raises many problems. Some of these can be solved by adequate design and suitable use of the ontology, but in other cases we have to go beyond the limitations imposed by the ontology and handle the difficulty by external means. We hope that using the previously mentioned lexicalist grammar it will become possible to deal with all of these problems in a uniform way that is internal to the ontology.

5.1 Disambiguation, anaphora resolution

One of the most frequently encountered challenges of free-form text analysis is ambiguity, which appears at several levels of the system. The result of the grammatical analysis can be ambiguous at any of the lexical, morphological and syntactic levels. There can be more than one constructions in the ontology that correspond to a single lexeme, and these constructions in turn will refer to different concepts. Also, in many cases several relations can hold between two concepts.

The presence of anaphoras can also be regarded as a kind of ambiguity, since we usually have to choose from more than one referent candidates. In such cases we look for already processed referents having the properties that are required by the anaphora.

In all of these cases, the basis of disambiguation is the completeness of the competing representations, in the sense of the amount of information they extract from the text under analysis. We calculate this quantity by measuring the specificity of relations in the representation in question: a representation is considered more complete than an other if it contains a larger number of more specific relations.

5.2 Unknown words

However large ontology we build, the text to be analysed will necessarily contain unknown words. We could simply omit these from the semantic representation, but it can easily happen that precisely an unknown word holds together certain parts of the meaning that would otherwise fall apart, since the syntactic structure of the sentence might unambiguously determine the unknown expression's role. Accordingly, we create instances of the THING top category to represent unknown words, and have also introduced the relation DUMMY_RELATION for representing unknown relationships.

5.3 Time and cardinality

Texts frequently do not use tense or do not use only tense to indicate the time of an event or state. In other cases, they contain indexical time adverbs, e.g. *most* 'now' or *két napja* 'for two days.'

Nonetheless, in order to be able to generate the medical record, we have to determine the time span of the complaints as precisely as possible. Consequently, the algorithm contains a module that tries to calculate the time of every complaint and symptom on the basis of the document's time of creation. Our method of representing time in the ontology squares well with this task (see Section 1 and Figure 2).

Cardinalities also present a challenge to ontology designers. Considering the purposes of the project, we opted for broadening the extensions of concepts in order to cover not only single individuals having a certain property, but *sets* of such individuals as well. The fact that an instance is a set is represented by the presence of information about its elements or cardinality. We have distinguished numerical CARDINALITY, which can be expressed by a number, from QUALITATIVE_CARDINALITY (e.g. many, few—see Figure 3).

5.4 External methods

Natural language texts contain many phrases that instead of referring to elements of the world model, modify the meanings of other expressions or their relations to each other. To handle these expressions, we created a dictionary of function words, which contains specific instructions, written in a simple syntax, about the treatment of each listed word.

In the simplest cases, there is nothing to do (e.g. conjuncts like *de* 'but' etc.). In these situations the syntactic analysis already contains the information (e.g.

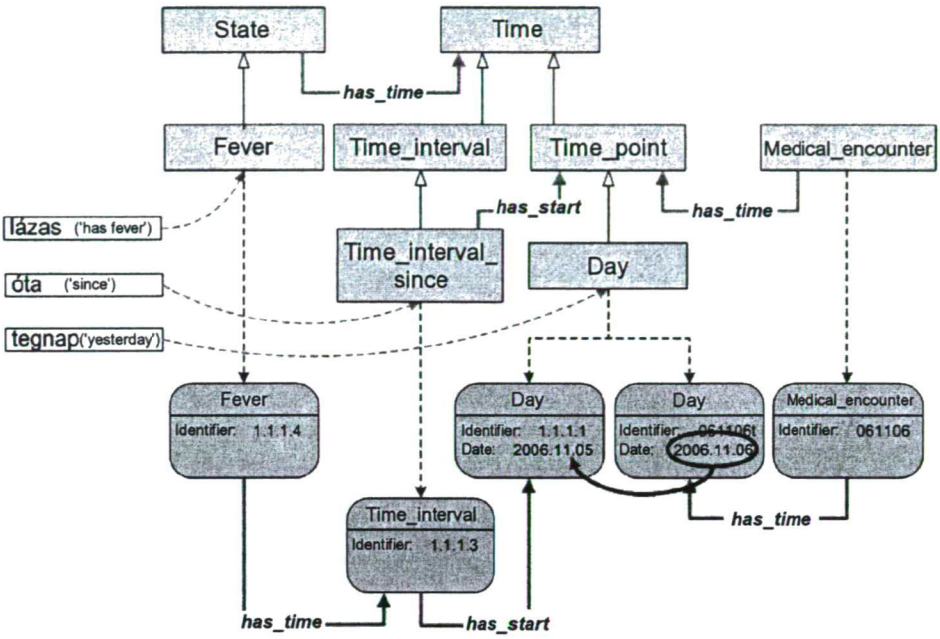


Figure 2: Ontology fragment and semantic representation corresponding to the sentence *Tegnap óta lázas* '[She/He] has had fever since yesterday.' The expression *tegnap óta* 'since yesterday' is represented by a time interval that started on the day preceding the day of the medical encounter when the complaints were recorded.

coordinated clauses in the case of conjuncts), or we do not want to represent the information conveyed by the expression (e.g. the subjective element in the case of *csak* 'only'). Negative particles and affixes are marked during preprocessing, and truth values are determined on this basis. Similarly, we extend dependents connected to lists or coordinated structures to all of the relevant elements already in the preprocessing phase.

There are expressions with special meanings that require complex handling of the sentence in question. For instance, the phrase *egyéb panasza nincsen* 'does not have other complaints' means that the set of complaints (whose elements are all problems and symptoms that were previously mentioned in the text) is closed in the sense that no new element can be added to it.

6 Related work

Although it is still widespread to consider ontologies as consisting of linguistic expressions, a growing number of research projects rely on the distinction made here between ontology, understood as a system of concepts, and expressions that *refer to* these concepts. The OntoWordNet project [5], for instance, aims at working out

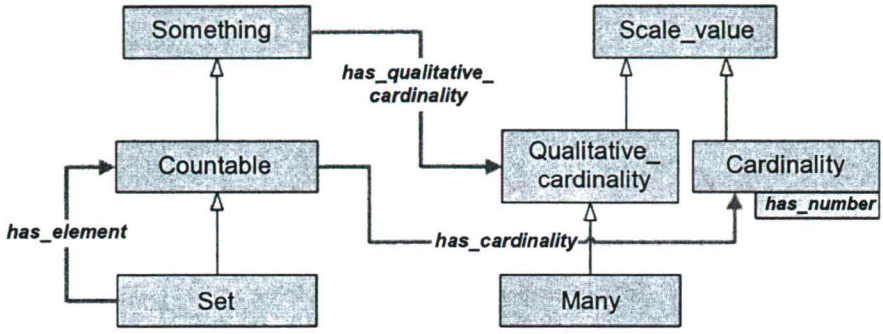


Figure 3: Representation of sets and cardinalities in the ontology.

the connections between the DOLCE foundational ontology [9] and the WordNet lexical database [4], and several projects use ontologies for improving on text search results ([8], [11]), and text generation.

Of the many ontology-based approaches to natural language processing, the closest to our work is the "ontological semantics" presented in [12], although the two projects have been developed independently. Both approaches generate the textual meaning in ontologies, and separate the lexicon from the world model. In contrast to our implementation, which is based on the well established OWL ontology language, ontological semantics implementations use a special, non-standard formalism (so called TMRs) for semantic representation.

An important, relatively early contribution to the semantics of the Hungarian language in the context of natural language processing is also worth mentioning: [3] presents a dependency grammar-inspired system of semantic representation for Hungarian texts, where, similarly to our approach, the semantic links between head verbs and their dependents play a central role.

Even though we do not aim at giving a general semantic theory for natural languages, our work is related to a number of influential approaches in this area as well.

First of all, our ontology based semantic representations share many features of the representations used in Discourse Representation Theory [7]. Both approaches build a partial formal representation of the situation that is described in the represented text by introducing objects for referring expressions with the properties and relations that are explicitly mentioned. The fundamental difference is that in contrast to DRs, our representations are not separated from the general representation of the domain described by the text, but are conceived as extensions of it. This makes it easy to model the dependence of semantic content on conceptual and factual background knowledge.

The idea that the content of a linguistic unit is to be identified with its potential to extend our knowledge of the world is one of the central insights of possible world semantics [13]. One of the main differences here is that the possible world approach does not work with *syntactic* representations: instead, semantic content is identified with classes of possibilities, or mappings between classes of possibilities.

Finally, cognitive linguistics [2] can also be mentioned as a related approach: a specific syntactic construction *together with* its semantic representation in the ontology can be regarded as a linguistically expressed conceptualisation of a situation. In contrast to cognitive linguistics, however, we distinguish the syntactic and semantic layers of this conceptualisation.

7 The future

The medical record generating program we have presented answered many questions, but from the point of view of our long-term plans, it is only a demo having several serious limitations. Further development has to undertake the following tasks.

The architecture of the system is untenable in the long run: the separation of semantic and syntactic analysis is against the philosophy of the method. Therefore, our most important research goal is to work out a grammar with the help of which the syntactic and semantic analysis could be carried out simultaneously—in this way the two processes could cooperate and help each other, e.g. when trying to resolve ambiguities. The grammar in question will be, most probably, a lexicalist grammar. In this case, we will be able to use descriptively the rules that we currently build in the semantic analysis procedurally.

The limitations of the Protégé ontology editor forced us into certain artificial solutions that made the structure of the ontology slightly complicated. We will be able to use a much more natural ontology when the MEO ontology editor will be ready for use [14].

A real life application requires a very extensive ontology, the creation of which would take several years. Therefore we are trying to find a solution that, at least in the case of certain applications, would not require a complete ontology—for instance, the ontology could be built at the time when the program is in use. The most important such application would be a semantic search engine that would produce results based on the meaning of search expressions.

Although the prototype presented here shows the usability of our method, several questions can be raised regarding the analysis of more complicated texts (e.g. the problem of universally or existentially quantified sentences)—we intend to further develop the method and make it capable of coping with these, presently problematic cases as well.

References

[1] Alberti, G., Balogh, K., Kleiber, J., and Viszket, A. A totális lexikalizmus elve és a GASG nyelvtan-modell. In Maleczki, M., editor, *A mai magyar nyelv leírásának újabb módszerei V.*, pages 193–218. Szeged, 2002.

[2] Croft, W. and Cruse, D. A. *Cognitive Linguistics.* Cambridge University Press, 2004.

[3] Farkas, E. and Naszódi, M. Magyar nyelvű mondatok elemzése természetes nyelvű interfész céljából. Technical report, SZTAKI, 1990.

[4] Fellbaum, C., editor. *WordNet: An Electronic Lexical Database.* The MIT Press, 1998.

[5] Gangemi, A., Navigli, R., and Velardi, P. The OntoWordNet Project: extension and axiomatisation of conceptual relations in WordNet. In *International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE 2003)*, Catania, 2003. http://www.loa-cnr.it/Papers/ODBASE-WORDNET.pdf.

[6] Horrocks, I., Patel-Schneider, P. F., and Hayes, P. OWL web ontology language semantics and abstract syntax. W3C recommendation, W3C, 2004. http://www.w3.org/TR/2004/REC-owl-semantics-20040210/.

[7] Kamp, H. A theory of truth and semantic representation. In *Formal Methods in the Study of Language, part 1.* Stichting Matematisch Centrum, Amsterdam, 1981.

[8] Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. Semantic annotation, indexing and retrieval. *Journal of Web Semantics*, (2), 2005.

[9] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L. The WonderWeb Library of Foundational Ontologies: Preliminary Report. http://www.loa-cnr.it/Papers/DOLCE2.1-FOL.pdf, 2003.

[10] Merényi, Cs. and Tihanyi, L. A MetaMorpho fordítóprogram projekt 2006-ban. In *MSZNY 2006*, pages 169–179, Szeged, 2006.

[11] Nagypál, G. Improving information retrieval effectiveness by using domain knowledge stored in ontologies. In *OTM Workshops 2005, LNCS 3762*, pages 780–789. Springer, 2005.

[12] Nirenburg, S. and Raskin, V. *Ontological Semantics.* The MIT Press, 2004.

[13] Stalnaker, R. Assertion. In *Syntax and Semantics 9.* Academic Press, New York, 1978.

[14] Szakadát, I., Szőts, M., and Gyepesi, Gy. MEO – ontology infrastructure. In Magyar, G., Knapp, G., Wojtkowski, W., Wojtkowski, G., Zupancic, J., and Wrycza, S., editors, *Advances in Information Systems Development: New Methods and Practice for the Networked Society, Proceedings Information Systems Development.* Springer, In press.

[15] Szőts, M. and Lévay, Á. Szerepfogalmak az ontológiában – az OntoClean metodológia továbbfejlesztése. In *MSZNY 2005*, Szeged, 2005.

Acta Cybernetica 18 (2007) 341–354.

Outlines of a Model of General Ontology*

Károly Varasdi† Zsófia Gyarmathy† András Simonyi‡ and Dániel Szeredi‡

Abstract

This article outlines a new model of general ontology that draws significantly on the results of contemporary philosophy and cognitive science. It combines ideas from Gärdenfors's Conceptual Spaces Model and Edmund Husserl's philosophical insights concerning ontology and dependency. We put forth a possible architecture for general ontologies based on a "horizontal" (dependency-based) and a "vertical" (abstraction-based) arrangement of the concepts in the ontology.

Keywords: general ontology, conceptual spaces, modal dependence

1 Introduction

According to Thomas Gruber's oft-cited slogan, an ontology is "an explicit, formal specification of a conceptualization" which is mutually accepted by the communicating agents [4]. While this characterization can be met relatively easily in the case of narrow-scope domain ontologies, large-scale general ontologies pose special problems. In particular, in the case of general ontologies:

- What is meant by "mutual acceptance"?
- What is meant by "conceptualization"?
- How should "an explicit, formal specification" of a general ontology be construed?

*The research reported here was carried out within the framework of the Hungarian Unified Ontologies Project (MEO – Grant No.: NKFP-2/042/04.), although the results did not get implemented, but stayed on the level of theoretical research. Thanks are due to Enikő Héja, Iván Mittelholz, István Szakadát and Miklós Szőts, all of whom helped the formation of the basic ideas of this article. Obviously, none of them is responsible for the mistakes that may have remained in the text.

†Research Institute for Linguistics, Hungarian Academy of Sciences, E-mail: {varasdi, gyzsof}@nytud.hu

‡Media Research and Education Centre (MOKK), Technical University, Budapest, E-mail: simonyi@gmail.com, daniel@szeredi.hu

The structure of the paper is as follows. In **Section 2** we describe Peter Gärdenfors's Conceptual Spaces Model, which we hope provides a putative answer to the first two questions above. The rest of the paper is devoted to answering the third question. In **Section 3** we enlist a set of meta-properties and a distinguished relation (called *dependency*) which is inspired by Edmund Husserl's seminal work, on the basis of which a general ontology can be organized. We also give a concrete example as an illustration. In **Section 4** we briefly discuss some questions pertaining to the particular choice of ontology description language (more specifically, AVM's or DL's). Finally, **Section 5** provides a short summary of the paper.

2 The Conceptual Spaces model

In the case of general ontologies, *mutual acceptance* is guaranteed by the high degree of similarity between the cognitive structures of different members of the human kind. Therefore we may turn to the cognitive sciences for a theory of such structures. A recent and quickly developing branch of cognitive science has abandoned those mechanisms that only assume purely symbolic representations; instead, it operates on broader assumptions that fully acknowledge the importance of the spatial element in human reasoning. This approach is eminently exemplified in the Conceptual Spaces model, developed by Peter Gärdenfors in, for instance, [3], whose philosophical forerunner is Robert Stalnaker's [10]. Stalnaker's goal was to work out an alternative foundation of modal logic. According to Stalnaker, any entity, whether actual or possible, can be represented as a vector in an abstract space whose dimensions are the independent properties that can be predicated of the entity. For example, a particular red ball b with a radius of 3 cm may be localized in a two dimensional space one of whose dimensions consists of all the possible colors and the other one consists of all the possible radii. In this space b is identified as being at the tip of a position vector whose projection on the color dimension falls in red and its projection on the size dimension falls on the value of 3 cm. It is easy to see that in this space each and every position vector determines a possible object — a concrete ball with a particular color and size. This simple example also shows the connection of the model with the traditional symbol-based approach. For instance, the complex property (concept) "to be a red ball with a radius of less than 4 cm" will be represented as a set of points P in this two dimensional space, and the proposition that b has this property simply translates into checking whether the position vector of b ends in region P or not. Similarly, to the concept "to be a red ball with some radius" there belongs a region $P' \supseteq P$, and the relationship between the two regions further makes it possible to establish the inference that any red ball with a radius less than 4 cm is a red ball as well. This approach has other forerunners in philosophy beside Stalnaker's. The theory, according to which any physical entity can be seen as the collection of its properties, is known in the history of philosophy as *trope theory* [9]. The tropes of a particular entity are those "pieces of property" that belong to it at a particular point in time; for example, the specific color of a particular rose at a particular time. Since the

same trope cannot belong to different objects, the fact that two roses are exactly of the same color may be expressed by saying that the color tropes belonging to the roses are perfectly similar (without being identical). In the present approach we take tropes to be primitive entities, the bundles of which make up complex objects (e.g., physical objects). (In this we are following the DOLCE Ontology [8].)

Gärdenfors builds his theory on the philosophical base described above. But he also wants to put empirical content into Stalnaker's ideas. According to Gärdenfors, the inherent organizing principle within the particular dimensions is *similarity*, that is, the more similar two properties, say, shades of color, are, the closer they are located in the color dimension. He also proposed a method resembling factor analysis for identifying the set of relevant dimensions (see [3] for details). However, in the project we mentioned at the beginning, we chose to identify the relevant dimensions manually, since our task was to organize various lexical material (several word-meanings) in a coherent way, so we could borrow the methods of componential semantics (see e.g. [7]).

Gärdenfors, following Stalnaker, identifies concepts with regions in the conceptual space. On the basis of similarity as the main organizing principle of cognitive dimensions he is able to derive some very general features of human cognition concerning, e.g., learnability, but since that issue is beyond the scope of the present article, we refer the interested reader to the works cited above.

It is an important fact concerning dimensions that their values are linearly ordered. Strictly speaking, this is not a necessary condition (certain dimensions might have a different structure), but more often than not they are indeed linear, so we adhered to this assumption in our work.

While our theoretical commitments are similar to those of the DOLCE ontology, there are important differences. For example, whereas DOLCE is primarily a theory of top level categories, we are equally interested in lower level concept descriptions. Because of this, we had to find answers to questions that the writers of DOLCE did not have to face.

3 The structure of a general ontology

A general ontology is a description of the various connections between general concepts, that is, a system of concepts. A system of concepts is, therefore, a relational structure over the set of the concepts involved. These relations can be sorted in two broad types: the horizontal and the vertical. Let us start with the former.

3.1 Horizontal organization

Under *horizontal organization* we mean the necessary (essential) connections, or dependency relations, to be more accurate, between the various types of entities that concepts of the ontology denote. For example, such is the fact that for any instance of color there corresponds a particular instance of surface on which it ap-

pears. An instance of color depends on an instance of surface in the sense that it could not manifest itself without the latter. This also means that this relation is necessary: each instance of color necessarily involves the existence of a corresponding instance of surface. This example also illustrates the fact that dependency relations need not be asymmetric, since one could also argue that an instance of surface also necessarily implies the existence of an instance of color. An example in which symmetry obviously does not hold is the following: each event of wedding essentially depends on the existence of a bride, while the dependency naturally does not hold the other way round (a bride can exist without ever participating in a wedding). Relations like this exemplify the highest level of connections between concepts, which are conceptually necessary, and thus do not tolerate exceptions. These, therefore, form the most general layer of ontology. This approach to ontology is traditionally attributed to Edmund Husserl [5], and there have been recent attempts at laying the dependency relation on stricter, mathematical foundations, notably by Kit Fine [2]. In the sequel, however, we are going to follow a simpler method, which is more suitable for our goals, than Fine's formalism.

3.1.1 A formal characterization of the dependency relation

It would be beyond the scope of this article to attempt an exhaustive characterization of the dependency relation. In what follows, therefore, we will only put a *necessary condition* on this type of relation, which is meant to filter out at least some of the relations that are not dependency relations. Let A, B be two arbitrary types of the ontology (e.g., the concepts of surface and of color). If R is a dependency relation between A and B, then R has to observe the following condition:

$$\Box\forall x(x \text{ instanceOf } A \rightarrow \exists!y(y \text{ instanceOf } B \wedge R(x,y))). \tag{1}$$

In words: it is *necessary* that for any instance x of A, there exists exactly one instance y of B, such that x is in relation R with y. (This basically means that R is necessarily a function from A to B.) In natural languages dependency relations are often expressed by the genitive case (e.g., *color of*, *shape of* etc.), but — as shown by the example of the wedding — this is more of a tendency than a rule.

The conceptual and intensional character of the meta-predicate "dependency" is guaranteed by the presence of the '$\Box$' (the necessity operator). Thereby, given that there are various degrees of necessity, we arrive at dependency relations of different strength. In the case of the connection between color and surface above, we saw an example of the so-called metaphysical necessity. This type of necessity is extremely strong, almost of logical strength. Let us now consider a weaker type of necessity and the dependency relation based on it: if, for instance, $\Box$ is construed as "it is necessary according to the laws of biology that", and A is identified as the type of man and B as the type of woman, then

$$\Box\forall x(x \text{ instanceOf man} \rightarrow \exists!y(y \text{ instanceOf woman} \wedge \text{mother-of}(x,y)))$$

will be true, whereas

$$\Box\forall x(x \text{ instanceOf man} \rightarrow \exists!y(y \text{ instanceOf woman} \wedge \text{sibling-of}(x,y)))$$

will be false. In other words, the mother-of relation may be a dependency relation between the types of man and woman (since it is biologically necessary that every man has a mother), while the sibling-of may not be one, since it is not biologically necessary for a man to have a sister. The moral of the example is that using '$\Box$'-es of various strength, we arrive at different degrees of dependency, which renders it possible to have a smooth transition from the most general conceptual structures to the more specific ones, which apply within the domains of different professions.

It should be noted that if R is chosen to be the identity relation, then the resulting

$$\Box\forall x(x \text{ instanceOf } A \rightarrow \exists!y(y \text{ instanceOf } B \wedge x = y)) \iff$$
$$\Box\forall x(x \text{ instanceOf } A \rightarrow \exists!y(x \text{ instanceOf } B)) \iff$$
$$\Box\forall x(x \text{ instanceOf } A \rightarrow x \text{ instanceOf } B)$$

formula is the familiar generic (isa) relation between A and B. Indeed, identity may be seen as a *trivial* form of dependency, because any entity tautologically depends on itself.

From the foregoing discussion, lessons relating to the *formal specification* can also be drawn: we need a sufficiently strong graph description language to describe dependency relations. In order to decrease the strength of the language used for describing the various types, information relating to possible circular or symmetric dependencies is distributed throughout the whole of the ontology, rather than packed in just the given concept descriptions. Thereby, although particular concept descriptions are formalized with DAG's (*directed acyclic graphs*), potential circular dependencies can be restored through comparing information stored in the different concept descriptions of the ontology. In the examples below, we will be using AVM's (*Attribute–Value Matrices*) to describe DAG's, but it should be noted that research is currently being conducted to determine the appropriate language (primarily in the area of Description Logics [1], see Section 4). Nodes representing particular types will thus correspond to matrices and eventually values (variables, tropes or even whole regions of dimensions), while edges will correspond to attributes. An attribute–value pair A_i–V_j in the matrix of a given type T is interpreted as the necessary existential implication discussed above stating that there is exactly one value of type V_j belonging to T such that it is in relation A_i to it.

3.2 Essential and contingent properties

Dependency relations belonging to a given type include necessary constraints pertaining to the individuals of the relevant type. For instance, no event *can ever be* an instance of wedding unless an actual instantiation of the bride's role by someone can be identified.

Our knowledge about reality, however, can be grouped into two classes. One class comprises the above-mentioned *a priori* (conceptually necessary) connections, while the other one includes *a posteriori*, or contingent connections. A priori, or

necessary connections belong to the net that — according to Wittgenstein[1] — we lay upon reality in order to be able to manage the originally formless "mass". The laws of this net are, therefore, the laws of language and logic. However, what actually fills the meshes of the net depends on the characteristics of actual reality, and is therefore contingent. A general ontology should also be able to depict the contingencies characterizing our world.

Traditionally, the necessary properties (features) of individuals are called essential properties, and have already been discussed extensively above. Essential and contingent traits, however, are of course linked. For example, the fact that there is always some actual color belonging to every actual surface (in the macro-sized world) is necessarily true, but this statement, naturally, does not specify that this color be, for instance, red. Which color the given surface will actually possess will depend on the contingent properties of reality. Similarly, it can be essential for an individual belonging to a certain type that the value of one of its properties fall into a given interval, while which value it specifically possesses, may be solely contingent.

Using our general concepts, the meshes of the "purely a priori net" can be divided into smaller units. In these smaller "compartments", various types of experience may possibly be present. Experience, however, cannot contradict the "geometry of the net", which is defined in the a priori statements, but it can contain elements, though, that characterize with a substantial probability the objects to be found in the given mesh of the net. These empirical generalizations that are allowed to make an exception render us capable of making default inferences, which can, of course, be "contradicted" by actual instances. Such a generalization with only default force is called a *proprium* in the present study. Below, we are going to elaborate on this and other concepts relating to contingency.[2]

3.3 Contingent properties

Contingent properties can be subdivided according to how stably they characterize a certain object in time. First let us define the class of contingent properties in general (accidences), then we will proceed to the definition of proprium and phase, its subclasses.

Accidence A property $\mathcal{A}$ is accidental in c, if c possesses $\mathcal{A}$, but not necessarily so; in other words, if it possible for c to exist but not possess $\mathcal{A}$.

During the existence of the entity c, there *may* be periods when it does not possess $\mathcal{A}$. There need not, however, be such periods. c might possess $\mathcal{A}$ during all of its

[1] "Although the spots in our picture are geometrical figures, nevertheless geometry can obviously say nothing at all about their actual form and position. The network, however, is purely geometrical; all its properties can be given a priori. Laws like the principle of sufficient reason, etc. are about the net and not about what the net describes." (Tractatus Logico-Philosophicus: 6.35.)

[2] The terms "concept" and "type" will henceforth be treated as synonyms — allowing for some sloppiness.

existence by chance without it being necessarily and inevitably the case that it *has* to be so. This justifies introducing the concept below.

Proprium A property $\mathcal{P}$ is a proprium of c at t, if it characterizes c during all or most of the moments of its existence up to t, but is not an essential property of c.

Whether a property $\mathcal{P}$ characterizes c as a proprium cannot be decided based solely on the present temporal slice of c, but only by taking all of c's history (up till now) into consideration. A proprium is an inductive generalization based on the history of c ("c has so far been mostly characterized by $\mathcal{P}$"). Consequently, c may be lacking the property $\mathcal{P}$ at a given moment without $\mathcal{P}$ ceasing to be a proprium of c. If, on the other hand, c has been lacking $\mathcal{P}$ during most of its history, $\mathcal{P}$ is not a proprium of c.

The difference between an essential property and a proprium is the difference between the necessary and the probable, and — accordingly — while an essential attribute does not tolerate exceptions in time, a proprium does so to a certain extent. "Proprium" therefore, is an umbrella term for trend-like properties characterizing an entity persistently but not necessarily. Features that characterize an entity only briefly and temporarily, during a small stretch of time, form the subject of the next subsection.

3.3.1 Phase

The term "phase" is a back-formation of "phase space" known from physics. The phase- or state space consists of dimensions called "degrees of freedom", in which all the possible states of a system are represented such that for every possible state of the system there is exactly one point corresponding to it in the phase space. A not at all far-fetched *analogy* can be drawn between this concept and Gärdenfors's concept of cognitive space, in that the degrees of freedom of the system correspond to the dimensions of the cognitive space. The analogy is as follows.

The degrees of freedom of a system correspond to the properties that can in principle be predicated of the entity. If we project the trajectory covered by the system during its existence on the relevant degrees of freedom, the projections which the system cannot possibly leave correspond to the the range of its essential properties. Projections of the trajectory covered by the system during its existence up to t, in which the system resides "most of the time during the temporal interval from the coming into existence of c to t", correspond to propria. Finally, states which the system occupies at a given moment correspond in ontology to what we have called phases in the present paper:

Phase The properties $\mathcal{F}$ that an entity c possesses at a given time are called c's phases.

The relationship of the above meta-concepts are illustrated in the table below.

	stable in time	instable in time
necessary	ESSENCE	—
not necessary	PROPRIUM	PHASE

In the foregoing discussion, the terms "proprium" and "phase" were defined as applying to *individual entities*. However, drawing on this, analogous definitions can be established for types. A proprium of a given *type*, for example, can be thought of as all those properties that are (individual) propria of most of the actual instances belonging to the relevant type. Similarly, the concept of a phase could be extended to types, but since the value for use of this concept is rather limited, this extension will be omitted here.

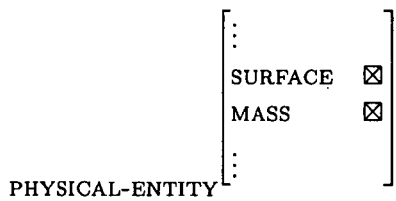
3.4 Vertical organization: three levels

Based on the distinctions drawn up above, three levels of the concept nodes in the ontology can be distinguished vertically (going downwards):

level	name	components
I.	Essential concepts	essences
II.	General concepts	essences and propria
III.	Individual concepts	essences, propria and phases

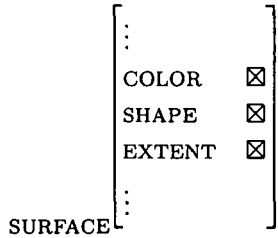
3.4.1 Level I. (essential) concepts

These describe the characteristics of the "linguistic–logical net", and are, therefore, a priori. Relations defined by these are deemed necessary; in other words, we adhere to their truth irrespective of what form reality is taking. An example for such a constraint is that (macro-)physical entities — beside numerous other necessary attributes — possess a surface and mass:

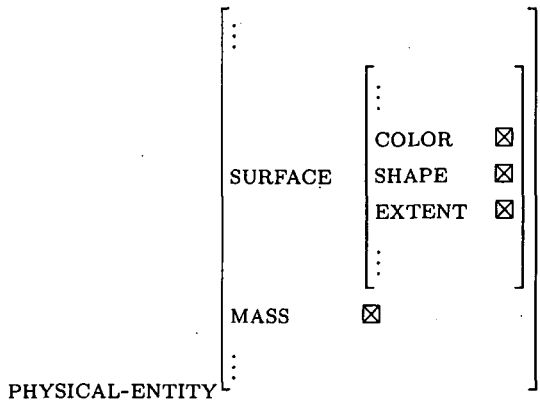


This can be interpreted in the following way: For any instance of the concept of PHYSICAL-ENTITY, there corresponds some instance of the concept of SURFACE, some instance of the concept of MASS etc. The actual value of the type assigned to it by the dependency relation cannot be specified at this point, of course, since that is contingent.

In the same way, we know a priori that a surface necessarily has a color, shape and extent (and the list may include further elements):



In other words, to any instance of the concept of SURFACE, we can assign an instance of COLOR, an instance of SHAPE, an instance of EXTENT etc. Obviously, necessarily obligatory features of a necessarily obligatory feature are also necessarily obligatory:



All of the above are level I. concepts, since they only include *purely* modally necessary relationships.

3.4.2 Level II. (general) concepts

An example for a general concept is the concept of "cat". The description of a general concept comprises two kinds of information:

1. essential information
2. contingent information

A general concept rigidly inherits all essential information from level I. concepts above it. For example, every instance of the concept of "cat" is a physical entity, hence, the concept of CAT — or to put it more accurately — the description assigned to the node belonging to the general concept of CAT will include everything that necessarily characterizes physical entities. At the same time, a general concept may include further essences, as well — for example, that the body temperature of a cat cannot be 15000°C; this we know for sure without there being any need for performing experiments to that effect. It is easy to see at this point, though, that what we are dealing with is a weaker — biological — kind of necessity, since

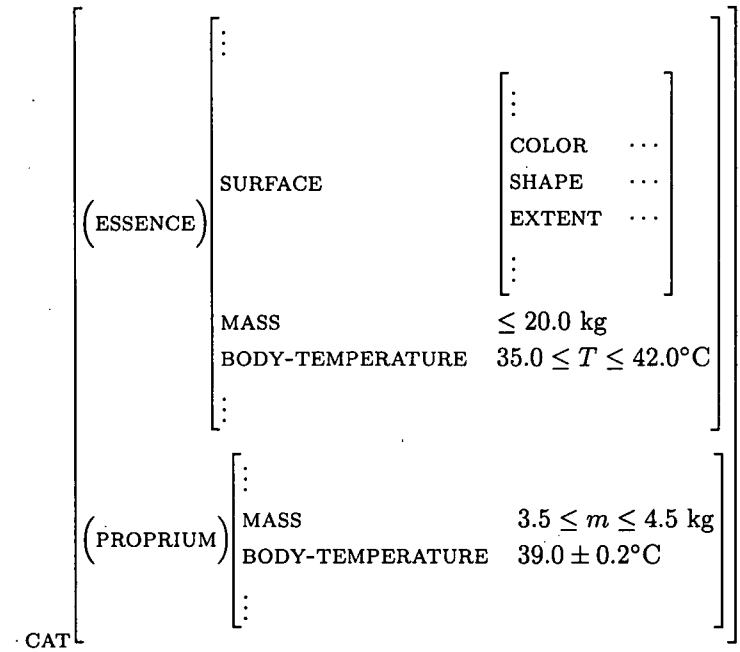
it is not in itself logically contradictory to assume that a cat remains a cat even at 15000°C, whereas it is contradictory to hold this based on our knowledge of biology. That we still wish to regard this as essential information is justified by the fact that common-sense (and presumably, scientific) thinking both deem a creature functioning at 15000°C *impossible.* We will thus consider it an essential feature of CAT that its BODY-TEMPERATURE *should* fall between 35.0°C and 42.0°C, because should it leave this range, it will cease to exist.[3]

Contingent information characterizes most of the instances, but conceivably not all of them. A piece of information like that is the fact that the BODY-TEMPERATURE of CAT takes its value from the value range 39.0 ± 0.2°C, given that the body temperature of the majority of cats falls into this range at almost every moment, which makes it a proprium. A proprium is not necessary in any respect, being merely a result of a generalization based on the actual instances of the concept, which means that a proprium can never contradict a constraint that is regarded essential.

Propria of general concepts are not necessarily generalizations based on a single person's own experience, but rather codify the accumulated experience of (the professionals of) a community. Through the mediation of culture, however, community-level experience relating to the various types of being are built into the conceptual representations of each person as contingent — but very probable — world-knowledge.

Since propria assigned to general concepts are not a priori necessities, they are subject to *default inheritance* between nodes. For example, although the BODY-TEMPERATURE of CAT ranges between 39.0 ± 0.2°C, that of the ANGORA-CAT ranges between 39.5 ± 0.1°C.[4]

[3]This information is inherited by CAT from MAMMAL in a more elaborate version of this example.
[4]The data, of course, only serve illustrative purposes.



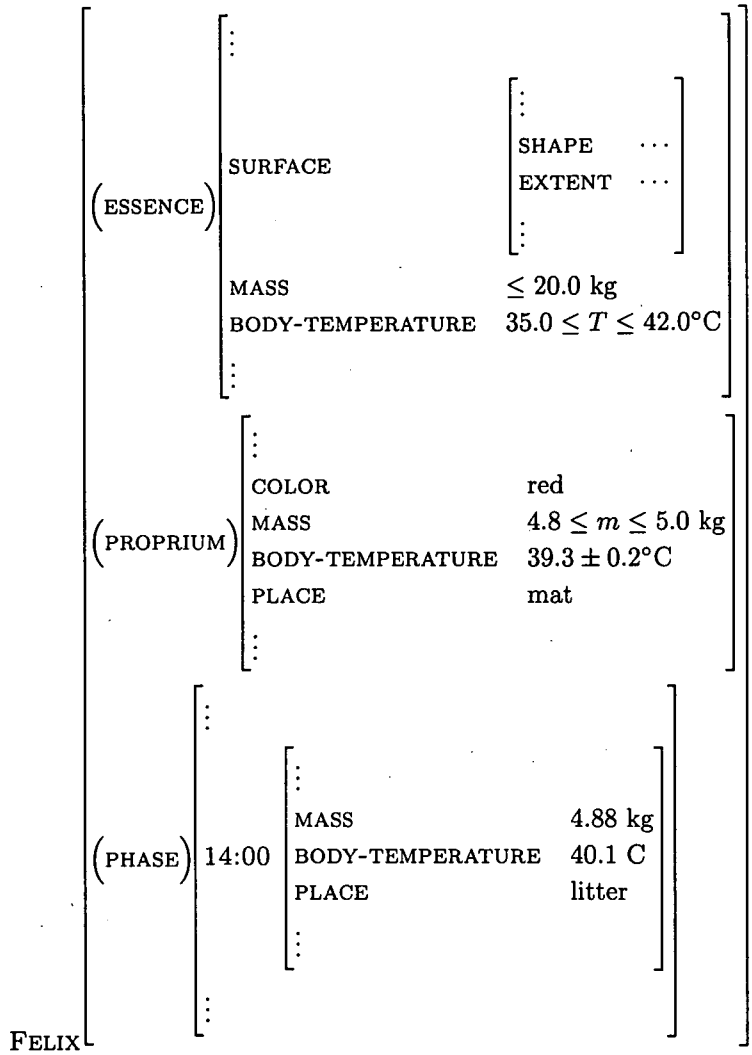
3.4.3 Level III. (individual) concepts

An individual concept describes a specific individual, for instance, a specific cat, say, Felix. The essential features of the individual concept FELIX will be the most specific essential features of the concepts above it, which it cannot overwrite.

Propria attributed to FELIX are *double-layered*: part of them is inherited with default inheritance from the general concept immediately above it[5], and the rest of them are propria that are *individual propria*, characterizing only FELIX. For example, if Felix spends most of his time on the mat in front of the door, then this will be an individual proprium characterizing his individual concept — but not the general concept of CAT.

Finally, Felix might be in specific *phases*, as well, and the values for these phases can contradict both his individual propria and his general propria. Phases, however, can even become individual propria of Felix with time (and if it happens for a substantial amount of cats, a proprium assigned to the general concept of cat may also undergo change). According to the description below, for example, at 14:00, Felix was lying in his litter with a fever — based on his body-temperature.

[5]Thus Felix may overwrite these general propria.



4 A possible formalization

As can be seen from the previous discussion, our model of ontology builds basically on graphs. Consequently, the ontology can be described using a graph description language. In our preliminary work at describing concepts of common sense, we began using AVM, which are a common tool in linguistics. However, this choice is, as yet, arbitrary, and basically any graph description language would be suitable.

An issue that can be raised is how our chosen language (in this case, AVM's) relates to description logics — a common tool used in ontologies.

Comparing DL and our model, one issue that arises is that while even the most basic Description Logics include the negation or complement operator, implement-

ing negation is not so straightforward in our model. AVM's used in linguistics usually prohibit negation and disjunction (however, see e.g., [6], [11]), and it may be argued that these operators are not needed when creating common-sense ontologies.

Another difference that sets our model and DL apart is that Description Logics that do not use role chains have a considerable deficiency in describing some complex concepts that make reference to the token identity of two concepts standing in a certain relation to the given concept. To give a very simple example, the mother-in-law of a man is identical to the mother of his wife. This can only be translated to a DL if it includes role-value-maps and the concept agreement operator (cf. [1]). However, these versions of DL are only decidable if they restrict the roles in the role-value-maps to functional roles. (R_F is a functional role iff $\{(a,b),(a,c)\} \subseteq R_F \rightarrow b = c$.) Models using AVM's usually assume attributes to be functional attributes, but when building an ontology, this restriction may cause some problems. The generic and mereological relations can hardly be defined as functions, and the description of some domains could easily need some non-functional relations as well.

On the other hand, the generic and mereological relations are special relations that have a distinguished role in ontologies, not to mention the fact that the generic relation is not part of the description language, but is a second-order relation between concepts.

The issue of formalization, as can be seen, is not yet fully resolved, and further research is necessary in this area. Currently, AVM's seem a suitable candidate for the role of representing dependency relations in general.

5 Summary

In this study, we have — rather roughly — presented an architecture for a general ontology. The central concept of this architecture is the dependency relation between types and the dependency graph representing it. Type relations like this constitute the horizontal structure of the ontology. The highest and most abstract level of the vertical organization of ontology comprises descriptions of "strong", logical–metaphysical kinds of dependencies. These graphs define few, but very general constraints on possible beings without tolerating exceptions. Weaker ("profession-dependent") modalities, as well as propria characterizing a given type in our world appear on the next level. Finally, on the level of individual concepts, values that have so far been underspecified will receive specification.

References

[1] Baader, Franz, Calvanese, Diego, McGuinness, Deborah L., Nardi, Daniele, and Patel-Schneider, Peter F., editors. *The Description Logic Handbook.* Cambridge University Press, 2003.

[2] Fine, Kit. Part–whole. In Smith, Barry and Smith, David Woodruff, editors, *The Cambridge Companion to Husserl.* Cambridge University Press, 1995.

[3] Gärdenfors, Peter. *Conceptual Spaces: The Geometry of Thought.* The MIT Press, 2000.

[4] Gruber, Thomas. Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[5] Husserl, Edmund. *Logische Untersuchungen*, volume I.–II. Max Neimeyer: Halle, 1900–1901.

[6] Karttunen, Lauri. Features and values. pages 28–33. Association for Computational Linguistics Morristown, NJ, USA, 1984.

[7] Katz, J.J. and Fodor, J.A. The Structure of a Semantic Theory. *Language*, 39(2):170–210, 1963.

[8] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L. The WonderWeb Library of Foundational Ontologies: Preliminary Report. `http://www.loa-cnr.it/Papers/DOLCE2.1-FOL.pdf`, 2003.

[9] Simons, Peter. Particulars in particular clothing: Three trope theories of substance. In Laurence, Stephen and Mcdonald, Cynthia, editors, *Contemporary Readings in the Foundations of Metaphysics*, chapter 4, pages 364–385. Blackwell, 1998.

[10] Stalnaker, Robert. Antiessentialism. *Midwest Studies of Philosophy*, 4:343–355, 1981.

[11] Wedekind, Jürgen. Classical logics for attribute-value languages. pages 204–209. Association for Computational Linguistics Morristown, NJ, USA, 1991.

Acta Cybernetica 18 (2007) 355–362.

Conversion of continuous speech sound to articulation animation as an application of visual coarticulation modeling

Gergely Feldhoffer* and Tamás Bárdi*

Abstract

A voice to facial animation conversion system is presented in this paper. In particular the temporal structure of the multimodal speech is discussed. Mutual information and neural network training is used to estimate the optimal temporal scope for audio to video conversion.

Keywords: Voice to animation, visual coarticulation, mutual information

1 Introduction

The most common form of the language is the personal talk which is an audiovisual speech process. Our research is focused on the relation of the audio and the visual part of talking to build a system converting voice signal into face animation. A voice to animation conversion system (VACS) targets hearing impaired persons to help them understand voice only communication channels. A VACS get a speech signal as input, and produce a face animation which is understandable for persons who can lip-read. This task is similar to speech inversion which tends to extract information from speech signal about the state sequence of the speech organs. However, speech inversion aims to reproduce every speech organ to exactly the same state as the speaker used his organs, with every speaker dependent property. VACS is different, the target is to produce a lip-readable animation which depends only on the visible speech organs and not depends on the speaker dependent features of the speech signal.

Recent research activities are on speech signal processing methods specially for lip-readable face animation [7], face representation and controller methods[8], and more natural face animation systems [3]. In this paper a working system is presented focusing on the temporal structure of the audiovisual speech process.

VACS are not speech recognition systems, the target is to produce an animation without recognizing any of the language layers as phonemes or words, as this part

*Pázmány Péter Catholic University, Faculty of Information Technology, Budapest, Hungary, E-mail: {flugi, bardi}@itk.ppke.hu

of the process is left to the lip-reader. Because this, our VACS uses no phoneme recognition, furthermore there is no classification part in the process. This is the continuous VACS, avoiding any discrete type of data in the process. Discrete VACS are using visemes as the visual match of phonemes to describe a given state of the animation of a phoneme, and using interpolation between them to produce coarticulation.

Training a continuous VACS needs audio-video data pairs. Since plenty of speech audio databases exist but only a few audiovisual ones, building a continuous VACS means building a multimodal database first. A discrete VACS is a modular system, it is possible to use existing speech databases to train the voice recognition, and separately train the animation part on phoneme pairs or trigraphs[1]. So continuous VACS needs a special database, but the system will handle energy and rhythm naturally, meanwhile a discrete VACS has to reassemble the phonemes into a fluid coarticulation chain of viseme interpolations. Let we call the overall time of a coarticulation phenomena as temporal scope which means that the state of the mouth is depending on this time interval of the speech signal. In continuous VACS the calculation of a frame is based on this audio signal interval. In discrete VACS the visemes and the phonemes are synchronized and interpolation is applied between them, as it is popular in text to visual speech systems[5]. Figure 1 shows this difference.

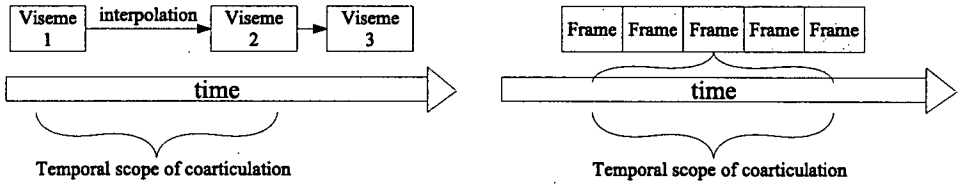


Figure 1: Temporal scope of discrete and continuous VACS

In this article we will describe a continuous VACS, and show how the system handles the visual coarticulation. We will show that using the average phoneme length as the length of temporal scope is a confirmable choice.

2 Database

The database for continuous VACS contains audio and video feature vector pairs. Basically it is a preprocessed multimedia material specially to use it as training set for neural networks. For this purpose the data should not contain strong redundancy for optimal learning speed.

2.1 Audio

The feature extraction starts with windowing. The length of a window depends on the frequency of the video camera which is 25 fps in this case, this means 40 ms

long windows. For temporal scope estimations we used 1 ms window step, and 40 ms window step for system training. Preemphasis is used, and FFT after Hamming window. Radix-2 FFT was applied for CPU efficiency, so the first 2^n element of the window is processed. The spectrum is mel-scaled to 16 bands, and logarithm and DCT is applied. The result is the mel-scaled cepstrum, the MFCC.

2.2 Video

For video processing we used two methods. Both methods are based on video recording of a speaker and feature tracker applications. The first method is based on markers only which are placed around the mouth. The markers were selected as a subset of the MPEG-4 face description standard. Tracking the markers is a computer aided process; a 98% precise marker tracker algorithm was developed for this phase. The mistakes were corrected manually. The marker positions as a function of time were the raw data which was normalized by control points as the nose to eliminate the motion of the whole head. This gives a 30-36 dimensional space depending on marker count. This data is very redundant and high dimensional, it is not suitable for neural network training, so PCA was applied to reduce the dimensionality and eliminate the redundancy. PCA can be treated as lossy compression as only the first 6 parameters were used for training. Using only 6 coefficients can cause about 1 pixel error on PAL screen which is the precision of the marker tracking.

The second method uses only 2 markers but uses color information of the mouth to extract markers which can not be painted as the inner contour of the mouth. This technique is still under development. In this paper the results were measured on video data which was extracted by the first method.

2.3 Training

The synchrony of the audio and video data is checked by word "papapa" in the beginning and the end of the recording. The first opening of the mouth by this bilabial can be synchronized with the burst in the audio data. This synchronization guaranties that the pairs of audio and video data were recorded in the same time. For the best result the neural network has to be trained on multiple windows of audio feature vectors where the window count have to be chosen based on the optimal temporal scope.

3 Temporal structure

To achieve the best results a good estimation of temporal scope is needed. Using a too short temporal scope can cause losing information about coarticulation. Using a too long temporal scope results longer training time without any quality improvement since the training will calculate with data which is independent from

the actual state. In this section the method of mutual information estimation will be shown as a possible solution for the question.

3.1 Mutual information

The mutual information is (1):

$$MI_{X,Y} = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

Mutual information is high if knowing X helps to find out what is Y, and it is low if X and Y are independent. To use this measurement for temporal scope the audio signal will be shifted in time compared to the video. If the time shifted signal has still high mutual information, it means that this time value should be in the temporal scope. If the time shift is too high, mutual information between the video and the time shifted audio will be low due to the relative independency of different phonemes.

Using a and v as audio and video frames:

$$\forall \Delta t \in [-1s, 1s] : MI(\Delta t) = \sum_{t=1}^{n} P(a_{t+\Delta t}, v_t) \log \frac{P(a_{t+\Delta t}, v_t)}{P(a_{t+\Delta t})P(v_t)} \tag{2}$$

where $P(x,y)$ is estimated by a 2 dimensional histogram convolved with Gauss window. Gauss window is needed to simulate the continuous space in the histogram in cases where only a few observations are there. Since audio and video data are multidimensional and MI works with one dimensional data, all the coefficient vectors were processed, and the results are summarized.

The channels were calculated by Independent Component Analysis (ICA) to keep down the interchannel dependency. The 16 MFCC channel was compressed into 6 independent component channels. The 6 PCA channels of video information was transformed into a ICA based basis. Interchannel independency is important because the measurement is the sum of all possible audio channel – video channel pairs, and we have to prove that each member of mutual information sum is not from the correlation of different video channels or different audio channels which would cause multiple count of the same information.

Since mutual information is a commutative, 6 x 6 estimations gives 15 different pairs.

Figure 2 shows the result of the estimation. Certain asymmetry can be observed in the sum of mutual information curves of all channel pairs of audio and video data. This measurement was done on a recording which aimed deaf people for lip-reading. This is a special situation; the speech speed is decreased to 5-6 phonemes per second. This gives an average phoneme length of 200 ms. As it can be seen on the figure, there is a high mutual information at 200 ms in the future of the voice, but a relatively low value in the past. This result shows that the visible speech organs are preparing for the next phoneme during the visual coarticulation

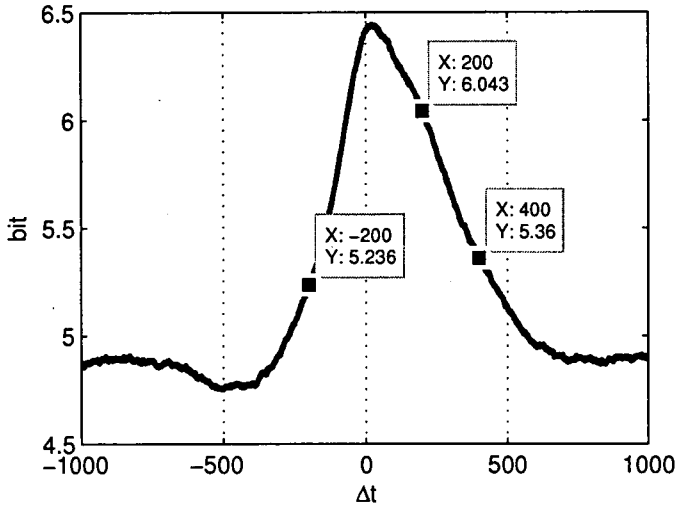


Figure 2: Sum of $MI(\Delta t)$ results of all channel audio-video pairs (6 x 6 : 15 pairs). Positive Δt means voice in the future was measured to the video frame in $\Delta t = 0$. The unit of time is millisecond.

while the speech audio signal is not changing. If both modalities would be changing together, there would be no asymmetry in mutual information.

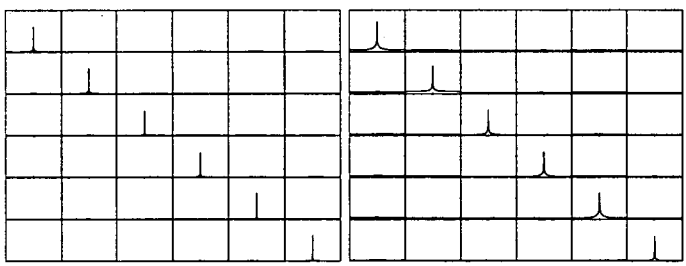


Figure 3: Interchannel $MI(\Delta t)$ results show independency in audio (left) and video (right) channels. The scaling of the curves are Δt = -1..1 in seconds on x axis, and 0..10 range in bits on y axis

On Figure 3 can be seen the independency of the channels. A channel with itself produces high mutual information in $\Delta t = 0$ because of equality. Short rising and decreasing phases can be observed in both modalities, much shorter than on Figure 2, however video data shows longer window of autocorrelation. This difference between audio and video data is partly because video information is from a 25fps recording which is 40ms of window length but the audio information was measured on every milliseconds, so video data was interpolated to fit to the audio data, and

possibly partly because of the difference between invisible and visible speech organs, this question is a part of our future work.

3.2 Network training

In practical way the measurement of the temporal scope is to estimate it with training efficiency. Efficiency is measured in this case by training error after a given epoch number. The same data were trained with different window counts, and after 10.000 epochs the training error was compared. Training error means the average difference of the network's output and target values in the training set. Using the training error of single frame training as 100%, we found that training errors are nearly linearly decreasing to 50% at 200ms and stay around 50% (even higher due to the increased difficulty and fixed epoch count) if the scope is increased further. See Figure 4. This confirms in practice the mutual information measurement.

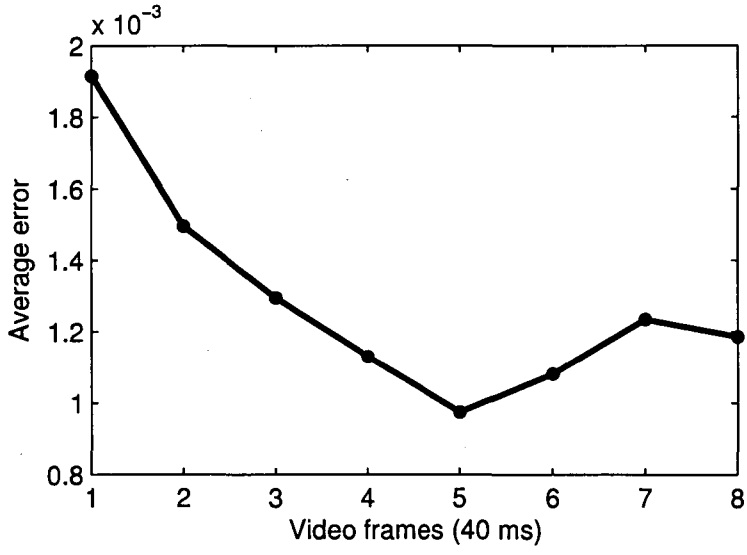


Figure 4: Training errors of different temporal scopes. The error is given in neural networks training data which is normalized to [-1..1] interval.

3.3 Visual coarticulation model

As mutual information estimation resulted that any given state of the video data stream can be calculated fairly on a defineable relative time window of the speech signal. This model predict that the transient phase of the visible speech can be calculated in the same way as in the steady phase as Figure 1 shows. This model gives a prediction about asymmetries in the multimodal speech process. This asymmetry can be explained with mental predictivity in the motion of the facial muscles to

fluently form the next phoneme. This explanation needs more proof, and it is an important part of our future work.

4 Results

The described modules were implemented and trained. The system was measured with a recognition test with deaf people. To simulate a measurable communication situation, the test covered numbers, names of days of the week and months. As the measurement aimed to tell the difference between the VACS and a real person's video, the situation had to be in consideration of average lip-reading cases. As we found [4] deaf persons recline upon context more than hearing people. In the cases of numbers or names of months the context defines clearly the class of the word but leave the actual value uncertain. During the test the test persons had to recognize 70 words from video clips. One third of the clips were original video clip from the recording of the database, other one third were output of the VACS from audio signals and the remaining one third were synthesized video clips from the extracted video data. The difference between the recognition of real recording and the face animation from the extracted video data gives the recognition error from the face model and the database, as the difference between animations from video data and audio data gives the quality of the audio to video conversion. Table 1 shows the results.

Table 1: Recognition rates of different video clips.

Material	Recognition rate
original video	97%
face model on video data	55%
face model on audio data	48%

The results show that our VACS have satisfactory precision in audio to video conversion, but the face model has to be more fine. As it was mentioned, a new video feature extraction method is in progress.

The system uses 200 ms temporal scope. It was showed that this time interval is confirmable with both mutual information estimation and neural network training experiments.

A visual coarticulation model was introduced based on the results of mutual information estimation.

5 Acknowledgements

We are thankful for György Takács and Attila Tihanyi for the guidance and useful remarks. Thanks for the SINOSZ (National Association of the Deaf and Hard of

Hearing in Hungary) and all the hearing impaired people who lent a helping hand in testing and development advices.

References

[1] B. Granström, I. Karlsson, K-E Spens *SYNFACE - a project presentation*, Proc of Fonetik 2002, TMH-QPSR, 44: 93-96. 2002.

[2] M. Johansson, M. Blomberg, K. Elenius, L.E.Hoffsten, A. Torberger *Phoneme recognition for the hearing im-paired*, TMH-QPSR. vol 44 Fonetik pp. 109-112, 2002.

[3] E. Sifakis, A. Selle, A. Robinson-Mosher and R. Fedkiw *imulating Speech with a Physics-Based Facial Muscle Model*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA), pp. 261-270, 2006.

[4] Gy. Takács , A. Tihanyi, T. Bárdi, G. Feldhoffer, B. Srancsik *Speech to Facial Animation Conversion for Deaf Customers*, Proceedings of EUSIPCO Florence Italy, 2006.

[5] M. Cohen and D. Massaro *Modeling coarticulation in synthetic visual speech*, Computer Animation 93. Springer-Verlag, 1993.

[6] P. Kakumanu and R. Gutierrez-Osuna and A. Esposito and R. Bryll and A. Goshtasby and O. Garcia *Speech Driven Facial Animation*, Proc. of the Workshops on Perceptual/Perceptive User Interfaces (PUI), Orlando, FL, PUI 2001.

[7] P. Kakumanu,A. Esposito, O. N. Garcia, R. Gutierrez-Osuna *A comparison of acoustic coding models for speech-driven facial animation*, Speech Communication 48 pp 598-615, 2006.

[8] P. Scanlon, G. Potamianos, V. Libal, and S. M. Chu *Mutual Information Based Visual Feature Selection for Lipreading*, in Proc. of ICSLP 2004, South Korea, 2004.

CONTENTS

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Csirik János