

# Zpravodaj Československého sdružení uživatelů TeXu

---

Miroslav Hrad; Petr Sojka  
Automatizace sazby a skenování formulářů

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 12 (2002), No. 3-4, 123–139

Persistent URL: <http://dml.cz/dmlcz/149896>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2002

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

---

---

# Automatizace sazby a skenování formulářů<sup>1</sup>

MIROSLAV HRAD, PETR SOJKA

V článku je popsána technologie SCAT pro přípravu a skenování formulářů, která byla v minulých třech letech použita na Masarykově univerzitě a v dalších organizacích na zpracování téměř statisíce formulářů vysazených  $\text{\TeX}$ em a zpracovaných za použití nástrojů otevřených systémů. Jsou diskutovány prvky navrženého systému směřující k minimální chybovosti (použití čárových kódů), rychlosti, anonymitě, maximální bezpečnosti a automatizaci zpracování.

*Klíčová slova:* SCAT, skenování, rozpoznávání, formuláře, čárové kódy,  $\text{\TeX}$ , Linux, automatizace testů

## 1. Úvod

Vkládání dat člověkem do počítače je činnost častokrát časově náročnější než jejich samotné počítačové zpracování. Ruční vkládání dat s sebou navíc přináší velké procento chyb a jejich odstraňování bývá často netriviální záležitostí. Nejinak je tomu i v případě vytváření, zpracování a vyhodnocování písemných testů.

### 1.1. Potřeba automatizace

Vytváření, zpracování a vyhodnocování testů je běžnou stereotypní prací všech, kteří zadávají písemné zkoušky na všech typech škol. Může se jednat jak o obyčejné zkoušení studentů – třeba na střední škole, tak o přijímací zkoušky na vysokou školu.

### 1.2. Východiska

Použití všech nabízených komerčních řešení dostupných na trhu selhávalo na nemožnosti přizpůsobení konkrétním požadavkům. Outsourcing každodenních potřeb se nejeví jako ideální stav. Vhodnost použití „in-house“ řešení, jak se ukázalo později, umožňuje *mnohem* vyšší komfort a flexibilitu vyhodnocení testů než řešení založené na uzavřených systémech.

V rámci série bakalářských a diplomových prací [4, 2] se postupně vyvíjely algoritmy a ověřovala se technika potřebná pro realizaci automatizace zpracování agendy přijímacího řízení nebo vyhodnocení písemných testů. Postupným

---

<sup>1</sup>Článek byl poprvé publikován ve sborníku Jan Kasprzak, Petr Sojka: SLT 2002, KONVOJ 2002, ISBN 80-7302-043-2, na stranách 11–26. Publikováno se souhlasem vydavatele a autorů.

vývojem a dlouhým testováním systému se nakonec ukázalo, že je možno vytvořit spolehlivý systém a technologii nevyžadující vysoce specializovaný a drahý hardware.

### 1.3. Členění článku

V oddíle 2 je definován proces vytváření a zpracování formulářů a jsou popsána východiska projektu. V sekcích 3 a 4 je diskutována problematika návrhu a sazby testů s použitím technologie čárových kódů. Oddíl 5 rozebírá algoritmy rozpoznávání symbolů a textů ve formulářích. V oddíle 6 je pak zmíněn způsob vyhodnocení a zajištění anonymity zpracování až do finálního zveřejnění výsledků testu.

## 2. Proces vytváření a zpracování formulářů

Technologie pro automatizované vyhodnocování testů s využitím skeneru s podavačem musí řešit tyto etapy:

1. analýza specifických požadavků projektu
2. vytvoření testů a jejich testování, sazba
3. návrh odpovědních formulářů
4. sazba a tisk odpovědních formulářů
5. distribuce a sběr testů a formulářů
6. skenování formulářů
7. vyhodnocení naskenovaných formulářů
8. vyhodnocení výsledků testu a jejich prezentace
9. vyhodnocení statistických parametrů testových otázek a příkladů
10. zatřídění formulářů se zadáními pro dlouhodobou archivaci.

### 2.1. Technologie SCAT

Při vzniku projektu SCAT se vycházelo z dosavadního způsobu vyhodnocování přijímacího testu na fakultu informatiky a pedagogickou fakultu MU v Brně. Byl používán výsledkový list, který měl dvě části. Část s identifikačními údaji a část s oválky pro odpovědi na otázky. Tyto dvě části byly oddělitelné a logicky je spojovalo pouze shodné, předem vygenerované číslo. Tento způsob byl použit právě z důvodu anonymity vyhodnocování.

### 2.2. Cíle a požadavky při návrhu systému SCAT

Pro projekt jsme si stanovili následující cíle a požadavky na systém:

- flexibilní vytváření vlastních testových formulářů,

- jednoduchá integrace vytvořených formulářů do systému zpracování,
- bezchybné rozhodování – zpracování testů,
- zvýšení rychlosti zpracování,
- vytvoření logistiky zpracování pro přijímací zkoušky,
- zaručení bezpečnosti – neovlivnitelnosti výsledků,
- standardizovaný výstup, aby bylo možné výsledky exportovat do současných systémů,
- jednoduché ovládání,
- otevřenost systému vůči novým prvkům a přístupům,
- minimální hardwarové požadavky,
- případná přenositelnost na jiné systémy, standardně bude použit operační systém Linux.

Požadovali jsme systém, který umožní vyhodnotit *více správných odpovědí*, tedy bude zpracovávat každý vyhodnocovací prvek – čtvereček a určí o něm jeho stav (vyplněn/nevyplněn) bez ohledu na ostatní odpovědi v dané otázce.

### 2.3. Výpočetní technika a software systému SCAT

Požadovali jsme, aby bylo v maximální míře využito všeobecně dostupného hardware a software, s maximálním podílem otevřeného software, což zajišťuje vysokou míru přenositelnosti a rozšiřitelnosti.

Využíváme běžných počítačů PC s operačním systémem Linux a skenerů běžné cenové kategorie. Nárazové nasazení (přijímací zkoušky) systému tedy nevyžaduje (s možnou výjimkou skeneru) vyšší finanční nároky.

Pro implementaci jádra systému SCAT jsme zvolili jazyk C. Ostatní komponenty (především logistika systému) je napsána ve skriptovacím jazyce Perl. Sazba všech výstupů včetně formulářů je realizována sázecím systémem  $\text{\TeX}$  a programem METAFONT.

## 3. Návrh vyhodnocovacího formuláře

Významným omezením některých softwarových produktů pro vyhodnocování testů je fixní množina předtisknutých formulářů – například Recognita Test. Naším cílem bylo navrhnout technologii dostatečně flexibilně, bez omezení na počet otázek, odpovědí, textů či způsob identifikace.

Při vyhodnocování písemných testů pro přijímací řízení je nutné maximálně eliminovat možnost záměrného zásahu do objektivnosti zjištěných výsledků, tedy mimo jiné dosáhnout anonymity. To lze zajistit například tím, že oddělíme odpovědní část formuláře od části identifikační. Obě části pak zpracováváme samostatně nebo identifikační část zpracováváme až po zpracování části odpovědní (ve stavu, kdy už nelze zjištěné výsledky ovlivňovat). Tento způsob zpracování

ovšem vyžaduje navrhnout formuláře tak, aby části byly jednoduše oddělitelné (lze zajistit třeba perforací formuláře), ale především, abychom byli schopni přiřadit výsledky z odpovědní části k identifikaci. Jedním ze způsobů, jak zajistit onu *logickou vazbu*, je obě části vybavit shodným číslem – *vazebním číslem*. Je zřejmé, že jednotlivé testové formuláře pak musí mít vždy vzájemně rozdílná vazební čísla. Z této nutnosti plyne potřeba generovat a tisknout jednotlivé formuláře, naopak není možné formuláře kopírovat. Vazební číslo je vhodné navrhnout tak, aby se dalo využít technologie čárových kódů [1], neboť rozpoznávání čárových kódů je při dodržení určitých podmínek výrazně méně chybové než technologie OCR<sup>2</sup>.

V případě, že není nutné zajišťovat anonymitu v průběhu vyhodnocování, a lze tedy zpracovávat formulář, který obsahuje jak odpovědi, tak identifikaci, pak mohou být testové formuláře vzájemně shodné a lze vytvářet jejich kopie a ty potom zpracovávat. Takový formulář využijeme třeba při běžném písemném zkoušení. Při vytváření nám odpadne potřeba vazebních čísel a také vyhodnocení je jednodušší, identifikace i odpovědi se zpracovávají zároveň – při jednom průchodu skenovacím zařízením.

## 4. Sazba formulářů testů

Na rozhodnutí, jakým způsobem a kterým sazečím systémem budeme vytvářet testové formuláře, měly vliv následující potřeby:

- automatické generování formulářů, jejich čísel a čárových kódů,
- možnost převedení informací z výstupu na souřadnice pro systém,
- nezávislost přípravy formulářů na operačním systému.

Sazba čárových kódů *závisí* na použitém výstupním zařízení a tiskové technice (laserová tiskárna, ofsetový tisk) a je třeba provést *kalibraci* (bar correction) tlouštěk čar sázených čárových kódů.

### 4.1. Návrh testů

Nejsnazším způsobem je test navrhnout jako sadu příkladů s možností výběru z  $N$  správných řešení, „multiple-choice test“,  $N$  obvykle 3–5. Pro minimalizaci možnosti opisování vedle sebe sedících zkoušených je vhodné mít testy generované zcela individuálně [7], nebo mít alespoň několik variant testu s vzájemně permutovaným pořadím otázek pro znesnadnění podvádění.

Detailní rozbor návrhu obsahu testů je mimo rozsah tohoto příspěvku, každopádně je vhodné softwarově podporovat alespoň statistické vyhodnocení složitosti jednotlivých příkladů z realizovaného testu.

---

<sup>2</sup>Optical character recognition

## 4.2. Generování testů

Až na výjimky je třeba připravit různé, případně i plně individuální verze testových odpovědních formulářů nebo i samotných testů. Zejména pro možnost vysázení individuálních testů (databázového publikování [11]) je ideální a plně se osvědčil sázeací systém  $\text{T}_{\text{E}}\text{X}$ <sup>3</sup>.  $\text{T}_{\text{E}}\text{X}$  dokonce může sám permutovat příklady [7] a zapisovat soubory pro vyhodnocení, ale toto lze většinou realizovat místo složitěho makroprogramování externím skriptem ve skriptovacím jazyce (obvykle Perl).

Ani sazba čárových kódů EAN nepředstavuje problém, existují například makra popsaná v článku [6]. Pro námi požadovaný rozsah kódovaných čísel postačuje například čárový kód EAN-13, používaný i v systému ISBN.

Při návrhu formuláře musí existovat způsob, jak komunikovat prvky formuláře a jejich umístění vyhodnocovací části systému. V otevřeném a dobře dokumentovaném systému, jako je  $\text{T}_{\text{E}}\text{X}$ , to není problém.

### 4.2.1. Vložení vlastní informace pro generování souřadnic

Do zdrojového souboru pro sazbu  $\text{T}_{\text{E}}\text{X}$ em můžeme vložit příkaz `\special` [3, kapitola 21]. Tento příkaz, jehož argument nebude sázen, je beze změny přenesen do souboru DVI.

Pro automatický vznik souřadnic a pro správné ohodnocení odpovědních čtverečků potřebujeme do zdroje pro sazbu vložit informaci o počtu odpovědí v otázce. Tedy např. `\special{pocet_ctv 3}` znamená, že budou následovat otázky se třemi odpovědními čtverečky –  $A, B, C$ . Tuto informaci musíme vložit před každou změnu počtu odpovědí. Standardně je možné nastavit až deset odpovědí v rámci jedné otázky.

### 4.2.2. Soubor DVI

Výstupní soubor  $\text{T}_{\text{E}}\text{X}$ u je binární soubor DVI<sup>4</sup>, který čtou ovladače jednotlivých výstupních zařízení, například ovladač obrazovky, ovladač tiskárny. Jeden z takových ovladačů je program `dvitype`, který převede všechny informace z binárního souboru do čitelné podoby – textového souboru, který obsahuje všechny údaje o pohybu bodu sazby a o kódech jednotlivých znaků v relativně srozumitelné podobě.

Pokud se provádí sazba znaku například povel `set_char_i`, znak se umístí tak, aby se jeho referenční bod kryl s bodem sazby a po vykreslení znaku se bod sazby posune o šířku znaku doprava. K tomu ovladač potřebuje znát šířky všech znaků, které jsou uloženy v TFM. Ovladači může postačit formát PK, protože tam

---

<sup>3</sup>Používáme distribuci  $\text{T}_{\text{E}}\text{X}$ Live

<sup>4</sup>DVI – Device Independent – formát nezávislý na zařízení

jsou údaje o šířkách znaků (nikoli však o výškách a hloubkách) duplikovány se stejnou přesností jako v TFM.

#### 4.2.3. Shrnutí vzniku systému souřadnic

Zaměřovací prvky, zaškrťovací prvky a prvky pro identifikaci vytvoříme v METAFONTu, jako METAFONTové znaky. Na obrázku 1 jsou zobrazeny připravené METAFONTové znaky. Ty pak můžeme jednoduše používat pro tvorbu nových testových formulářů. Příklady sazby formuláře jsou uvedeny na straně 136 a straně následující. Pomocí sázecího systému T<sub>E</sub>X získáme obraz vysázeného formuláře ve formátu DVI. Z formátu DVI jsou pak zjistitelné pozice sazby na straně s využitím programu `dvitype`.



Obrázek 1: Prvky v METAFONTu

Například víme, že zaškrťovací prvek je v METAFONTu reprezentován jako znak číslo dvě v daném fontu. Zvolíme úroveň výpisu programu `dvitype` alespoň tři a ve výpisu potom hledáme řetězec „`setchar2`“, např.:

```
4654: setchar2 h:=2660904+792488=3453392, hh:=219
```

Tímto získáme horizontální souřadnici  $h$  v kartézském systému souřadnic  $(h, v)$ , resp. přímo souřadnici  $hh$  v pixelech. Souřadnici  $v$ , resp. souřadnici  $vv$  v pixelech, najdeme ve výpisu jako poslední změnu vertikální pozice před místem sázení znaku „`setchar2`“, např.:

```
4649: push level 6: (h=--2797019, v=15156135, w=0, x=0,
y=0, z=0, hh=--177, vv=960)
```

Takto můžeme programově získávat údaje z formuláře do konfiguračního souboru skenovacího programu.

#### 4.2.4. Specifikace požadavků tisku

Protože jde o relativně velké tiskové objemy, je třeba si před hromadným tiskem odsouhlasit přesný seznam formulářů a jejich parametrů. Ukázalo se nanejvýš vhodné, aby se z jednoho a téhož souboru tiskly jak zkušební sada formulářů, tak jejich seznam s jejich počty. Vše je zadáno v textovém souboru, ze kterého se údaje čtou pro různé výstupy (stačí příkazem `\let` změnit před načtením chování hlavního makra).

```

\datum{červen 2002}      % nastavení implicitní hodnoty data
\setcounter{nanswers}{4} % počet možných odpovědí
\vartext{Fakulta X}     % variabilní text
\idtext{}               % identifikační text
\id = 0                 % počáteční nastavení čítače čísel testů

```

```

\obor{Písemný test zaměřený na ověření obecných
studijních předpokladů}
\studium{\TeX u a Linuxu}
\sada 10 1 99 1000      % počet testů, kód, verze, typ studia
\idtext{20.\,6.\,2002}

```

```

\obor{Přijímací písemná zkouška na obor anglický jazyk
pro základní školy}
\studium{bakalářské studium kombinované}
\sada 20 1 76 300      % počet testů, kód, verze, místnost

```

...

Na straně 136 najdete příklady vysázených formulářů.

## 5. Skenování

Úspěšnost skenování je vysoce závislá na schopnosti ovládat a využít technické parametry dostupné techniky a doladit parametry použitého software [9, 10]. Ceny specializovaných vysokokapacitních skenerů dosahují statisícových částek, a proto jsem se zaměřili na ověření možnosti využít skenerů cenových kategorií v řádu desetitísiců korun českých.

### 5.1. Skenovací zařízení

Pro vývoj našeho systému jsme použili skenovací zařízení firmy Hewlett Packard HP 6100 C s automatickým podavačem a skenovací zařízení HP 6350 C s automatickým podavačem ScanJet ADF. Zařízení byla připojena přes SCSI řadič.

Podavač skeneru HP 6100 C má kapacitu maximálně 50 listů. Rychlost podávání a skenování jedné černobílé strany při rozlišení 300 dpi je 27–30 sekund. Předloha ani snímač se ve skeneru nepohybují. Pohybuje se čtecí paprsek, úzký paprsek světla usměrňovaný soustavou zrcadel. Množství světla, které se od snímaného předmětu odrazilo, se měří a digitalizuje.

Skener HP 6350 C je dodáván s automatickým podavačem dokumentů na 25 stran. Rychlost podávání a skenování jedné černobílé strany při rozlišení 300 dpi je 17–18 sekund. Komunikační rozhraní je USB nebo SCSI.

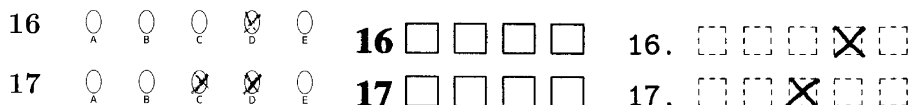
Skener HP 6350 C (obdobně jako jeho předchůdce) ke snímání používá pouze jednu řadu CCD senzorů. Zásadní rozdíl oproti skeneru HP 6100 C je v použitém



podavači. Automatický podavač ScanJet ADF, jenž je dodáván ke skeneru HP 6100 C, nejprve papírový dokument zavede do skeneru a potom skenuje, zatímco u skeneru HP 6350 C je dokument skenován v průběhu vedení podavačem. Tato změna přinesla výhodu ve formě zrychlení podávání, nicméně ohýbání dokumentu se děje nad čtecím prostorem a proto je vhodné pravidelně po několika stovkách naskenovaných listů toto místo vyčistit. Pro testové formuláře je vhodné právě z tohoto důvodu používat kvalitní papír, který se otírá minimálně a je bez kazů.

## 5.2. Varianty zaškrťovacích prvků

Na obrázku 2 jsou zobrazeny příklady zaškrťovacích prvků – mohou být použity různé tvary. Oválky byly voleny pro jednodušší vyplňování. Klasickým tvarem pro zaškrťování je čtvereček. Pro snížení tmavosti strany, resp. pro zvýraznění vybraných odpovědí, byly navrženy čtverečky kreslené přerušovanou čarou.



Obrázek 2: Varianty zaškrťovacích prvků

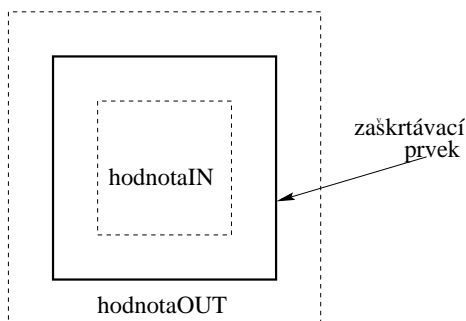
## 5.3. Čím zaškrťávat, jak vyhodnocovat

Ideálním prostředkem pro zaškrťování jsou tenké fixy, propisovací tužky a pera v barvách černá, modrá, červená, zelená. Program je navržen tak, že rozpoznává zaškrtnutí i tenkou obyčejnou tužkou (pentelkou). Lze tedy vyplňovat téměř jakýmkoliv běžným prostředkem pro psaní po papíru s výjimkou nestandardních psacích pomůcek jako jsou žlutý či oranžový zvýrazňovač.

Významnou otázkou bylo, co vlastně považovat za vyplněný (zaškrtnutý) čtvereček. Při návrhu a měření schopností jsme se dostali na hranici, kdy je rozpoznána i drobná tečka, malá čárka a podobně. Nicméně již ve fázi testování se ukázalo, že některé odpovědi jsou takto drobně označeny jenom proto, že nad nimi student přemýšlel, ale nelze je považovat za zaškrtnuté.

Hodnotová funkce  $f$  je počítána z obou oblastí *hodnotaIN* a *hodnotaOUT* tak, jak je zobrazeno na obrázku 3. Majoritní úlohu ve výpočtu hodnotové funkce má oblast *hodnotaIN*, takže i stejná značka (čárka, tečka, malý křížek) má v oblasti bližší ke středu větší hodnotu, než kdyby byl umístěn na okraji. Obor hodnot funkce  $f$  je  $(0, \dots, 5600)$ , kde 0 je nevyplněný a 5600 je zcela vyplněný čtvereček. Běžně, křížkem vyplněný čtvereček má hodnotu cca 1600–1800.

Drobná tečka cca 400. Současná *prahová hranice* pro rozhodnutí je nastavena na 960.



Obrázek 3: Výpočet hodnotové funkce

#### 5.4. Posun a pootočení při tisku a skenování

Posun a pootočení naskenovaného obrazu vůči očekávaným pozicím je jednou z nejmarkantnějších nepřesností, které musíme řešit. Z převodu DVI očekáváme přesné pozice umístění vytisknutých prvků na papíru, ty se však v závislosti na tiskárně, kopírovacím stroji i skeneru liší. Posun i pootočení je obvykle způsobeno podavačem, většinou se jedná o několik milimetrů až centimetrů. Právě řešení tohoto problému je zcela nezbytné k dosažení potřebné přesnosti.

#### 5.5. Detekce a korekce změn velikostí

Při tisku a zvláště při kopírování testových formulářů může docházet na některých zařízeních k zvětšení nebo zmenšení tiskového obrazu vůči standardním rozměrům daným v souboru DVI, resp. vůči originálu při kopírování. Změna je tak ve shodném poměru v ose  $x$  i  $y$ .

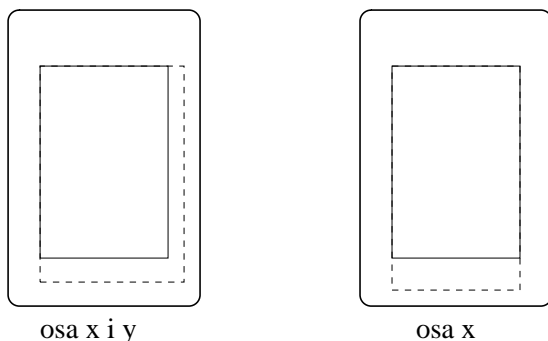
$$\frac{x'}{x} = \frac{y'}{y}$$

Může taktéž docházet ke změně velikosti pouze v ose  $y$ , tedy k jakémusi protažení obrazu. Tento defekt se projevuje právě u použitého skeneru HP 6350 C. Skener má podavač, jenž při průchodu papíru zároveň daný dokument skenuje. Při nepřesném podávání tak dochází k nechtěnému protažení, jak je ukázáno na obrázku 4.

$$\frac{x'}{x} = 1$$

$$\frac{y'}{y} = k, \text{ kde } k \text{ je např. } 1,007$$

Při větším množství zpracovávaných testů (více než několik tisíc) může docházet i k následnému opotřebení zařízení podavače. Vodící kolečka pro podávání papíru vlivem opotřebení mírně zmenšují svůj průměr a načítání dokumentu tak trvá delší čas, což se projeví protažením naskenovaného obrazu ve směru osy  $y$ , a tím se tedy mění i  $k$ . Hodnota  $k$  se průběžně počítá ze zaměřovacích prvků.



Obrázek 4: Zvětšení v osách  $x$ ,  $y$  a pouze  $y$

## 5.6. Zaměřovací prvky

Zaměřovacími prvky mohou být buď zaměřovací křížky nebo čtverečky. Slouží k jednoznačnému a přesnému zaměření nasnímaného obrazu tak, aby mohly být přepočteny všechny souřadnice.

Nalezení prvků musí být bezchybné, jednoznačné a rychlé. Byly vyzkoušeny různé varianty, nicméně zaměřovací čtvereček je nejvíce odolný vůči různým nežádoucím poruchám snímání i nepřesnému zaostření. Na obrázku 5 jsou zobrazeny zvětšeniny zaměřovacích prvků tak, jak je nasnímá skener. Okraje prvků jsou nerovné, neostře. Vzhledem k tomu, že nalezení a kontrola nalezení černého čtverečku je i algoritmicky jednoduché, je v současnosti používán právě tento zaměřovací prvek.

Ze zaměřovacího prvku zjistíme *bod zaměření*, a to pro každý zaměřovací prvek. V našem případě obvykle používáme dva v protilehlých rozích testu, jak je vidět v příkladech na straně 136 a dalších. Ze zjištěných údajů a očekávaných údajů (vzniklých při převodu DVI na souřadnicový systém) vypočteme hodnoty zpětného posunutí, pootočení a zvětšení (respektive zmenšení). Tím z nasnímaného obrazu získáme očekávaný obraz.



Obrázek 5: Varianty zaměřovacích prvků

### 5.7. Identifikace pomocí osobního čísla a čárového kódu

Na obrázku 6 je zobrazena identifikační část formuláře, který byl použit pro přijímací zkoušky v roce 2000. Vyplňující vypíše své příjmení, jméno, adresu, rodné číslo a především své osobní číslo. Na identifikační části i na části odpovědní je vytisknut shodný čárový kód. Zpracování tohoto formuláře spočívá v ručním opísání osobního čísla studentů (6 znaků) a kódového čísla formuláře (4 znaky). Tato identifikační část je oddělená od odpovědní části perforací. Formulář se po vyplnění rozdělí na dvě samostatné části, které jsou propojeny pouze logickou identifikací – předem na obě části vygenerovaným shodným čárovým kódem. Zpracování jednotlivých částí pak probíhá odděleně, a tím tedy i anonymně.



Přijímací zkouška – písemná část

červen 2000



Údaje v části nad čarou vyplňte čitelně hůlkovým písmem, oddělte a ústřížek na pokyn odevzdejte.

Příjmení: ..... jméno: .....

adresa: .....

rodné číslo: UUUUUU / UUUU osobní číslo (viz pozvánka): UUUUUU 0001

Obrázek 6: Identifikace s využitím čárového kódu

### 5.8. Identifikace ve formě digitálního čísla

Příklad na obrázku 7 ukazuje zapsání UČO: 003908 a číslo písemky: 724 do formy digitálních čísel.

JMÉNO: \_\_\_\_\_

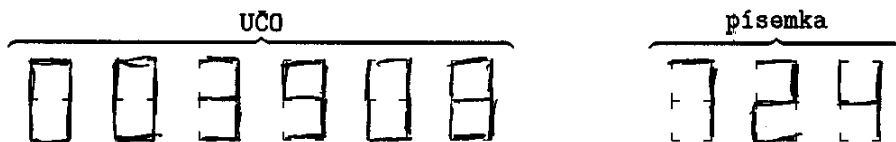
Č. PÍSEMKY: \_\_\_\_\_

UČO: \_\_\_\_\_

DATUM: \_\_\_\_\_

PŘEDMĚT: \_\_\_\_\_

Podle tohoto vzoru: 0:123456789 vyplňte následující údaje:



Obrázek 7: Identifikace zapsaná ve formě digitálního čísla

Ve výstupu zpracování digitálního čísla se může objevit i znak „-“, který znamená nevyplněné digitální číslo. Chyba je označena znakem „x“ a hlášením na standardní chybový výstup.

Příklad výstupu zpracování:

-50821,176, 1.A: 2.A: 3.C: 4.D: 5.D: 6.-: 7.C: 8.-: 9.B:  
10.ACD: 11.-: 12.D: 13.B: 14.AD: 15.D: 16.-: 17.B: 18.D:  
19.D: 20.-: 21.D: 22.ABCD: 23.-: 24.D: 25.-:

V reálném testování se jako nejpřesnější ukázal algoritmus pracující na základě hledání souvislostí mezi jednotlivými body (rohy) digitálního čísla. Je odolný vůči různým intenzitám vyplnění – od obyčejné tužky až po pero, propisku či tenkou fixu.

## 6. Vyhodnocení výsledků a kontroly

Pro nasazení našeho systému v reálném prostředí zpracování testů v přijímacím řízení bylo rozhodnuto, že budeme skenovat pouze část pro odpovědi. Část pro identifikaci bude zadávána ručně. Systém ukládá do souboru výsledky zpracování naskenovaných odpovědí, je schopen načíst vzorové odpovědi a uložit je do souboru. Na obrázku 8 je identifikační část označena zkráceně „ID.“, odpovědní část „ODP.“ a čárový kód jako „EAN“. Náš systém na základě zadání souboru výsledků a zadání souboru správných vzorových odpovědí vyhodnotí bodovou (procentuální) úspěšnost a je schopen uložit tyto body spolu s identifikační částí čárového kódu do souboru. Za správně zodpovězenou otázku se považuje shodnost všech odpovědních prvků (obvykle tří, čtyř, resp. pěti čtverečků) se vzorovým řešením a bude započtena 1 bodem, v opačném případě 0 body. Lze nastavit i jiné bodové ohodnocení odpovědí (třeba záporné body za chybné odpovědi).

System upozorňuje obsluhu na případné automaticky zjistitelné chyby, např. nesprávně naskenovaný formulář, nečitelný – poškozený čárový kód EAN, poškozené zaměřovací čtverečky a podobně. System umožňuje obsluze kontrolovat správnost rozpoznání odpovědí.

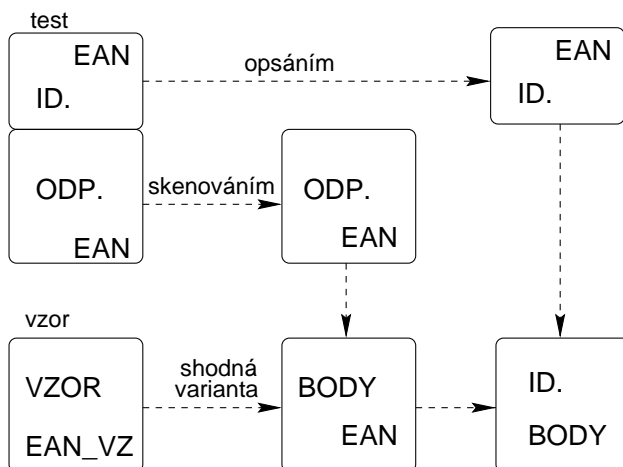
### 6.1. Testování v reálném prostředí

V letech 2000, 2001 i 2002 byl systém SCAT nasazen v reálném prostředí přijímacích zkoušek na vysokou školu. V roce 2001 byl i pro testovací účely využit jako prostředek vyhodnocení písemného zkoušení na vysoké škole. Každým rokem bylo vyhodnoceno cca 30 000 testových formulářů.

#### 6.1.1. Slabá místa systému

Časově nejnáročnější je proces samotného skenování, které je vždy omezeno rychlostí podavače skeneru. Vzhledem k cenové dostupnosti tohoto zařízení lze však použít skenerů několik a tím celý proces zpracování výrazně urychlit.

Z důvodu anonymity se nejčastěji používá pro účely přijímacích zkoušek systém, kde je fyzicky oddělena identifikační a odpovědní část. Ukázalo se, že ruční zpracování identifikační části testu obvykle nečiní žádné potíže. Náročné na logistiku může být na některých fakultách vyhodnocení relací různých oborových testů, toto však je zcela nezávislé na technologii zpracování odpovědí.



Obrázek 8: Logistika přijímacího řízení



Institut mezioborových studií  
Brno

## PŘIJÍMACÍ ZKOUŠKA

studijní program **Pedagogika**, obor **Sociální pedagogika**

ak. rok 2002/2003



0 143820 210515

Údaje v části „A“ (nad čarou) vyplňte čitelně hůlkovým písmem. Ústřížek na pokyn odtrhněte a odevzdejte.

Část „A“

Příjmení: ..... Jméno: .....

Rodné číslo: /  Registrační číslo:  **1051**



Část „B“

	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	21	<input type="checkbox"/>	<input type="checkbox"/>	41	<input type="checkbox"/>	<input type="checkbox"/>	61	<input type="checkbox"/>	<input type="checkbox"/>	81	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22	<input type="checkbox"/>	<input type="checkbox"/>	42	<input type="checkbox"/>	<input type="checkbox"/>	62	<input type="checkbox"/>	<input type="checkbox"/>	82	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23	<input type="checkbox"/>	<input type="checkbox"/>	43	<input type="checkbox"/>	<input type="checkbox"/>	63	<input type="checkbox"/>	<input type="checkbox"/>	83	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	24	<input type="checkbox"/>	<input type="checkbox"/>	44	<input type="checkbox"/>	<input type="checkbox"/>	64	<input type="checkbox"/>	<input type="checkbox"/>	84	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	25	<input type="checkbox"/>	<input type="checkbox"/>	45	<input type="checkbox"/>	<input type="checkbox"/>	65	<input type="checkbox"/>	<input type="checkbox"/>	85	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	26	<input type="checkbox"/>	<input type="checkbox"/>	46	<input type="checkbox"/>	<input type="checkbox"/>	66	<input type="checkbox"/>	<input type="checkbox"/>	86	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	27	<input type="checkbox"/>	<input type="checkbox"/>	47	<input type="checkbox"/>	<input type="checkbox"/>	67	<input type="checkbox"/>	<input type="checkbox"/>	87	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28	<input type="checkbox"/>	<input type="checkbox"/>	48	<input type="checkbox"/>	<input type="checkbox"/>	68	<input type="checkbox"/>	<input type="checkbox"/>	88	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	29	<input type="checkbox"/>	<input type="checkbox"/>	49	<input type="checkbox"/>	<input type="checkbox"/>	69	<input type="checkbox"/>	<input type="checkbox"/>	89	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	30	<input type="checkbox"/>	<input type="checkbox"/>	50	<input type="checkbox"/>	<input type="checkbox"/>	70	<input type="checkbox"/>	<input type="checkbox"/>	90	<input type="checkbox"/>	<input type="checkbox"/>
11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	31	<input type="checkbox"/>	<input type="checkbox"/>	51	<input type="checkbox"/>	<input type="checkbox"/>	71	<input type="checkbox"/>	<input type="checkbox"/>	91	<input type="checkbox"/>	<input type="checkbox"/>
12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32	<input type="checkbox"/>	<input type="checkbox"/>	52	<input type="checkbox"/>	<input type="checkbox"/>	72	<input type="checkbox"/>	<input type="checkbox"/>	92	<input type="checkbox"/>	<input type="checkbox"/>
13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	33	<input type="checkbox"/>	<input type="checkbox"/>	53	<input type="checkbox"/>	<input type="checkbox"/>	73	<input type="checkbox"/>	<input type="checkbox"/>	93	<input type="checkbox"/>	<input type="checkbox"/>
14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	34	<input type="checkbox"/>	<input type="checkbox"/>	54	<input type="checkbox"/>	<input type="checkbox"/>	74	<input type="checkbox"/>	<input type="checkbox"/>	94	<input type="checkbox"/>	<input type="checkbox"/>
15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	35	<input type="checkbox"/>	<input type="checkbox"/>	55	<input type="checkbox"/>	<input type="checkbox"/>	75	<input type="checkbox"/>	<input type="checkbox"/>	95	<input type="checkbox"/>	<input type="checkbox"/>
16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	36	<input type="checkbox"/>	<input type="checkbox"/>	56	<input type="checkbox"/>	<input type="checkbox"/>	76	<input type="checkbox"/>	<input type="checkbox"/>	96	<input type="checkbox"/>	<input type="checkbox"/>
17	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	37	<input type="checkbox"/>	<input type="checkbox"/>	57	<input type="checkbox"/>	<input type="checkbox"/>	77	<input type="checkbox"/>	<input type="checkbox"/>	97	<input type="checkbox"/>	<input type="checkbox"/>
18	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	38	<input type="checkbox"/>	<input type="checkbox"/>	58	<input type="checkbox"/>	<input type="checkbox"/>	78	<input type="checkbox"/>	<input type="checkbox"/>	98	<input type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	39	<input type="checkbox"/>	<input type="checkbox"/>	59	<input type="checkbox"/>	<input type="checkbox"/>	79	<input type="checkbox"/>	<input type="checkbox"/>	99	<input type="checkbox"/>	<input type="checkbox"/>
20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	40	<input type="checkbox"/>	<input type="checkbox"/>	60	<input type="checkbox"/>	<input type="checkbox"/>	80	<input type="checkbox"/>	<input type="checkbox"/>	100	<input type="checkbox"/>	<input type="checkbox"/>



0 143820 210515

Jednu správnou odpověď zaškrtněte takto:

Výsledkový formulář vyplňte přesně podle pokynů zkoušejícího!



# Pedagogická fakulta MU v Brně

Přijímací písemná zkouška na obor technická výchova

prezenční studium

červen 2002

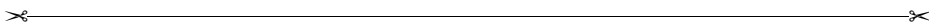


Údaje v části nad čarou vyplíte čitelně hůlkovým písmem, oddělte a uštrížek na pokyn odevzdejte.

Příjmení: ..... jméno: .....

rodné číslo:         /

číslo přihlášky:         **1301**



A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D				
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	21	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	22	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	23	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	33	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	34	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	35	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	36	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	17	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	27	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	37	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	18	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	28	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	38	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	29	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	39	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Právě jednu správnou odpověď zaškrtněte takto: ✕  
Výsledkový formulář vyplíte přesně podle pokynů zkoušejícího!





## 7. Závěr

### 7.1. Probíhající vývoj

Vývoj systému SCAT se nezastavil. Dále probíhá řešení rozpoznávání ručně psaných čísel. Testují se nové algoritmy, zkoušejí se nové přístupy, aby se eliminovala chybovost při analýze čísel. Jedna z možností je kontrolovat ihned při skenování (resp. zpracování) relace typu *rodné číslo – číslo přihlášky* a připravit uživatelské rozhraní pro odpovídající zásahy obsluhy.

Jinou oblastí vývoje, ve které se také pokračuje, je hledání nových skenovacích zařízení. Nové skenery umožňují zajímavé zrychlení. Skener HP 7450 C skenuje cca 10 stran za minutu, skener Brother MFC 9180 cca 8 stran za minutu. Vývoj možná povede i do oblasti digitální fotografie – tedy formulář by se již neskenoval po řádcích, ale fotografoval jako jeden celek. To by mohlo přinést opět výrazné urychlení celkového zpracování. Dále chceme systém rozšířit o možnost rozpoznávání čísel [5] a optimalizovat nastavení prahovacích konstant pomocí algoritmů z [8].

### 7.2. Shrnutí výsledků

Naším cílem bylo prokázat, že lze s využitím současných, běžně dostupných zařízení automatizovat vytváření, zpracování a vyhodnocení písemných testů. Díky automatickému způsobu převodu souřadnic přímo z elektronického obrazu formuláře, zaměření a srovnání naskenovaného obrazu jsme dosáhli požadovanou přesnost vyhodnocování. Vypracovali jsme způsob rozpoznání digitálních čísel. Navrhli jsme a vytvořili prostředí a logistiku vyhodnocování tak, že vznikl robustní a flexibilní systém. Systém byl úspěšně použit na zpracování téměř sta tisíce formulářů.

## Reference

- [1] Adriana Benadiková, Štefan Mada, a Stanislav Weinlich. *Čárové kódy, automatická identifikace*. Grada Publishing, 1994.
- [2] Miroslav Hrad. *Automatizace vytváření, zpracování a vyhodnocení písemných testů*. Diplomová práce, Masarykova univerzita v Brně, Fakulta informatiky, duben 2001.
- [3] Donald Erwin Knuth. *The T<sub>E</sub>Xbook*. Addison-Wesley, Reading, Massachusetts, 1984.
- [4] Miroslav Mařák. *Automatizované rozpoznávání testů*. Bakalářská práce, Masarykova univerzita v Brně, Fakulta informatiky, 1999.
- [5] Roman Mikolaj. *Gravitačný algoritmus rozpoznávania znakov*. Diplomová práce, MFF UK, Bratislava, katedra informatiky, březen 1998.

- [6] Petr Olšák. The EAN barcodes by T<sub>E</sub>X. *TUGboat*, 15(4):459–464, 1994.
- [7] Jordi Saludes. Fast and secure multiple-option tests. *TUGboat*, 17(3):310–319, 1996.
- [8] Michail I. Schlesinger a Václav Hlaváč. *Deset přednášek z teorie statistického a strukturního rozpoznávání*. Vydavatelství ČVUT, Praha, 1999.
- [9] Petr Sojka. An Experience from a Digitization Project. *Cahiers GUTenberg*, (28–29):276–281, březen 1998.
- [10] Petr Sojka. Publishing Encyclopaedia with Acrobat using T<sub>E</sub>X. V *Towards the Information-Rich Society. Proceedings of the ICCC/IFIP conference Electronic publishing '98*, strany 217–222, Budapest, Hungary, duben 1998. ICC Press.
- [11] Petr Sojka, Rudolf Červenka, a Martin Svoboda. T<sub>E</sub>X for database publishing. V Jiří Zlatuška, editor, *Proceedings of the 7th European T<sub>E</sub>X Conference, Prague, 1992*, strany 53–58, Brno, září 1992. Masarykova univerzita v Brně.

## Summary: Batch Typesetting and Scanning of Forms

The paper describes SCAT (SCAnning Test) technology for batch typesetting and scanning of forms on a desktop PC. SCAT has been used for typesetting and scanning of almost 100,000 pieces of forms at the Masaryk University and other institutions in the last three years. The system is based on T<sub>E</sub>X and other Open Source tools. The design of the system that aims at minimal error rate (using bar code technology) is presented. Speed, security and maximal automatization aspects of SCAT are discussed.

*Miroslav Hrad, Petr Sojka*  
*Masarykova univerzita v Brně*  
*Fakulta informatiky, Botanická 68a, 602 00 Brno*  
*Email: xhrad@fi.muni.cz sojka@fi.muni.cz*  
*a*  
*AMD, v. o. s., Návřší Svobody 26, 623 00 Brno*  
*Email: psojka@mistral.cz*