



Tvorba uživatelských rozhraní

Výtah z knihy Human-Computer Interface

I. Úvod

Naše společnost pokročila do věku, kdy je lidská práce nahrazována prací strojů. K tomu, aby byla účinná, je nezbytné, aby člověk mohl se strojem komunikovat na takové úrovni, jaká postačuje ke sdělení všech požadavků, které má stroj vykonat.

Mnohdy však se stroji nepracují jen profesionálně, a tak je nutné, aby tato komunikace probíhala pokud možno co nejintuitivnějším způsobem. Intuitivní je pro člověka to, s čím se setkává běžně v průběhu celého života. Ačkoliv se během vývoje mnohdy uživatelská rozhraní orientovala spíše na stroj než na člověka a pro efektivní práci bylo od uživatele vyžadováno značné přizpůsobení se stroji, dnes již technika pokročila natolik, že trend směřuje k co nejpřívětivějšímu ovládání z hlediska uživatele. To znamená takové ovládání, které napodobuje úkony běžného života nebo simuluje komunikaci člověka s člověkem.

Tato práce by měla sloužit jako stručný průvodce tvorbou takových uživatelských rozhraní která odpovídají moderním trendům. Jejím cílem není přesně postihnout celou problematiku, jako spíše poskytnout přehled pravidel a nápadů, které se v průběhu vývoje uživatelských rozhraní objevily.

II. Komunikační cyklus

Komunikační procesy, se kterými se setkáváme v běžném životě jsou ve svém principu podobné, jako ty, o kterých bude pojednáno v této práci. Základním schématem pro ně může být například Normanův komunikační cyklus. Komunikaci člověka se strojem je podle Normana možné rozdělit do několika bodů, které postihnou celý proces. Jsou to:

- Určení záměru
- Formulace záměru
- Specifikace posloupnosti akcí
- Provedení akcí
- Odbdržení reakce
- Interpretace reakce
- Vyhodnocení reakce vzhledem ke stanoveným cílům

Abychom lépe porozuměli jednotlivým bodům uvedeme si zde obecný příklad:

Příklad: Rožnutí světla

Člověk sedí na židli v místnosti, v níž chce rožnout světlo. **Záměrem** je tedy získat větší množství světla. Světla docílíme díky lampě. **Formulací** záměru je tedy rožnutí lampy. **Posloupnost** akcí k dosažení cíle je tedy přiblížení se k lampě a **zmáčknutí** tlačítka. **Reakcí** systému po provedení je nárůst světla v místnosti, který člověk **zaznamená** a **vyhodnotí**. Pokud je světla **dostatek** je proces ukončen, pokud ne, začíná od začátku například s hlavním světlem v místnosti.

Normanův systém se využívá k demonstraci, jaké problémy způsobuje návrh uživatelského rozhraní (user interface/UI). Ne všechny body jsou pro člověka snadno vyhodnotitelné a zároveň ne všechny body umí počítač vždy přeložit a vykonat. Čím více úsilí musí uživatel vykonat, aby dosáhl cíle, tím méně efektivní je komunikace.

Pro efektivní komunikaci je tedy třeba vyvinout efektivní uživatelské rozhraní. UI leží mezi uživatelem a systémem. Způsob jakým člověk komunikuje se systémem se nazývá vstup. Úkol, jenž zadává člověk musí být proto artikulován v nějakém jazyce, kterému systém rozumí. Vstupní jazyk je přeložen do jazyka systému. Systém se transformuje podle daných operací zadaných uživatelem. Po dokončení se dostává do dalšího stavu, v němž musí uživateli zpřístupnit plody své práce. Ty uživatel pozoruje na výstupu, analyzuje je a vrací se do stavu formulace a zadávání dalších úkolů na vstup. Tím se celý proces opakuje.

Jazyková vazba na vzájemnou interakci je časým zdrojem chyb v komunikaci mezi uživatelem a strojem, které způsobují nižší efektivitu práce a mohou dojít až k frustraci uživatele. Chyby, jichž se dopouští uživatel lze rozdělit do dvou kategorií:

- Přehmaty
 - Vyplývají z nepřesností – jedná se např. o kliknutí na špatné tlačítko
 - Řešení – vzdálit tlačítka více od sebe, zvětšit je, zvýraznit
- Neporozumění
 - Vyplývají ze špatného pochopení – ikona s lupou nemusí v nějakém konkrétním případě znamenat hledání, ale zvětšování
 - Řešení – musí být v lepší specifikaci rozhraní – logičtější ikony

Pro předcházení chybám a pro zefektivnění práce se systémem lze ovládací prvky a výstupy organizovat podle tří kritérií, a to funkcionálně, sekvenčně a frekvenčně.

Funkcionální organizace je taková, kde spolu funkčně podobné prvky sousedí a jsou uspořádány do logických celků. Sekvenční řadí prvky tak, aby byly pospolu a seřazeny podle pořadí, v němž se používají. Frekvenční organizace umísťuje ovládací prvky tak, aby ty nejpoužívanější byly na nejvýraznějším a nejlépe dostupném místě.

Dalším způsobem jak předcházet vzniku uživatelských chyb je správná volba barev.

II. 1 Použití barev

Barvy se neužívají pouze pro okrasu, ale jsou užitečným pomocníkem pro zvýrazňování důležitých prvků. Jak již bylo řečeno, v současnosti je vyvíjena snaha o tvorbu co nejintuitivnějších uživatelských rozhraní. Proto je dobré volit barvy tak, aby měly pro uživatele svůj zažitý význam a reprezentovaly adekvátní informace. Například modrá není příliš výrazná, proto se nehodí pro zvýraznění kritických informací. Naproti tomu červená, oranžová a zelená jsou barvy zažité všem řidičům, a proto se hodí k relevantním účelům. Červená jako výrazné upozornění atd.

Při návrhu je vždy třeba si uvědomovat, že zejména u velkých mezinárodních projektů mohou uživatelé z různých částí světa vnímat různé stimuly různě. V západní společnosti znamená červená výstrahu, naproti tomu v Číně je symbolem radosti. Toto upozornění se netýká pouze barev ale má široké pole zastoupení.

II. 2 Druhy interakčních rozhraní

V průběhu vývoje se vyskytlo mnoho druhů uživatelských rozhraní. Od některých je v současnosti již upouštěno, jiná zažívají vzestup. Celkem lze jmenovat tyto:

- Příkazová řádka – hodí se spíše pro zkušené uživatele. Pro nováčky jsou příkazy matoucí
- Menu – zobrazuje možnosti, po jejichž zvolení se uživatel dostává k výsledkům, kterých chce se systémem dosáhnout
- Přirozený jazyk – prozatím náročný
- Dialogová rozhraní – člověk vybírá z několika možností podle toho, jaké „otázky“ mu systém klade. Problém nastává v případech, kdy chce zvolit více možností
- Vyplňovací formuláře – nejčastěji pro vstup. Člověk navolí co potřebuje, systém to zpracuje a na výstup dá odpověď
- WIMP – nejčastěji používané, viz dále
- Klikací rozhraní – souvisí s WIMP. Využívá se zejména myši
- 3D rozhraní – uživatel se pohybuje nikoli ve 2D, ale v 3D. Poskytuje více prostoru, ale s 2D zařízeními jako je myš nepříliš efektivní. Možné využití ve virtuální realitě

II. 3 WIMP

Obsahuje mnoho elementů, které se souhrně nazývají Widgets a slouží k ovládání systému. Některé jsou klíčové, jiné volitelné. WIMP je v současnosti nejklassičtějším řešením UI.

Widgets:

- Okna – základní ovládací prvek, každé je nezávislý terminál
- Ikony – odkazy na funkční prvky
- Kursory – slouží k orientaci, ovládání, vyjadřují probíhající procesy
- Menu – sdružují důležité funkce systému. Problém je s množstvím a rozmístěním prvků. Příliš mnoho je matoucí, některé jdou sdružovat, ale mohou být sdruženy do více skupin a je třeba rozhodnout podle jakého klíče je zařadit. Menu může být “schovávací”. Takové, které se objeví až po přjetí k okraji obrazovky. Některé prvky menu je dobré dávat dál od sebe, aby nedošlo k chybě uživatele – místo save zmáčkne delete atd.
- Tlačítka – ovládací prvky, lepší jsou 3D – lákají ke kliknutí
- Toolbary – rozhraní s ikonami, uživatelé mívají problém se naučit význam ikon, hodí se k položkám menu přidat stejné ikony jako jsou v toolbaru, aby se uživatel snáz naučil jejich význam
- Palety – rozhraní, které podle zvoleného modu práce zobrazují jen k němu určené nástroje
- Dialogové boxy – slouží k upoutání pozornosti. Oznamují důležité zprávy a upozornění, které předchází možným chybám

III. Kritéria pro návrh uživatelských rozhraní

Pro vytvoření kvalitního uživatelského rozhraní, které bude jak vyhovovat uživateli, tak i umožňovat rychlou a účinnou práci je třeba dodržovat jistá kritéria.

Uživatelské rozhraní, které má být úspěšné musí být **ergonomické**. To znamená, že velikost tlačítek pro ovládání musí být taková, aby nedocházelo k chybám zmíněným výše. Dále musí být skutečně **vhodné** pro ten typ práce, k jakému je určeno. Jiné ovládání má například textový editor a jiné grafický editor.

Pokud vytváříme uživatelské rozhraní pro nějaký stroj, musíme brát v potaz i **bezpečnostní hledisko**, aby například uživatel nepřišel do styku s elektrickým proudem a podobně. Stejně tak musí být ovládací prvky na fyzickém zařízení takové, aby umožňovaly například **snadnou údržbu**.

Rozhraní musí splňovat jistá **ekonomická omezení**. To znamená, že jeho vývoj a výroba nesmí přesáhnout takovou hodnotu, kdy už by výsledná cena byla pro zákazníka příliš vysoká. Je samozřejmé, že každý zákazník, zejména jedná-li se o neprofesionála vybírá hlavně podle prvního dojmu, který do značné míry utváří **estetický dojem** z rozhraní. Proto je nutné, aby náš produkt byl nejen kvalitní uvnitř, ale také aby se dobře prezentoval navenek.

K těmto základním kritériím lze vymyslet ještě mnoho dalších. Pro naši potřebu lze uvést například Schneidermanova kritéria pro UI:

- Viditelnost klíčových prvků
- Zřetězení všech akcí a adekvátní odpovědi od systému
- Vratnost všech uživatelských akcí tak, aby systém nebránil v experimentování
- Syntaktická korektnost taková, aby každá uživatelská akce byla legální operací
- Nahrazení komplexních příkazů jednoduchými manipulacemi s viditelnými objekty

III. 1 Proces navrhování

III. 1. 1 Cíle a omezení návrhu

Při navrhování vhodného uživatelského rozhraní je vždy nutno zvažovat několik aspektů. V první řadě jsou to cíle projektu. To znamená, jaký je účel designu který navrhujeme? Pro koho je určen?

Je zřejmé, že jiné požadavky mají například důchodci a jiné teenageři. Jakmile si ujasníme cíle, je třeba vzít v úvahu i omezení. Například jaké použít materiály? Jaké standardy využít? Kolik to bude stát? Kolik času je na vývoj? Má návrh nějaká bezpečnostní rizika? Jakmile si položíme tyto a další jim podobné otázky, je důležité zvolit ten správný kompromis. Účelné je vybrat si nejdůležitější cíl, podle něj se zařadit a následně vše přizpůsobit omezením a minoritním cílům.

Pro úspěch naší práce je nutné dobré plánování. Máloměrý projekt má šanci být úspěšný, pokud před jeho realizací nedojde ke zvážení předchozích bodů.

Žádné rozhraní není perfektní. Snaha o perfekcionismus vede vždy k obrovským časovým a finančním nárokům. Proto není tak důležité vyřešit všechny problémy, ale vybrat ty, které stojí za vyřešení. Produkt, který sice má dokonalé ovládání, ale přišel na trh v době, kdy už není použitelný, nebo produkt který je výborně odladěný pro všechny druhy práce, ale kvůli složité implementaci stál mnoho finančních zdrojů a prodává se za vysokou cenu, kterou není nikdo

ochoten platit, je horší, než ten ne zcela dokonalý, ale zato přiměřeně funkce i cenově vyvážený.

Při návrhu je vždy třeba mít dvě základní znalosti:

- Znalost počítačů – omezení, kapacity, nástroje, platformy
- Znalost člověka – psychologické a sociální aspekty, lidské nedostatky

Dobrý návrh uživatelského rozhraní musí být uvážen před započítáním projektu. Mnohdy je uživatel brán v potaz až při konečných úpravách, což vyúsťuje v přílišnou orientaci na systém a následné špatné přizpůsobování se člověku.

Designový proces se skládá z následujících kroků:

- Co je potřeba – prvním krokem je zjištění, co je nezbytně nutné. Zde je důležité znát potřeby uživatelů, jaké jsou v současnosti využívané postupy. Toho se dosahuje rozhovory s lidmi, sledováním jejich práce, analýzami objektů a dokumentů s nimiž pracují.
- Analýza – po sesbírání dostatku informací je nutno provést jejich analýzu
- Design – na základě analyzovaných materiálů je proveden návrh. Od procesu zjišťování co udělat, přichází posun, jak to udělat.
- Iterace a prototypy – v průběhu návrhu a vývoje vznikají prototypy, které se vrací zpět do stádia analýzy a jsou vyhodnocovány. V této fázi se produkty poprvé dostávají k uživatelům, jejichž práce s nimi je posuzována.
- Implementace – jakmile jsme spokojeni s výsledkem, můžeme se pustit do programování, případně výroby hardwaru.

III. 1. 2 Zaměření na člověka

Při návrhu UI je nutné vždy myslet na koncového zákazníka. Není vždy možné se při realizaci spoléhat na to, že já bych to chápal takto a takto, a že proto to tak bude chápat každý. Proto je třeba zjistit, kdo budou oni budoucí uživatelé, jak budou staří, jaké zkušenosti budou mít s počítači atp. Je také užitečné uživatele pozorovat při práci. Sledovat, na co se soustředí, co naopak přehlíží, jakým způsobem pracuje. Dále je rozumné vést s nimi dialog. Proč dělají to co dělají, s čím mají problémy. Při vývoji UI je dobré v jistých okamžicích předávat uživatelům prototypy systému k odzkoušení. Je jednodušší měnit schéma rozpracovaného projektu, než toho téměř hotového.

Pro vedení podniku, pro který budeme vytvářet systém může být jeho skutečná funkcionalita skrytá. Proto je dobré nevyslechnout si pouze, jak modelované procesy uvnitř zákaznickovy firmy mají fungovat, ale i vyzpovídat zaměstnance a zjistit, jak tyto procesy skutečně fungují.

Při realizaci je účelné držet se těchto zásad:

- Komunikace s druhými – jiní vývojáři, klienti a uživatelé
- Vyhodnocování jiných modelů – způsob jakým je vyřešen systém podobný našemu

může sloužit jako pozitivní i jako negativní příklad

- Dynamické vyhodnocování – sada screenshotů může říci, jak systém vypadá, ale ne jak se chová, v jistých fázích vývoje by měl být k dispozici prototyp
- Lineární přístup – lidé lépe zvládají lineární sadu úkonů, to znamená, že každá věc se lépe dělá v krocích než najednou

Uživatel využívající našeho systému musí vždy vědět, kde se právě v systému nachází, co v každém stavu, v němž se nachází, může zrovna udělat, ví kam směřuje a co nastane při které akci a ví kde byl a co už udělal.

III. 1.3 Design obrazovky a layout

U návrhu si musíme položit tyto otázky: Co chce uživatel dělat? Jaký druh informací k tomu potřebuje? V jakém pořadí tyto informace bude potřebovat?

Po zodpovězení těchto otázek přichází návrh designu. V něm je důležité aby forma sledovala funkčnost. Neboli aby se vzhled řídil podle toho, jaké operace budou prováděny.

Seskupování

Jedním ze základních prvků návrhu je seskupování relevantních informací. Pokud chceme zobrazit informace například o vlastníkovi auta, musí být tyto umístěny logicky pospolu, aby bylo ihned patrné, k čemu patří.

Rozmístění

Dalším prvkem je samotné rozmístění prvků a seskupených informací. Správný návrh by se měl odvíjet od důležitosti zobrazovaných entit pro uživatele. V tomto kroku je však opět nutné brát v potaz, že ne všichni uživatelé mají stejné vnímání. To, kam se soustředí pohled při hledání klíčových informací, je odvislé od mateřského jazyka uživatele. Evropan a Američan, zvyklí číst odshora dolů zleva doprava, budou logicky nejžádanější prvky hledat právě v tomto pořadí. Naproti tomu například Arab zvyklý číst zprava doleva bude mít od rozhraní právě takové očekávání.

Dekorace

U dekorování je třeba mít na paměti, že méně je mnohdy více. Pokud uživatele zahltime spoustou barev a fontů, bude sice možná okouzlen, jeho práci to však neprospěje. Stejně tak příliš syté barvy pozadí nejsou ku prospěchu věci. Černý text na tmavě modrém pozadí je nečitelný. U barev také opět musíme myslet na to, že ne právě zanedbatelné množství lidí je nějakým způsobem barvoslepých a náš návrh nemusí vnímat stejně jako my. Pokud jde o 3D prvky je lépe se jim vyhnout, pokud nemají nějaké funkční opodstatnění. Text, který je pozorován z úhlu, se velmi špatně čte.

III. 2 Prototypování

Jak již bylo řečeno, vývoj uživatelského rozhraní by měl být iterativní. Po jisté době by měl vždy dospět k určitému prototypu, který je odzkoušen a následně se z něj vychází při dalším doladování. Sebelepší ladění je však k ničemu, pokud projekt vychází ze špatného konceptu. Proto je nejen nutné efektivně rozvíjet návrh, ale mít jistotu, že tento návrh má smysl a že je kvalitní.

Jedině, když vycházíme z dobrého, informovaného nápadu máme šanci vytvořit fungující produkt.

III. 2. 1 Validace a verifikace

V průběhu vývoje musíme mít představu o tom, jakým směrem se náš projekt vyvíjí. Tomuto procesu se říká validace a verifikace. Validace je přesvědčování se, že děláme správnou 'věc', verifikace je, že děláme 'věc' správně. Aby byl projekt úspěšný, musí být splněny obě.

Validace designu demonstruje, že návrh odpovídá požadavkům a představám zákazníka. Je subjektivnější a méně formální než verifikace. Vzniká tak fenomén zvaný formalizační propast, který označuje neformální přechod od zákazníkem formulovaných požadavků k formálnímu zápisu a následné realizaci těchto představ. Validace může být prováděna buď samotným designérem nebo ve spolupráci se zákazníkem.

Verifikaci je možno realizovat formálnější cestou. Díky tomu je možné objektivně rozhodnout, zda je projekt vyvíjen správným způsobem a zda neobsahuje formální chyby.

Pokud sledujeme uživatele, jak pracuje s naším systémem, můžeme z jeho akcí vyvodit důležité důsledky. Analytickou metodou lze odhalit např. tyto skutečnosti:

- Doba nutná k dokončení úkolu
- Kolik procent z úkolu lze dokončit za jednotku času
- Poměr úspěšných a neúspěšných pokusů
- Čas strávený řešením chyb
- Počet použitých příkazů
- Frekvence užívání nápovědy
- Kolikrát rozhraní zmátlo uživatele
- Kolik příkazů nebylo užito
- Počet uživatelů preferujících náš systém
- Kolikrát uživatel pocítil uspokojení nebo frustraci
- a mnoho dalších

Pomocí těchto zjištění lze poté odhalit, v jakých směrech je náš systém silný, či naopak slabý a naši pozornost při vývoji soustředit právě tam.

III. 2. 2 Iterativní design a prototypování

Při vývoji je užitečné vytvářet prototypy. Vzhledem k účelu je možné prototypy rozdělit na následující:

- „Throw away“ prototyp – prototyp je vytvořen a testován. Poznatky díky němu získané jsou využity k dalšímu vývoji, ale prototyp sám je zahozen.
- Inkrementální prototyp – konečný produkt je vytvářen jako oddělené komponenty. Pro finální systém sice existuje jednotný design, vývoj však probíhá po částech, které jsou nezávislé a jakmile jsou hotovy, jsou uvolněny na trh jako série produktů.
- Evoluční prototyp – prototyp, jakmile je zhotoven, není po použití zahozen, ale slouží jako základ pro další vývoj. Celý systém se tak vyvíjí od velmi omezené

prvotní verze až po konečný produkt.

Problémy s vytvářením prototypů jsou zejména ty, že každý zabere relativně hodně času, který by mohl být teoreticky věnován jiným úkolům. Dalším problémem bývá, že prototyp je ze své podstaty nekompletním dílem. Tato nekompletnost se mnohdy projeví absencí některých fundamentálních prvků, jako je spolehlivost nebo bezpečnost. Ty však mohou být klíčové k přijetí konečného produktu. Naproti tomu pro plánování a realizaci je nezbytné mít představu o směru vývoje produktu a o uživatelských reakcích na produkt. Proto je přinejmenším vhodné nějaké prototypy v průběhu vývoje vytvořit a testovat.

Prototypování je možné provádět různými způsoby jako jsou:

- Storyboardy – jako ve filmech, kde je scéna nejprve předkreslena do rámečků, aby vznikla představa o tom, jak bude záběr vypadat, lze i u vývoje UI začít prototypováním přes nákresy. Ty mohou přinést jasnou představu o tom, zda jde vývoj dobrým směrem. Jejich nedostatek tkví ve staticnosti, která se dá ale částečně odbourat animacemi.
- Simulace s omezenou funkcionalitou – když potřebujeme odhalit interaktivní aspekty systému, musíme přejít k tomuto prototypu. To znamená vytvořit grafické a textové rozhraní které bude napodobovat chování systému. Jakmile je zhotoveno, může být snadno dotvářeno a měněno k získání reakcí různých uživatelů.

IV. Principy, jak zajistit užitnost

- Pochopitelnost – míra úsilí potřebná k naučení se pracovat se systémem
- Flexibilita – zajištění více způsobů, jak si uživatel se systémem vyměňují informace
- Robustnost – kolik podpory systém dává uživateli aby dosáhl svých cílů

IV. 1. 1 Pochopitelnost

Pochopitelnost můžeme rozdělit do několika bodů:

- Předvídatelnost – podporuje uživatele, aby dokázal ze svých minulých zkušeností se systémem předvídat jeho budoucí chování. Za normálních okolností uživatel nemá rád překvapení. Proto je vhodné, když se systém chová předvídatelným způsobem, tj. podle mentálního modelu, který si vytvořil na základě předchozích zkušeností.
- Soudržnost – systém se chová tak, že v jeho odpovědích je možné nalézt jistou integritu, nad níž lze vystavět myšlenkový model chování systému.
- Znalost z dřívějšíka – i zcela nový uživatel systému si s sebou bere z dřívějšíka nějaké zkušenosti s ovládáním podobných nástrojů. Proto je dobré, aby pro něj nebyla první zkušenost příliš velký šok, a aby byl schopen se zorientovat.
- Zobecnitelnost – podporuje uživatele, aby mohl rozšiřovat své znalosti o specifických druzích interakce z těch, které již nabyl, na ty jim podobné.

IV. 1. 2 Flexibilita

Flexibilita může být rozdělena do následujících bodů:

- Omezení vstupu a výstupu – občas je třeba zabránit uživateli v měnění nějakého obsahu, např. souboru který je otevřen někým jiným apod.
- Předávání úloh – v některých situacích uživatel potřebuje pomoci, aby se dostal dál. V takovém případě za něj může systém provést požadované kroky. Naopak někdy chce uživatel vstoupit do automatizovaného procesu a provést jej ručně.
- Zaměnitelnost – užívání různých jednotek pro stejný vstup. Např. někdo zadává délku v metrech, někdo ve stopách a automatický převod těchto jednotek uživateli usnadňuje práci.
- Přizpůsobitelnost – uživatel si mnohdy rád přizpůsobí systém tak, aby vyhovoval právě jemu, tj. přestaví ikony, otevře, či zavře panely atd. Naopak systém se může přizpůsobit uživateli tak, že rozezná úroveň jeho práce buď jako profesionální, nebo amatérskou, a tak k němu bude dále přistupovat.
- Multimodální design - Multimodální design je design využívající co nejširší palety komunikačních prostředků. Většina interaktivních počítačových systémů využívá zejména grafické prostředky pro komunikaci s uživatelem. Čím komplexnější tyto

systémy jsou, tím přehlcenější se stává grafické uživatelské rozhraní. To může vést k frustraci a chybám uživatele.

Využíváním i jiných smyslových vjemů může být ulehčeno zraku při zachování informační hodnoty komunikace. Užíváním více smyslů se zvětšuje šířka interakce a komunikace se více přibližuje přirozené komunikaci mezi lidmi. Nejde jen o přidávání nových, ale i o přidávání redundantních informací. Tím je myšleno, že informace prezentovaná graficky je též poskytnuta ve zvukové formě, což vylepšuje uživatelskou interakci se systémem.

IV. 1.3 Robusnost

Robusnost může být rozdělena do následujících bodů:

- Průhlednost – umožňuje uživateli rozpoznat, co se děje uvnitř systému podle toho, co vidí a slyší.
- Zotavitelnost – dovoluje uživateli řešit situace, když se objeví chyba.
- Rychlost odpovědi – systém by měl být co nejrychlejší. Nejlépe takový, aby jeho odpovědi vypadaly pro uživatele jako okamžité. Pokud není možné zajistit rychlou odpověď, je nutné, aby uživatel věděl, že jeho požadavek byl přijat, a že systém pracuje na jeho provedení.
- Přizpůsobivost – do jaké míry systém umožňuje uživateli vykonat požadované operace a jak jim uživatel je schopen porozumět.

IV. 2 Zlatá pravidla

Toto byla různá pravidla a pomůcky pro tvorbu uživatelských rozhraní zaměřených na uživatele. Je důležité si uvědomovat, že tato pravidla poskytují pouze rámcovou představu o návrhu a pro každou situaci je třeba je interpretovat individuálně. Na závěr ještě 8 zlatých pravidel a několik heuristik, které v sobě elegantně koncentrují mnoho z toho co bylo již výše řečeno .

8 Schneidermanových zlatých pravidel pro design prostředí

1. Usilování o konzistenci v sekvencích akcí, rozvržení obrazovky, terminologii, užívání příkazů apod.
2. Umožnění častým uživatelům používat zkratky, aby mohli klasické akce vykonávat rychleji.
3. Nabídnutí informativních zpětných vazeb pro každou akci uživatele uspořádané podle důležitosti akce, o níž informuje.
4. Zajištění. aby uživatel věděl, že systém vykonal příkaz.
5. Předcházení chyb nebo jejich snadné řešení, ideálně aby uživatel neměl možnost udělat chybu a kdyby ji udělal, aby systém nabídl jednoduché řešení.
6. Zajištění jednoduchého vracení akcí pro umožnění provádění experimentů se systémem.
7. Vytvoření ovládacích prvků, umožňujících uživateli manipulaci se systémem.
8. Šetření krátkodobé paměti uživatele vytvářením designově jednoduchých obrazovek.

Nielsenovy heuristiky:

- Viditelnost - uživatelé musí vidět, co se děje, a to vhodným způsobem a v přijatelném čase. Když bude nějaká událost trvat dlouho, je dobré na to uživatele upozornit a nabídnout mu např. stavovou řádku s ukazatelem kolik zbývá do konce. Uživatel by neměl být nucen si pamatovat mnoho údajů z předchozí obrazovky. Pokud je něco potřeba, mělo by to být na očích.
- Souhlas mezi systémem a skutečným světem - systém by měl mluvit jazykem uživatele. To znamená, že by místo systémově orientované komunikace měl spíše vycházet z komunikace mezi lidmi, dávat logické odpovědi v logickém pořadí apod.
- Uživatelská svoboda - uživatelé často dělají chyby a musí být snadné se z nich vrátit - tlačítko undo.
- Předcházení chyb - lépe než mít dokonalé oznamování a řešení chyb, je dobré nedovolit, aby tyto chyby mohly nastat.
- Flexibilita a efektivnost - pro zkušené uživatele by měly být k dispozici nástroje pro urychlení práce, o nichž nezkušený uživatel nemusí vůbec vědět, dokud systém lépe neovládne.
- Estetický a minimalistický design - vždy by měly být k dispozici jen relevantní informace. Každá nepodstatná věc zaplňuje obrazovku a činí ji méně přehlednou.
- Návod a dokumentace - jen málo systémů nepotřebuje dokumentaci, proto je nutné, aby měl uživatel k dispozici návody, jak řešit různé problémy. V těchto návodech by mělo být co nejsnadnější najít požadované řešení a měly by být koncipovány krok po kroku.

V. Závěr

Existuje dobrý důvod se domnívat, že vývoj výpočetní techniky bude v budoucnu pokračovat. Stále je mnoho možností kudy by se mohl ubírat a jednou z těch pravděpodobnějších je, že cestou co nejlepší a nejjednodušší komunikace s člověkem. Proto je dobré zůstat v tomto smyslu v obraze a držet krok s dobou.

Práce, kterou jste měli možnost si přečíst, má sloužit jako jistá osnova pro tvorbu kvalitního uživatelského rozhraní. Její obsah čerpá z knihy Human-Computer Interaction od kolektivu autorů Dix, Finlay, Abowd, Beale, která je mezi referenční literaturou pro předmět ITU. Celá kniha má přes osm set stran a je celá v angličtině, takže předchozí text berte jako určitý výťah, který Vám může umožnit informovanější pohled na tvorbu uživatelských rozhraní, ať už pro účely projektu do předmětu ITU, nebo do budoucího života.

Poznámka: Aktuální verze dokumentu není finální. Ta by měla obsahovat nějaké obrázky a formální náležitosti k podobným dokumentům patřící. Za chyby v textu se omlouvám a doufám, že Vám tato práce poskytne alespoň nějakou užitečnou informaci.

VI. Zdroje

- [1] Preece J.: Human-Computer Interaction,
Addison-Wesley, Wokingham, UK, 1995,
ISBN 0-201-62769-8