

UNIVERZITA KARLOVA V PRAZE

FAKULTA HUMANITNÍCH STUDIÍ

Katedra ELEKTRONICKÉ KULTURY A SÉMIOTIKY

Bc. Pavel Prokop

**Sémiotické inženýrství
a pravidla pro interakční design**

Diplomová práce

Vedoucí práce: Mgr. Jan Brejcha

Praha 2010

Prohlášení

Prohlašuji, že jsem předkládanou práci zpracoval/a samostatně a použil/a jen uvedené prameny a literaturu. Současně dávám svolení k tomu, aby tato práce byla zpřístupněna v příslušné knihovně UK a prostřednictvím elektronické databáze vysokoškolských kvalifikačních prací v repozitáři Univerzity Karlovy a používána ke studijním účelům v souladu s autorským právem.

V Praze dne 15.září 2010

Pavel Prokop

Obsah

1 Úvod.....	5
1.1 Počátky HCI.....	5
1.2 Hypotéza a cíl práce.....	7
2 Teoretická část.....	9
2.1 HCI.....	9
2.1.1 Obory HCI.....	9
2.1.2 Cesta k rozhraní UCD.....	10
2.1.3 UxD.....	11
2.1.4 komunikace.....	12
2.2 Sémiotika.....	13
2.2.1 Proč sémiotika?.....	13
2.2.2 Krátké srovnání Saussura a Peirce.....	14
2.2.2.1 Saussure.....	14
2.2.2.2 Peirce.....	15
2.2.2.3 Interpretans.....	18
2.2.2.4 sémióza.....	18
2.2.2.5 Abdukce.....	19
2.2.2.6 Umberto Eco – Teorie znakové produkce.....	20
2.3 Sémiotické přístupy k HCI.....	22
2.3.1 Andersen.....	22
2.3.1.1 Počítačová sémiotika.....	25
2.3.1.2 Počítačový znak.....	26
2.3.2 Další sémiotické přístupy.....	27
2.3.2.1 Nake a Grabowski.....	27
2.3.2.2 Mihai Nadin – Sémiotické paradigma.....	29
2.4 Sémiotické inženýrství v HCI.....	32
2.4.1 Historie.....	32
2.4.2 Poslední exkurz.....	32
2.4.2.1 Řečové akty.....	33
2.4.2.2 Komunikační maximy.....	33
2.4.3 Vymezení Sémiotického inženýrství.....	34
2.4.4 Základní pojmy SI.....	38
2.4.4.1 Metakomunikace.....	39
2.4.4.2 Designerův zástupce.....	40
2.4.5 Ontologie sémiotického inženýrství.....	41
2.4.5.1 Procesy signifikace.....	42
2.4.5.2 Procesy komunikace a jednotliví mluvčí.....	43
2.4.5.3 Prostor SI.....	44
2.4.6 Epistemologické následky Sémiotického inženýrství.....	47
2.4.7 Metody sémiotického inženýrství.....	48
2.4.7.1 Metoda sémiotické inspekce (SIM).....	49
2.4.7.2 Metoda hodnocení komunikativnosti (CEM).....	52
3 Pravidla sémiotického inženýrství.....	61
3.1 Postup tvorby.....	61
3.2 Pravidla.....	62
3.2.1 Pravidla odvozená z teoretické části.....	62
3.2.2 Pravidla odvozená z metody.....	65
3.2.3 Pravidla odvozená z praxe.....	68
3.3 Povaha pravidel.....	71
4 Závěr.....	72

Abstrakt

Sémiotické inženýrství je jedním ze sémiotických přístupů k problematice interakce mezi člověkem a počítačem. Jeho teorie je postavena na jednoduché myšlence rozhraní jako zástupce designera, skrze kterého designer promlouvá k uživateli o tom, jak by měl zacházet s programem, který pro něj připravil. Kromě teoretického základu disponuje také metodami pro zhodnocení způsobu komunikace testovaného rozhraní s uživatelem. Je tak schopno se vyrovnat ostatním teoriím v oblasti a díky svému zaměření na komunikaci poskytuje navíc nový podstatný úhel pohledu. Jeho teorie a metody jsou zde použity jako základ pro vytvoření sady pravidel pro designery, která by jim měla sloužit jako vodítka k zamyšlení. Ukazuje se tak, že sémiotická teorie je schopna v oblasti interakčního designu přinášet konkrétní výsledky pro běžnou praxi a není vázána pouze na akademické prostředí.

Semiotic Engineering is one of semiotic approaches to human-computer interaction problematics. Its theory is based on a simple notion of interface being a designers deputy through which the designer communicates with users of this interface. Telling them, how to work with a program, he or she designed for them. Apart from theoretical foundation, semiotic engineering dispose also with research methods that can be used to evaluate a way an interface communicates with its users. This way it can compare with other non-semiotic theories in this branch of research and thanks to its orientation to communication, it is even able to bring new points of view. Its theory and methods are used in this thesis as a basis for creation of a set of rules for user interface designers, that should serve them as guidelines for their thinking about interface. It turns out that semiotic theory is capable of bringing concrete results for praxis in the area of interaction design and it is not bound only to the academic environment.

Klíčová slova

Human Computer Interaction, Sémiotické inženýrství, Interakční design, Sémiotika, Designerův zástupce, Metakomunikace, sémióza

1 Úvod

1.1 Počátky HCI¹

Obor HCI se v zásadě vynořuje v závěsu za prvními počítači, jelikož již od počátků existuje potřeba nějak komunikovat s uživatelem, přijímat od něj vstupy a zpracovávat je. Práci se zařízením už najednou není možné provádět na základě přímé manipulace tak, jak tomu vždy bylo u klasických strojů, nebo nástrojů industriální éry. Neexistují přímé převodní mechanismy mezi naším vstupem a výstupem přístroje. Dříve sice také existovali komplikované přístroje s množstvím ovládacích prvků, ale jejich funkce byla vždy stejná a předvídatelná. Nebyla potřeba ji nijak neustále komunikovat. Brzda na vozíku s uhlím je vždy ta samá brzda, její účel je stejný za jakékoli situace a práce s ní také. Tlačítko „OK“ v prostředí MS Windows však nabývá množství různých funkcí v závislosti na kontextu a momentálním stavu programu. V případě výpočetní techniky se tedy funkce mění v závislosti na kontextu a tuto změnu je třeba uživateli dávat neustále najevo.

Principy řídicí procesy odehrávající se v počítači (případně jakémkoli jiném digitálním zařízení) jsou natolik abstraktní, že člověk není schopen přímo s těmito principy manipulovat a potřebuje nějakého prostředníka (rozhraní). I samotný programovací jazyk, kterým se programy vytvářejí je abstrakcí a snahou převést strojový kód, fungující pouze na bázi vypnuto/zapnuto do člověku srozumitelné podoby logických operací, cyklů a manipulací s proměnnými. Dalo by se říci, že už sám programovací jazyk je nějakým primitivním rozhraním (Jak uvidíme níže termín rozhraní se nemusí nutně omezovat pouze na část programu viditelnou na obrazovce). Nelze ovšem očekávat, že se každý člověk naučí novému jazyku jenom proto, aby mohl pracovat s počítačem. Je třeba převádět sdělení stroje do nějakých srozumitelných forem. To znamená do forem odpovídajících obecně našim kulturním zvyklostem a konvencím prostředí a kontextu, ve kterém je program využíván. Rozhraní v úzkém slova smyslu² je jakýsi filtr, ležící na hranici mezi námi a strojem, převádějící jeho sdělení do námi srozumitelných útvarů a naopak naše vstupy zpět do strojem čitelného kódu. Tento význam ale není konečný a zejména sémiotická analýza ukáže, že úkolem rozhraní v širším smyslu je zejména zprostředkovávat styk mezi lidmi.

Jak už bylo řečeno, důvodem nezbytnosti rozhraní je příliš velká komplexita

1 Human Compute Interaction (interakce člověka s počítačem)

2 Jak se ukáže později je toto poněkud zavádějící definice, ale pro tento okamžik postačí.

procesů odehrávajících se v počítači. Člověk není schopen tuto komplexitu celou pojmut a pracovat s ní, je proto třeba vytvořit nějakého prostředníka, který ji pro něj zredukuje³. Uživateli (nejen) počítačového programu je pak nabídnuta pouze omezená sada možností v závislosti na situaci, ve které je program využíván. Spisovateli se nabízí značně redukováná schopnost počítače ukládat a znovu obnovovat data. Matematikovi zase spíše schopnost provádět logické operace. Oba tyto procesy jsou samozřejmě přítomné ve všech způsobech využití počítače, ale na vstupu a výstupu lidského rozhraní se projevují jenom některé.

Zde vstupuje do hry obor HCI, jehož úkolem je tvořit taková rozhraní, který by prováděla redukcí počítačové komplexity tím nejlepším způsobem v závislosti na účelu, za kterým člověk daný program užívá a na kontextu, ve kterém ho užívá. Jeden ze zakladatelů sémitického přístupu k HCI Peter Andersen říká, že každé rozhraní vybírá z kontinua neomezených možností⁴ a je tedy spolu s Nadinem možné říci, že vytváří konkrétní z abstraktního⁵.

První počítačová rozhraní samozřejmě již od začátku splňovala základní požadavek. Tj., umožňovala člověku využít nezměrný potenciál počítače k jeho konkrétním potřebám. Jak se ale postupem času aplikace přesouvali z výhradně akademického prostředí do prostředí kanceláří firem, přišel i požadavek větší srozumitelnosti rozhraní pro normálního uživatele. Počítačový program by neměl být pro uživatele překážkou, měl by být pouze co nejtransparentnějším prostředníkem k vykonání jeho práce. Úředník v bance se nechce zabývat memorováním povelů příkazového řádku. Jde mu o to, odvádět jeho práci a výpočetní systém by pro něj měl být oporou.

Postupně tedy v osmdesátých letech přestává jít pouze o otázku, jak převést řeč stroje na řeč člověka a začíná jít i o to, jak člověku práci s počítačem co nejvíce usnadnit. Zde právě leží výzva pro obor HCI.

Samotný obor je roztržštěn do mnoha přístupů a stále se hledá a snaží vymezit. Jednotlivé přístupy postupují různým způsobem, ovšem cíl zůstává společný: Tvořit srozumitelná uživatelská rozhraní tak, aby jejich užívání nebylo pro člověka prací, ale prostředkem k práci a samozřejmě i zábavě. HCI jako svébytná část procesu tvorby aplikací si teprve razí cestu a je pravda, že není zatím mnoho společností, které by si mohli dovolit provádět dlouhé a nákladné výzkumy ještě předtím, než nastoupí samotná fáze tvorby designu. Z větších společností, které mají vlastní týmy HCI designérů je možné

3 Nabízí se zde paralela s luhmanovým termínem "redukce komplexity".

4 Andersen 1991 s.13

5 Nadin 1998 s.289

jmenovat například Microsoft, Apple nebo Nokia. Je pravděpodobné, že v budoucnosti díky ekonomickému tlaku budou i ostatní společnosti více přemýšlet nad tím, pro koho a jak své aplikace tvoří. Lidé mají zatím většinou pocit, že pokud nerozumí nějakému programu je to jejich chyba a slabost a ne problém tvůrce aplikace nebo zařízení.

1.2 Hypotéza a cíl práce

Mám za to, že sémiotika při tvorbě uživatelských rozhraní nástrojem, který má potenciál přinést do procesu návrhu interaktivních systémů nový pohled. Pokud by si ji designeři brali na pomoc, zcela jistě by to prospělo ke tvorbě přívětivějších uživatelských rozhraní. Zabránilo by se mnohým frustracím, které každodenně při užívání rozličných aplikací zažíváme a které plynou ze vzájemného neporozumění mezi designerem a uživatelem.

Problémem ovšem zůstává, že pojem sémiotika má pro mnohé stále suše akademický nádech a lze jen těžko IT veřejnost i odborníky přesvědčovat o tom, že by mohla sloužit k analýze rozhraní a podávat při jejich tvorbě užitečné návody. V prostředí teorie HCI se již za posledních pár desítek let stihla sémiotika etablovat a přináší cenné poznatky. Málokdy je ovšem používána v reálné praxi.

Problém převedení teoretických poznatků do běžné praxe je otázkou celého oboru HCI. Na trhu existuje jen málo společností disponujících dostatečnými zdroji na to, aby design interakcí mohli z procesu tvorby oddělit a nadřadit ho grafickému designu a samotnému programování. Ve většině malých firem, jak softwarových, tak web-designových se při návrhu jednotlivých interakcí spoléhá zejména na zkušenosti a intuici programátora. Málo který programátor má ale čas a vůli vzdělávat se v oboru HCI.

Částečnou podporu je v tomto ohledu schopno přinést UCD, které postupem času vytvořilo určitá vodítka a pravidla, kterými se ti, kdo sestavují rozhraní mohou řídit a zajistit tak svým uživatelům alespoň určitou míru komfortu užívání⁶ a to, aniž by bylo třeba vynakládat prostředky na zdoluhavý výzkum a testování. UCD sebou ovšem ze sémiotického pohledu nese určitá omezení (viz níže) spočívající mimo jiné i v přílišném lpění na zmíněných vodítkách a pravidlech, která postupem času zastarávají. Zdaleka také nevyčerpává celé spektrum problematiky HCI.

Ústředním tématem této práce je Sémiotické Inženýrství. Zvolil jsem ho proto, že se jedná o relativně mladý obor (První ucelená publikace vychází až v roce 2005.)

6 Např. Cooper 2007 s. 323 a dále

stavící na předešlé několikaleté tradici sémiotických výzkumů v oblasti HCI, ze které čerpá a dále ji rozvádí. Obsahuje nejenom teoretický základ, ale i poměrně rozpracovaný systém hodnocení komunikace s uživatelem na základně analýzy míst, kde selhává a analýzy způsobu, jakým je v rozhraní distribuována. Je schopno odhalit krizová místa a navrhnout řešení. Takzvaná SERG⁷ je také v posledních letech nejaktivnější skupinou zabývající se praktickým výzkumem v oblasti sémiotické teorie HCI.

Mám za to, sémiotické inženýrství je schopno poskytnout teoretický základ pro abstrahování určité jednoduché sady pravidel, kterými se mohou řídit designeři-programátoři pracující s omezenými zdroji a tvořit tak rozhraní, která sice nebudou v ohledu komunikace s uživateli dokonalá, ale budou se ideálnímu stavu minimálně přibližovat. Autoři Sémiotického Inženýrství sice sami kritizují pravidla vytvářená přístupem UCD⁸, jelikož jsou neflexibilní, rychle zastarávají a svým způsobem mohou i omezovat nové myšlenky při tvorbě rozhraní. Myslím si ale, že tyto problémy v případě pravidel vytvořených na základě SI⁹ nepřichází na pořad, jelikož se zaměřuje na fenomén komunikace, jejíž pravidla zůstávají v čase neměnná.

Pokusím se vyabstrahovat takovouto sadu pravidel z teorie sémiotického inženýrství a zejména z metod, které poskytuje k hodnocení schopnosti rozhraní komunikovat. Tato pravidla si samozřejmě nebudou dělat nárok na univerzalitu, jelikož SI samo nepokrývá celé spektrum problematiky HCI. Pokud se ale ukáže, že je opravdu možné takováto pravidla možné získat bude to zcela jistě znamenat výhodu pro běžnou praxi.

Cíle této práce jsou dva: ten užší je tedy uceleně popsat teorii sémiotického inženýrství spolu s jejími metodami a na základě tohoto popisu se pokusit získat sadu 15 – 20 základních pravidel, kterými by se mohli řídit designeři a programátoři v reálné praxi. Prokáže se tak, že sémiotické inženýrství neleží pouze ve sféře akademických diskusí, případně neslouží pouze k odborné analýze, ale může poskytnout nástroj, o který se může opřít běžný smrtelník. Druhým, širším cílem je poukázat na vhled, který může sémiotika přinést do tak technicky zaměřeného prostředí, jako je prostředí informačních technologií. Sémiotické teorie HCI nejsou žádnou novinkou (viz další kapitola). Většina jich ale zůstává pouze v uzavřené skupině teoretiků HCI. Nejbliže k praxi z nich má právě sémiotické inženýrství, kterému ale stále chybí ještě jeden krok k tomu, aby zcela překlenulo propast

7 Semiotic Engineering Research Group

8 DeSouza a Leitao 2009 s.1.

9 Sémiotické inženýrství

mezi sémiotickou teorií HCI a praxí.

2 Teoretická část

2.1 HCI

2.1.1 Obory HCI

Než budu pokračovat zmíním zde ve stručnosti některé možné přístupy k HCI, které se zatím stihly etablovat. Nepůjde o žádný vyčerpávající popis, ale spíše o ukázkou toho, jak je daný obor rozmanitý. Všechny níže uvedené přístupy k designu rozhraní spadají do obecné skupiny označované zkratkou IxD tedy interakčního designu.

- UxD (User Experience Design) – Hlavním měřítkem je zde pozitivní dopad na pocity uživatele.¹⁰ Jedním z teoretiků UxD je například Donald Norman, který ve své práci Emotional Design (2003) ukazuje důležitost lidského prožitku při užívání produktů. Podrobněji bude zmíněn níže.
- UCD (User Centred Design) – Tedy design zaměřený na uživatele. Vychází ze snahy co nejvíce „ušít“ rozhraní uživateli na míru podle jeho předpokládaných potřeb a cílů. Používá různé metody sběru dat a modelování uživatelů. Vývoj rozhraní je založený na základě následujících principů: 1) Vývoj je podřízen uživatelům. 2) Uživatelé na vývoji aktivně participují. 3) Tvoří se rané testovací prototypy. 4) Proces designu se neustále zhodnocuje a jednotlivé kroky se případně opakují. 5) Prolínání více disciplín. 6) Spolu s rozhraním se vyvíjí i vše co s ním souvisí (tréningové kurzy, nápověda)¹¹. Níže se budu podrobněji zabývat přístupem Alana Coopera.
- HCD (Human Centered Design) – Design zaměřený na člověka. Vymezuje se vůči UCD s tím, že UCD se příliš zaměřuje na rozhraní samotné a už ne na člověka. HCD se snaží soustředit na celkový kontext ve kterém člověk působí. Interakční design by se měl soustředit méně na technologie a postupy a více na člověka.
- Sémiotický přístup – Ačkoli by se mohlo zdát, že sémiotika nebude schopna poskytnout designu žádné konkrétní nástroje a bude setrvávat spíše v akademické sféře. Níže se, jak doufám, ukáže, že tomu tak z daleka není.

¹⁰ http://en.wikipedia.org/wiki/User_experience_design#The_design

¹¹ Gulliksen a spol. 2003

V následující kapitole krátce přiblížím v současnosti asi nejrozšířenější přístup k interakčnímu designu tedy Cooperovo pojetí designu zaměřeného na cíle, které spadá do skupiny UCD. Bude zde sloužit jako ilustrace toho, jak lze prakticky přistupovat k designu, tak aby odpovídal zásadám teorií HCI.

2.1.2 Cesta k rozhraní UCD

Velice důležitým požadavkem na HCI je schopnost připravovat programy přesně podle potřeb uživatele, proto Alan Cooper neustále zdůrazňuje zaměření na cíle uživatele (goal oriented design)¹². Přesně definovaný cíl uživatele je základním vodítkem pro redukci komplexity možností počítače a také něčím, co pomáhá sjednotit snahy celého pracovního týmu při tvorbě jednotného návrhu rozhraní. Cíle je zde třeba chápat poněkud širěji než na úrovni vykonávání jednotlivých úkolů v rámci rozhraní ("Teď chci uložit soubor.", "Teď se chci vrátit k předchozímu kroku."), jsou to obecnější cíle uživatele. Cílem spisovatele není ukládat a znovu načítat text, ale napsat knihu. Je tedy úkolem rozhraní postarat se o to, aby mu tento jeho cíl byl co nejjednodušeji umožněn.

Cooper dělí cíle na tři skupiny Prožitkové cíle (experience goals): jsou obecné cíle člověka. vyjadřují podle Coopera to, jak se člověk chce cítit při užívání programu¹³. (např. "Chci se cítit, jako profesionál.", "Nechci vypadat jako hlupák.") Konečné cíle (end goals): jsou cíle, které člověk má v souvislosti s užíváním daného produktu, jsou důvodem k tomu, aby ho vůbec použil. ("Chci si stáhnout hudbu.", "Chci napsat diplomovou práci") Životní cíle (Life goals): jsou celoživotní cíle člověka, které nejsou přímo spojeny s daným programem, ale jeho využívání je možné chápat jako jeden z kroků k jejich dosahování. ("chci být úspěšný.", "Chci vydělat peníze.", "Chci být v kontaktu se svými přáteli.")

Aby bylo možné vytvořit uživatelské rozhraní přesně na míru jeho potenciálním uživatelům, prochází Cooper složitým procesem modelování takzvaných person. Jsou to modelové uživatelské daného produktu, jejich vytvoření předchází řada rozhovorů s reálnými potencionálními uživateli a výzkumů, během nichž je naschromážděno značné množství dat, které jsou posléze v několika fázích zpracovány. Jejich výsledkem jsou pak modelové uživatelské s konkrétními zájmy a potřebami (tzv. persony). Důraz je zde kladen na realističnost jejich vykreslení.

S hotovými personami se poté tvoří scénáře použití daného rozhraní. Jedná se opět

¹² Cooper 2007

¹³ Cooper 2007 s. 92. – 97.

o komplexní scénář zabývající se nejen konkrétní interakcí se samotnou aplikací, ale i vnějšími okolnostmi, které ji provázejí. Jelikož z definice cílů vyplývá, že do hry nemusí vstupovat nutně pouze cíle, které jsou spojené s používáním konkrétního produktu. oslze se přistupuje k načrtání samotného rozhraní přičemž je neustále odkazováno k scénářům a personám, aby se tak zajistila koherence designu. Přesně podle výše zmíněných principů UCD.

2.1.3 UxD

Norman přichází ve své práci *The Design Of Everyday Things* (1998) s termínem Design zaměřený na uživatele (User Centered Design – dále jen UCD). Je to design zaměřený na uživatele takový, jaký byl popsán výše. Postupem času ale přehodnocuje svůj názor a zaměřuje se na prožitek člověka (tedy spíše perspektiva HCD) spojený s užíváním a přichází s termínem Emoční design (Emotional Design). Ukazuje, že v designu (ne nutně pouze designu uživatelských rozhraní) nejde pouze o funkčnost a efektivnost práce, ale z velké části i o pocit, jaký během užívání máme. Tvrdí, že emoce není možné od užívání oddělit.¹⁴ Popisuje tři roviny designu. Viscerální design: na této rovině je hlavním měřítkem vzhled a první kontakt s předmětem, nebo rozhraním. Pokud bychom si vedle sebe položily notebook Apple a HP, je zřejmé, který z nich bude více odpovídat této první rovině. Pak je rovina Behaviorální: v ní jde zejména o pocit z efektivnosti užívání. Program, který nám nebude klást zbytečné překážky do cesty a bude směřovat přesně k tomu, k čemu ho chceme použít, nám bude přinášet daleko větší uspokojení, než program ve kterém budeme muset neustále řešit úkoly netýkající se našeho cíle. Poslední rovinou je rovina reflektivního designu. Ta nesouvisí přímo s užíváním daného produktu, popisuje spíše, jak produkt zapadá do celkového kontextu života uživatele, co o něm říká. Například zavilý uživatelé Linuxu se ho neustále drží, i když zdaleka nemůže poskytnout takové možnosti, jako jiné systémy, přesto jsou s ním spokojeni, bez ohledu na občasné nástrahy v používání.

Dalším průkopníkem Designu zaměřeného na emoční prožitek (User Experience design) je J.J. Garret, který uvádí pět základních rovin Designu. Od té nejkonkrétnější po nejabstraktnější:¹⁵

- 1) Povrch: To, jak produkt ve finále vypadá. Jedná se o předmět zejména

14 Norman 2003

15 <http://www.jjg.net/elements/>

vizuálního designu. Konkrétní vzhled MS Word, barevné schéma, ikony atd.

- 2) Kostra: To, co nese samotný vzhled, jednotlivé prvky designu umožňující jeho funkce. To je tradiční problém oboru HCI, otázka funkcionality a interakcí. Jde už o konkrétní strukturu, která dává návod programátorům, jak postupovat při tvorbě kódu. Struktura jednotlivých menu, rozložení pracovní plochy.
- 3) Struktura: Návrh jednotlivých funkcí programu, nebo webové stránky, aby odpovídaly zamýšlené celkové funkci. Co všechno MS Word „umí“, editace textu, ukládání, vkládání obrázků.
- 4) Zaměření: Je mezistupněm mezi strukturou a strategií. Určuje jaké konkrétní vlastnosti by měl program splňovat na základě dané strategie.
- 5) Strategie: Obecné potřeby uživatele, čeho chce dosáhnout. Chce například psát diplomovou práci.

To jsou jednotlivé prvky uživatelského rozhraní podle J.J. Garreta.

2.1.4 komunikace

Všechny výše zmíněné teorie se zabývají způsobem, jak se dopracovat k co nejefektivnějšímu a pro uživatele nejjednoduššímu řešení. Jenom málo z nich se ale zabývá velice důležitou složkou tvorby rozhraní, kterou je způsob, jakým má komunikovat s uživatelem.

Je zřejmé že, srozumitelnost a použitelnost počítačového systému záleží na kontextu, ve kterém je použit a na výběru jednotlivých prvků na základě tohoto kontextu.¹⁶ HCI designér redukuje téměř neomezenou množinu možných funkcí počítačového systému a předkládá uživateli omezený set jeho možností, na základě toho, v jakém kontextu se právě uživatel nachází. Roli zde hraje i, jak se člověk při užívání daného produktu cítí (jak jsme viděli výše). Samotná komunikace s uživatelem ale není příliš samostatně tematizována. UI¹⁷ designéři zakomponovávají do svých návrhů nějaké formy komunikace s uživatelem, ať už ve smyslu různých dialogových informačních okem, tak ve smyslu samotného rozvržení a funkcí, kterým nějak uživateli sdělují, co je možné s daným produktem dosáhnout a jak. Rozhraní, jakožto sdělení ale vzniká spíše jakoby mimoděk a

¹⁶ Andersen 1991 s. 6.

¹⁷ User Interface – Uživatelské rozhraní

nikdo se na něj z tohoto konkrétního úhlu pohledu nedívá.

Sémiotické přístupy k uživatelskému rozhraní jsou tohoto pohledu schopné. Sémiotika umožňuje svými nástroji uchopit uživatelské rozhraní a analyzovat ho ze svého specifického pohledu. Nemůže pravděpodobně nahradit samotný proces designu, ale může velice dobře sloužit jako jeho doplnění.

2.2 Sémiotika

2.2.1 Proč sémiotika?

Vždy když přistupujeme k počítačovému rozhraní, setkáváme se s nějakými znaky, to znamená že, nedochází k přímé manipulaci jako v případě nástrojů před-výpočetní éry, ale manipulujeme vždy se znaky, které nám zastupují nějaké procesy odehrávající se uvnitř počítače a zároveň umožňují převést náš záměr do operací, které je schopen počítač zpracovat. Nemáme tedy před sebou svobodnou volbu výrazů, jako je tomu v případě přirozeného lidského jazyka. Naše volba je omezena na to, co je přístroj schopen zpracovat a co nám před-připravil tvůrce programu. Tato sada možností je nám sprostředkována skrze znaky rozhraní, pomocí kterých byla vytvořena. Rozhraní v sobě tedy obsahuje i sdělení o tom, co je možné s daným programem vykonat. Toto vše je nějak zakódováno do designem potažmo programátorem zvoleného souboru znaků.

Zdá se proto, že sémiotika je vhodným nástrojem pro studium rozhraní chápaného jako prostor, na kterém se odehrává komunikace mezi strojem a člověkem. Tedy spíše, jak říká teorie sémiotického inženýrství. Komunikace mezi designérem rozhraní a uživatelem, která probíhá zprostředkovaně pomocí počítačového programu.¹⁸

Samotnému oboru sémiotického inženýrství se budu věnovat podrobně až v následující kapitole, nyní ve stručnosti uvedu do relevantních sémiotických termínů a představím některé z hlavních sémiotických přístupů k HCI.

Sémiotická teorie HCI má několik specifíků, která jsou pro klasickou sémiotickou analýzu obrazových a textových sdělení nová. Je to zejména fakt, že v prostředí uživatelského rozhraní se nesetkáváme pouze se statickými obrazy a texty, ale důležitou

¹⁸ Viz např. De Souza 2005

roli tu hraje i změna, vzájemná interakce prvků, různé operace a procesy. Podle Petera Andersena je pro počítačovou sémiotiku specifické také to, že znaky označují, ne nějaké entity vnějšího světa, ke kterým se skrze ně můžeme vztahovat, ale entity, které jsou označujícím počítačem (potažmo jeho designérem) vytvářeny¹⁹. Tzn., designer rozhraní vytváří vždy nový systém znaků, který slouží pro označení něčeho, co vzniká paralelně s ním. Pro sémiotiku v tom tak spočívá nová výzva.

2.2.2 Krátké srovnání Saussura a Peirce

Sémiotická tradice se obecně dělí na dva základní proudy. Evropský proud vychází z lingvistické teorie F. Saussura a americký proud z o něco obecnější filosofické teorie Peircovi. V následující kapitole ve stručnosti představím obě teorie s důrazem na ty prvky, které jsou relevantní pro sémiotický přístup k rozhraní. Už předem je třeba se zmínit, že pro sémiotické inženýrství a obecně i pro ostatní sémiotické přístupy hraje větší roli teorie Peircova.

2.2.2.1 Saussure

Saussurova teorie vychází primárně ze zkoumání jazyka. Znak se podle něj sestává ze dvou částí označovaného a označujícího. Označující je tzv. image acoustic, tedy slovo tak, jak je vyřčeno. Označované je pak to, co znak označuje.²⁰ Není možné ho ale chápat jako nějakou materiální věc, je to spíše mentální obraz té věci v naší mysli. Slovo kladivo tedy nepředstavuje konkrétní objekt kladivo, ale mentální obraz kladiva v naší mysli.

Toto sebou nese zajímavé následky, jako je například tzv. Arbitrárnost znaku. Tj. Neexistuje žádný příčinný vztah mezi fyzickými objekty a znaky. Ty jsou objektům přiřazovány na základě konvence a na základě konvence také rozumíme tomu, co nám druzí říkají. V rozhraní počítačového programu neodkazují jednotlivé ikony přímo k procesům, které označují, ale spíše k mentálním obrazům designera, které si o těchto procesech vytvořil. Je proto také otázkou designera, aby byl skrze rozhraní schopen uživateli sdělit svoji vizi daného programu tak, aby byl uživatel tento program plně využívat podle designerova záměru.

19 Andersen 1997

20 Palek 1997 s. 10

2.2.2.2 *Peirce*

Peircovým záměrem nebylo jenom popsat strukturu jazyka, jak je je tomu primárně u saussurových zkoumání. Jeho projekt zahrnoval vytvoření celkového filosofického modelu, pomocí kterého by bylo možno uvažovat o mnoha různých fenoménech. Vybudoval systém kategorií, skrze který lze pohlížet nejen na jazyk, ale i na umění, vědecké zkoumání a další obory lidské činnosti. Sám Peirce původně svoji teorii nenazývá sémiotikou, ale logikou, respektive spekulativní gramatikou²¹.

Peircova teorie je postavena na tzv. Triadickém systému, z něhož vyplývá, i známé trojúhelníkové schéma znaku. Základními kategoriemi jsou prvost, druhost a třetost.

Znak má tři nerozlučné části: vehikulum, objekt a interpretant. Ty jsou navzájem provázány a to takovým způsobem, že vztah vehikula a objektu je předpokladem vztahu interpretantu a objektu, který je díky tomu ve vztahu i k vehikulu. Je důležité si uvědomit, že podle Peirce není objekt něčím primárním, co předchází oběma ostatním, ale objekt je tvořen vehikulem v procesu značení (tj. není objekt bez vehikula stejně tak, jako není znak bez interpretantu). Podstatné je, že nelze jednotlivé části znaku brát samostatně, ale vždy ve vztahu k ostatním částem.²²

K triádě znaku se ještě přidává triáda modalit bytí. Těmi jsou: 1) Prvost: Ta ve znakovém schématu odpovídá vehikulu v jeho materiální podstatě. Jakožto modalita bytí by se dala prvost přirovnat k ideji. Jedná se o první bezprostřední jsoucí zatím k ničemu nevztážené, na nic neodkazující. 2) Poté následuje druhost: Ta už předpokládá nějaký vztah. Odpovídá ve znakovém trojúhelníku objektu, respektive objektu ve vztahu ke znaku. Jedná se o formu bytí, která je k něčemu vztažena do souvislosti. Objekt, ne sám o sobě, ale mezi ostatními objekty. Zatím se ale stále jedná o mechanický přímý vztah. Něco na způsob vztahu příčiny a následku, do kterého nevstupuje žádná interpretace. 3) Poslední modalitou je třetost: Zde už vstupuje do hry něco třetí nějaký zákon, nebo obecný koncept, na jehož základě vzniká interpretant. Tedy máme první, které je samo o sobě, druhé které předpokládá výskyt dvou a třetí, které přináší do vztahu dvou něco třetího.

Toto rozdělení se dá ilustrovat na uživatelském rozhraní. Prvost na nás působí v převažující míře při prvotním pohledu na obrazovku, kdy nejsme schopni rozlišit funkci a účel jednotlivých prvků. Druhost přichází ke slovu při první interakci, když například absolutně mechanicky spojíme pohyb myši s pohybem ikony na obrazovce. Jakmile

21 Viz Charles Peirce *Gramatica Speculativa* in Palek 1997 s.29 a dále

22 tamtéž

začneme program k něčemu používat, nacházíme se v oblasti třetíosti. Ta předpokládá nějakou znalost případně alespoň nějaký záměr. Je zřejmé, že tyto jednotlivé fáze našeho zážitku nelze od sebe v čase a prostoru oddělit, už v této fázi umožňují základ nějaké klasifikace prvků na obrazovce s ohledem na jejich vzájemný vztah a vztah k uživateli. To, že nelze říci, že některé prvky by byly zásadně otázkou prvosti, některé druhosti a některé třetíosti vyplývá z povahy Peircovy klasifikace.

Odbočka: Jednotlivé modalities aplikuje Peirce na jednotlivé části znakového schématu z čehož mu vychází matice devíti základních pojmů, pomocí nichž pak tvoří jednotlivé třídy znaků. Nyní se budu ještě věnovat jednotlivým triádám tak, jak jsou ve svém vztahu k vehikulu, objektu a interpretantu definovány modalitami prvosti druhosti a třetíosti.

První tři kategorie se týkají samotného vehikula, bez explicitního odkazu k druhému, nebo třetímu. Prvním druhem znaku je Kvalisignum. Jak už název napovídá, jde o znak v podobě nějaké kvality. Kvalisignum je demonstrací prvosti ve znaku. Je to kvalita, která je sama o sobě a na nic se neváže. Lze si ji představit jako například „pocit červeného“, nebo třeba zápach. Sinsignum je pak znak, ve kterém už se odráží druhostní modalita bytí, je to znak tvořený nějakou existující realitou. Je zřejmé, že takovéto druhé jakožto znak v sobě už předpokládá nějaké první, jelikož konkrétní existence má vždy nějaké kvality. Třetím typem znaku je Legisignum. Tím je znak, který je zákonem, může to být například hvizd rozhodčího píšťalky, případně klepnutí soudcova kladívka pokud nám označují konec hry, nebo vynesení rozsudku. Je vidět že zde se jedná o nějakou konkrétní entitu, která má nějaké kvality, ale navíc je ještě spojena s něčím třetím podstatným pro porozumění.

Další tři kategorie vznikají pokud aplikujeme modalities prvosti druhosti a třetíosti na objekt, respektive na objekt ve vztahu ke svému prvnímu, kterým je vehikulum. Na první úrovni se jedná o ikon. Zde je vehikulum k objektu ve vztahu podobnosti. Tj. sdílí s objektem nějaké vlastnosti a tím ho denotuje. Je zde vidět, jak se jednotlivé kategorie začínají prolínat. Ikon musí být nutně do jisté míry Kvalisignem, jelikož sdílí s objektem kvality. Všimněte si, že populární označení ikona v počítačovém prostředí může být zavádějící, jelikož velice často ikony nenesou s tím, co označují žádnou podobnost. (O tom ale víc níže.) Index je objektem, který není se svým vehikulem v žádném přímém vztahu, ale pouze na něj nějak svojí povahou poukazuje. Důležité také je, že znak je svým

objektem nějak skutečně ovlivněn. Vztah je zde zatím spíše mechanický, bez kontaktu s nějakým třetím. To přichází do hry až v případě symbolu. Symbol je vehikulum, které nemusí mít se svým objektem nutně nic společného a je k němu vázáno až na základě nějaké konvence, nebo zákona. Už nyní by se dalo říci, že většina znaků tvořících obrazovku počítače jsou symboly, nejedná se ale většinou opět o čisté formy symbolů (tedy v případě grafického rozhraní), někdy značně převažuje indexiální, nebo ikonická složka. O všech výše zmíněných kategoriích i o posledních třech, které budou následovat je třeba mít vždy na paměti, že se nikdy nevyskytují v čisté formě. Spíše vždy jedna ze tří více či méně převažuje.

Poslední triádu tvoří kategorie odvozené od interpretantu, který má vztah k vehikulu a prostřednictvím něj i k objektu samotnému. Interpretant je cosi jako obraz, který vzniká o celku označujícího a označovaného v mysli interpreta. Dá se mluvit o jakémsi příslibu. Interpretant pojatý z pohledu prvosti, se nazývá Réma. Rématem je interpretant, který vzniká ze znaku jako příslib nějaké možnosti, kvality. Peirce říká že: „Snad každé réma poskytne nějakou informaci.“²³. Dicisignum je pak interpretant vycházející z druhosti, který příslibem nějaké konkrétní existence. Je zřejmé, že v sobě nutně obsahuje něco z rématu, jelikož každý příslib konkrétní existence v sobě také skrývá příslib nějaké kvality, kterou tato existence má. Nejsložitější kategorií interpretantu ve vztahu ke znaku a objektu je pak Argument. Znak je pro něj znakem zákona. Dalo by se říci jakýmsi materiálem pro práci rozumu. Peirce říká: „Argument je znak, který lze chápat tak, že reprezentuje svým charakterem svůj objekt jakožto znak.“²⁴ Tedy reprezentuje objekt jako nějaký materiál k myšlenkovému zpracování, kterým znak zcela jistě je.

Výše zminěná klasifikace je pouze hrubým náčrtem sloužícím k ilustraci toho, jak složitý je systém znaků, ke kterému peirce došel. Bylo by pravděpodobně možné namítat, že tento systém rozdělení nemá co do oblasti HCI přinést, ale opak je pravdou. Kategorie prvosti, druhosti a třetosti umožňují analýzu a následnou klasifikaci jednotlivých znaků v rozhraní, která posléze umožňuje tvořit o daném rozhraní závěry.

DeSouza například poukazuje na svou analýzu výukového programu Super Tangrams.²⁵ Přichází během výzkumu na to, že pokud jsou uživatelská rozhraní orientovaná primárně na třetost, tedy na manipulaci s nějakými obecnými koncepty. Jsou ve své funkci daleko účinnější, než rozhraní orientovaná např. Na prvost, tedy výhradně na

23 Ch. S. Peirce in Palek s. 44

24 Ch. S. Peirce in Palek s.45

25 De souza 2005 s.52.

přímou manipulaci. Je tedy na základě těchto poznatků a výše načrtnuté klasifikace možné analyzovat jednotlivé prvky rozhraní a posléze činit o celku rozhraní závěry typu: pokud se v rozhraní nachází většina prvků orientovaných na třetí, bude toto rozhraní úspěšně podporovat výukovou funkci programu atd.

Je ovšem nutné si stále uvědomovat, že samotné rozhraní je pouze zprostředkovatelem (a pro teorii sémiotického inženýrství to platí dvojnásob²⁶) obsahu, který je nám tvůrcem programu skrze ně posílán. Podrobná sémiotická analýza jednotlivých znaků rozhraní je tedy pouze jednou z částí komplexního pohledu.

2.2.2.3 *Interpretans*

Interpretans je pravděpodobně tím nejobtížnějším pojmem peircovi sémiotiky. Je pomyslným bodem, ve kterém se střetává označující a označované a všechny tři pak dohromady spolu tvoří jednotu znaku. Přináší sebou důraz na interpretujícího, se kterým ale nemůže být v žádném případě slučován. Interpretans je možné definovat jako znak v mysli interpretujícího. Je výsledkem našeho přímého střetnutí se znakem.

U Saussura pokud slyším slovo pes, spojí se mi s mentálním obrazem psa v mé hlavě a proto vím, na základě nějaké dané konvence, že se mluví o psovi. V případě peircovi teorie, pokud slyším pes má toto označující (representamen) vztah k nějakému označovanému (objektu), který je v tomto případě sice také konvenční (jak vyplývá s peircovy širší klasifikace znaků²⁷), ale "finální produkt" vzniká až díky spojení předchozích dvou ve vztahu k interpretans. Je vidět, že striktně konvenční spojení nikdy neexistuje a vždy záleží na finálním vztahu, který vše završuje.

2.2.2.4 *sémióza*

Výsledný znak v podobě interpretans v mysli interpretujícího má potenciál se stát základem nového znaku, označovat nový objekt a vytvářet opět nový interpretans, který má opět potenciál stát se základem znaku atd atd. Tento proces se někdy nazývá nekonečnou sémiózou²⁸. Je něčím, co známe z každodenního života, jeden znak je schopný v nás vyvolat řetěz různých asociací.

Pojem sémiózi vnáší do sémiotiky dynamický rozměr. Znak je v rámci probíhající semiozy neustále-se vyvíjející dynamickou entitou. Tento pojem je pro sémiotické

26 Viz níže

27 Ch. S. Peirce in Palek s.40.

28 Viz Eco 2004

inženýrství velice důležitý jelikož popisuje způsob, jak o věcech uvažujeme v běžném životě. To znamená i způsob, jakým nakládáme s uživatelským rozhraním.

S pojmem *sémiosis* je úzce spjat pojem *abdukce*.

2.2.2.5 *Abdukce*

Abdukce je spolu s dedukcí a indukcí metodou logického uvažování.

V případě dedukce vyvozujeme jednotlivé závěry z obecného pravidla. Tj. pokud například předpokládám pravidlo, že všichni obyvatelé Pekingu jsou Číňani a mluvím s jedním z obyvatel Pekingu, budu dedukovat, že je Číňan. Co se týče indukce je postup obrácený. Obecné pravidlo vzniká až na základě jednotlivých pozorování. To znamená, že pokud jsem v kontaktu s několika lidmi z Pekingu a všichni jsou Číňani, mohu na základě toho usuzovat, že všichni obyvatelé Pekingu jsou Číňani. Je zřejmé, že zde hraje roli i jistá forma pravděpodobnosti.

Peirce poukazuje na třetí způsob logického vyvozování. Takový, který je nám pravděpodobně nejbližší. Abdukce dle Peirce je "metoda tvoření obecné predikce bez pozitivní jistoty, že predikce bude úspěšná, ať ve zvláštním případě nebo obvykle, přičemž oprávnění této metody spočívá v tom, že je to jediná možnost, jak racionálně řídit naše budoucí chování a že indukce z minulé zkušenosti v nás vzbuzuje naději, že predikce bude úspěšná i v budoucnosti"²⁹. Celý tzv. *Abduktivní cyklus*³⁰ tedy funguje následujícím způsobem.

Na základě pozorovaného výsledku nějaké činnosti si tvoříme hypotézu o obecně fungujícím pravidle. Pokud naše další pozorování toto pravidlo potvrzují, zůstává i nadále platné. Pokud se ovšem objeví nějaký případ, který hypotézu nepodporuje, jednoduše si dané pravidlo přetvoříme tak, aby vyhovovalo a tato hypotéza pak platí do té doby, dokud není opět vyrácena a následně modifikována. To platí i pro způsob, jakým běžně používáme počítačové programy, jelikož málokterý uživatel si dopodrobna přečte uživatelský manuál tak, aby věděl, co přesně která funkce provádí. Spíše si neustále tvoří úsudky o funkci jednotlivých částí programu a ty si buď potvrzuje úspěšným užíváním, nebo vyvrací na základě chyb. Je proto velice důležitá bezprostřední komunikace funkcí

29 PALEK, Bohumil, *Role abdukce v procesu nového poznání*
(<http://is.cuni.cz/studium/predmety/index.php?do=down&did=9388>)

30 Viz. DeSouzza 2005 s.43.

rozhraní skrze ně samotné, aby byl abduktivní způsob užívání co neefektivnější. To je také velice podstatný prvek pro celou teorii sémiotického inženýrství.³¹

Přiblížit si to můžeme na triviální příkladu z uživatelského rozhraní:

Na základní ploše systému MS Windows se nachází ikona koše. Uživatel je zvyklý na ni tažením přemísťovat ostatní ikony a tímto způsobem je mazat. Jeho pracovní hypotéza je následující: "Stejně jako nepotřebný odpad, mohu vyhodit wordový dokument do koše a ten už přestane zabírat místo na mém stole (na mém harddisku) tak, jako když zmačkám papír s poznámkami a odhodím do koše." Pak ale přijde situace, kdy potřebuje uvolnit větší množství na disku a přetáhne na ikonu koše například rozměrný soubor s videem. Očekává pak na základě své stávající hypotézy, že se mu uvolní dostatečné množství místa, aby mohl soubor nahradit jiným. Zjistí ale, že se tak nestalo a místo se neuvolnilo. Po chvíli zkoumání dojde k tomu, že soubor není ve skutečnosti nenávratně zdeformován a vyhozen, ale pouze odložen do koše a je možné ho, narozdíl od zmačkaného roztrhaného papíru ve skutečném koši, vytáhnout v neporušené formě. Upraví tedy svoji hypotézu o mazání souborů na: "Nepotřebné soubory mohou odstranit odsunutím do koše, neztrácím je tak ale na vždy, mohu se k nim kdykoli bez obav vrátit." Právě vytvořená hypotéza mu může dát důvod k tomu začít používat koš jako způsob zálohy dokumentů, které takto sice zmizí z očí, ale budou na druhou stranu vždy po ruce. Pak ale radikálně dojde místo na disku a systém se uživatele zeptá, zda si přeje uvolnit místo smazáním nepotřebných souborů a složek. Pokud uživatel žádost systému schválí, dojde i k vysypání koše s ním i celého "uschovaného" obsahu. Hypotéza se tak opět změní.

2.2.2.6 Umberto Eco – Teorie znakové produkce

Výše zmínění autoři jsou zakladatelé sémiotické teorie obecně a mnohé sémiotické přístupy v nich nacházejí svůj teoretický základ. To platí i pro Sémiotické inženýrství. Jeho přístup k uživatelskému rozhraní a znakům v jeho rámci je ale vybudovaný zejména na základě Teorie znakové produkce Umberta Eca³². Eco se zabývá tím, jak se znaky uvnitř společnosti rodí a reprodukuje. Ukazuje že kultura je neustálý znakový proces.³³

Základními pojmy jsou pro něj signifikace a komunikace. Signifikace je procesem neustálé kulturně podmíněné tvorby asociací mezi obsahy a výrazy. Důležitý je zde poukaz na kulturní podmíněnost. Jak Eco říká, neexistuje nic jako čistý peircův ikón, kde by

31 Viz níže

32 De Souza 1993 str. 754

33 Eco 2004

označované přímo odpovídalo označujícímu, tento vztah je vždy kulturně daný.³⁴ Nebo například Index není nikdy přímým následkem označovaného. To, že mu rozumíme jako znaku něčeho, vždy pramení z našeho kulturního kontextu. Červená tečka na kůži pacienta není automaticky indexem jeho nemoci. Lékař musí mít nejprve nějakou znalost před tím, než ji jako znak nemoci pochopí. Pro neznalého to je prostě tečka.

Dalším podstatným pojmem je komunikace. Při ní dochází k používání znakových systémů ve společnosti. To neprobíhá výhradně v souladu s těmi to systémy. Dochází neustále k jejich modifikaci, porušování jejich pravidel a tvorbě systémů nových. Je zřejmé, že podle Eca není možné znaky jen tak zařadit na základě nějakého přesně daného systému, ale že jsou dynamickým systémem, který podléhá neustálému vývoji. To je prvek, který je zejména pro teorii sémiotického inženýrství velice podstatný (viz. níže).

Eco představuje i svoji klasifikaci jednotlivých modů produkce znaků ve společnosti.

- 1) Typ „práce“ vykonané při produkci znaku – od pouhého rozeznání po vytvoření nového znaku. Při komunikaci tak znak pouze vyberu ze zásobárny, která je mi dána kontextem a kulturou, nebo vytvořím zcela nový. Psa tak můžu pojmenovat „pes“, nebo třeba „štěkač“.
- 2) Vztah mezi typem a užitým tokenem – variuje mezi tzv. „ratio facilis“ tedy případem, kdy existuje nějaký pevně daný a zažitý vztah typ-token. Tedy například slovo pes označující psa. A „ratio difficilis“ což je případ, kdy neexistuje nějaká daná konvence vztahu mezi tokenem a typem. To může být například u mapy, každý tvar v krajině je třeba označit individuálně.
- 3) Typ kontinua – Pokud je výraz stejné povahy (stejně látky) jako obsah, mluvíme o homomateriálním kontinuu. Pokud je to naopak, jedná se o heteromateriální kontinuum. Například slovo „slovo“ je příkladem výběru z homomateriálního kontinua, jelikož označuje něco, co je stejné povahy, jako ono samo. Slovo „žena“ je příkladem tvorby znaku z heteromateriálního kontinua, jelikož ono slovo nemá s reálnou ženou nic společného.
- 4) Modus a komplexita artikulace – charakterizuje stupeň gramatizace znaků. Variuje od systému, který má přesně daná pravidla kombinace po systém, kde jsou možnosti kombinací otevřené. Příkladem prvního je přirozený

34 tamtéž

jazyk, kde jsou pravidla tvorby jednotlivých prvků přesně daná. Jeho opakem je například výtvarné umění, kde je tvorba jednotlivých znaků otevřena lidské fantazii a nic ji v zásadě nesvazuje.

2.3 Sémiotické přístupy k HCI

Sémiotika, jakožto nauka o znacích, které Peirce trefně definoval jako něco, co zastupuje něco pro někoho v nějakém určitém ohledu³⁵, se jeví jako velice užitečný nástroj pro zkoumání rozhraní. Uživatelské rozhraní není totiž nic jiného, než soubor znaků v různých vztazích a hierarchických zařazeních, které něco znamenají pro uživatele v kontextu jeho užívání daného programu. V následující části provedu krátký exkurz do sémiotických přístupů k teorii HCI.

První z teoretiků Peter Andersen je guruem ve svém oboru a bude mu proto věnován největší prostor. Přináší ucelenou sémiotickou tematizaci HCI a ukazuje, jak může sémiotika oboru HCI sloužit. Jeho pojetí je možné chápat jako základ pro ostatní sémiotické přístupy. To samozřejmě neznamená, že by z něj všechny ostatní slepě vycházeli. Další Mihai Nadin je pak ukázkou až radikálně sémiotického postoje. Říká, že sémiotika je jedinou možnou teorií designu³⁶. Mezi ně jsem vložil náhled do práce Friedra Nakea a Susane Grabowski. Jejich hlavním přínosem je poukaz na dvojí povahu počítačového znaku (znak a signál) a jeho dopad pohledu na obraz komunikace v HCI.

Všemi zmíněnými přístupy se táhne jako červená nit' myšlenka komunikace a, jak níže uvidíme, je zde možné nalézt některé základní myšlenky, které využívá sémiotické inženýrství. Účelem této kapitoly je ukázat, že sémiotické inženýrství není osamoceným oborem, ale je začleněno do širokého proudu sémiotické tematizace HCI, která má zcela jistě co říci a přináší nové úhly pohledu zůstávající tradičním přístupů skryté.

2.3.1 Andersen

Peter Andersen ve svém článku "What semiotics can and cannot do for HCI"³⁷ označuje sémiotiku jako matematiku humanitních věd. Sémiotika podle něj poskytuje analytický nástroj pro pojmenovávání stejných fenoménů (znaků) v různých humanitních oborech (literatura, výtvarné umění, HCI). Umožňuje tak navzájem tyto jednotlivé obory

35 Ch. S. Peirce in Palek 1997 s.37

36 Nadin 1988 s. 269

37 Andersen 2001

porovnávat a spojovat podobně, jako matematika tvoří společný základ oborů přírodních. Poukazuje na to, že právě pro tuto svoji vlastnost je sémiotika ideálním nástrojem pro HCI³⁸, jelikož zkoumání interakce počítače a člověka v sobě spojuje mnoho teorií humanitních (ale i exaktních) věd. Sémiotika by v HCI neměla sloužit pouze jako nástroj analýzy a hodnocení, ale i jako nástroj pro návrh nových funkčních řešení. To je možné chápat i jakési ospravedlnění vůči možným námitkám o čité akademičnosti sémiotických teorií.

Hlavní přínosy sémiotické teorie pro problematiku HCI jsou dle Andersena následující:³⁹

- 1) Obor se stává mnohem soudržnějším. Od počátků se v teorii HCI střetávají 2 modely: model systémový (tzn. model programátorů a výzkumníků v oboru výpočetní techniky) a model uživatelský (tzn. model psychologů, kognitivních vědců atd.). Sémiotika je schopna nacházet tam, kde se jedná o lidskou interpretaci, spojení mezi těmito dvěma modely. Dokáže stejnými termíny popsat jak systém reprezentace, tak i systém interpretace toho, co je reprezentováno. Pomáhá chápat programy a systémy v rámci procesů komunikace. Pro sémiotické inženýrství, které je hlavním tématem této práce, je tento faktor velice podstatný.
- 2) Přináší do oboru HCI poznatky i z jiných médií. Andersen poukazuje na to, jak je možné například využít letité poznatky filmového průmyslu.
- 3) Kromě toho, že sémiotika spojuje jednotlivé obory, dokáže i klást důraz na vlastnosti, které jsou oboru HCI charakteristické. To, co odlišuje znaky HCI a znaky například filmu, nebo fotografie je interakce. Jednotlivé znaky nejsou interpretovány jenom na základě toho, co pro uživatele znamenají, ale i na základě toho, jak se navzájem ovlivňují a jaké tím uživateli přinášejí možnosti.
- 4) Jedním z Andersenových vyzdvihovaných přínosů pro oblast HCI poznatek, že lidé jsou tzv. "nutkaví interpretátoři" a "nutkaví mluvčí"⁴⁰.
 1. Lidé si nemohou pomoci, musí neustále interpretovat nejen to, co vidí, ale i procesy probíhající v pozadí bezprostředně vnímaného. Andersen

38 Andersen 2001 s.1.

39 tamtéž

40 Andersen 2001 str. 4.

poukazuje na to že, v osmdesátých letech kdy se počítače postupně přesunuli z čistě vědeckého prostředí do prostředí kanceláří, bylo nutné nějak lidem ukázat přátelštější a srozumitelnější stránku programů (to stojí za vznikem HCI) a některé procesy skrýt. Běžní uživatelé si ale neustále vytvářejí nějaké představy o tom, co se děje na pozadí a to přímo ovlivňuje jejich práci s daným softwarem. Vede to také k technické dělbě práce, kdy jsou uživatelé závislí na technících a jejich vědomostech. Andersen říká, že díky sémiotice je možné se poučit z jiného komunikačního systému (například z jeho schopnosti sebe-reference⁴¹), který existuje tisíce let a sám se udržuje bez vnějších zásahů odborníku. Tím systémem je samozřejmě jazyk.

2. Druhou stránkou věci je poznatek, že lidé a systémy, které si vytvářejí. Nejsou schopni existovat bez neustálé komunikace.⁴² Pro většinu organizací jsou médii komunikace výpočetní systémy. Andersen říká, že to je důvod, proč musí být takovéto systémy komunikovatelné.⁴³ To jinými slovy znamená, že znaky počítačového systému musí odpovídat znakům, které se vyskytují v okolí ve kterém se počítačový systém nachází, aby tak bylo možné plynule přecházet od komunikace v rámci systému ke komunikaci mimo něj. To je důraz, který se opakuje i v dalších teoriích HCI poukazujících na to, že rozhraní nelze chápat odděleně od prostředí, ve kterém se nachází.

41 Neplést z Luhmanovým pojmem

42 Viz například i Luhman: Nejmenší jednotkou sociálního systému je komunikace

43 Andersen 2001 str. 5

2.3.1.1 *Počítačová sémiotika*

Ve své práci o počítačové sémiotice⁴⁴ ukazuje Andersen počítače jako znakové systémy a definuje počítačové znaky. Počítače jsou v rámci počítačové sémiotiky chápány ze specifického úhlu pohledu jako znaky, které uživatel interpretuje tak, že pro něj něco znamenají.⁴⁵ Jasně zde říká, že sémiotika se nemůže zabývat samotnými daty, která jsou obsahem programu, ale spíše znaky rozhraní a jejich schopností sloužit jako přenašeče významů. Toto je také jeden z prvků, na kterých de Souza staví své sémiotické inženýrství. Je třeba se zde alespoň ve stručnosti zabývat Andersenovou teorií, jelikož staví základy něčeho, co je posléze v SI použito jako základ a dále rozvíjeno.

Andersen se nezastavuje pouze u toho, že by sémiotiku aplikoval na samotné rozhraní. Říká, že počítače jsou tzv. "symbolické stroje" (symbolic machines), tedy stroje zpracovávající symboly⁴⁶. Tvorba a interpretace znaků probíhá v počítačích na několika úrovních. Uživatelské rozhraní je souborem znaků pro uživatele. Pod ním je ovšem také nějaký kód. V případě webového rozhraní jsou to například nejrozšířenější PHP, JavaScript, HTML, CSS. Všechny tyto kódy jsou také soubory znaků jelikož jsou nějak interpretovány (HTML, CSS a JavaScript prohlížečem, jiné programovací jazyky jako je C# nebo JAVA kompilátorem). Internetový prohlížeč v sobě má zanesena pravidla, jak má dané znaky interpretovat a tvořit z nich znaky pro uživatele. Programátorovi stačí pohled na zdrojový kód a je schopen si v hrubých obrysech představit, jak webová stránka vypadá a jaká je její funkce. I pro něj je programovací jazyk, jazykem tvořícím znaky stejně, jako je pro uživatele soubor znaků rozhraní (posuvníky, kurzory, tlačítka ...). Jazykem, který pro něj tvoří vyšší celky znakového systému. Takto je možné se dle Andersena dostat skrze jednotlivé úrovně znakových systémů až na úroveň strojového kódu. Ten má opět svého tvůrce (kompilátor strojového kódu) a svého interpretátora, tedy procesor počítače.

Obecně tento pohled shrnuje Andersen do následující teze: "Počítačový systém je možné chápat jako komplexní síť vztahů, kde každá úroveň v sobě obsahuje prvky, které je možno sémioticky zpracovávat."⁴⁷ Tento přístup také pomáhá překlenout bariéru mezi uživatelským a programátorským pohledem, jelikož ukazuje, že k oběma lze přistupovat s využitím stejné teorie.

44 Andersen 1997

45 Andersen 2001 str. 4.

46 Andersen 2001 str. 6.

47 Andersen 2001 str. 7.

2.3.1.2 Počítačový znak

Při definici znaku vychází Andersen nejprve ze Saussurovské strukturalistické tradice. Podle té má znak dvě stránky: Stránku výrazu a stránku obsahu. Při tvorbě znaku dochází k vydělení výrazu z výrazového kontinua. Tj. z jakési beztvare masy, která je součtem všech možností tvorby výrazů. Výraz je pak spojen s určitou částí kontinua všech možných obsahů, která je tím vydělena od zbytku.⁴⁸

V případě výpočetních systémů je jako výrazové kontinuum možné označit paměť počítače. Sama o sobě nenesे žádný obsah a ani netvoří žádný konkrétní výraz. Je potřeba práce programátora, aby beztvare mase pomocí kódu vtiskl nějaký tvar. Andersen říká, že takto je možné tvrdit, že už samo programování je tvorbou znaků. "Programování je vskutku tvorbou znaků, jelikož prvkům vnucuje formu."⁴⁹

Obsahové kontinuum tak může být na příklad v případě řízení letového prvozu suma pasažérů, zavazadel, letadel a letů. Z této sumy je pak vybrána konkrétní část odpovídající potřebám člověka, který s těmito daty pracuje. Jiná část je programátorem "vypíchnuta" pro potřeby pilota a jiná pro potřeby odbavujícího pracovníka na letišti. Nikdy není možné oddělit daný interface od jeho funkce.

Andersen vytváří i jakousi základní klasifikaci počítačových znaků. Všimá si jejich vlastností a vyděluje tzv. ovládací rysy (handling features), což je například stisk tlačítka, který vysílá signál procesoru. Dálšími jsou permanentní rysy (permanent features). To jsou ty vlastnosti znaků, které zůstávají v průběhu času neměné, tedy například vzhled ikony na ploše, sekvence znaků v řetězci atd. Posledními jsou přechodné rysy (transient features). Ty se nějak mění v okamžiku kdy je znak použit. Například změna barvy tlačítka při stisknutí, nebo změna kurzoru myši při najetí na aktivní místo webové stránky.

Je zde také rys činnosti resp. akce, všechny počítačové znaky jsou obohaceny o tento rozměr, jelikož všechny nesou ve vztahu k jiným nějakou akci. Například text psaný ve wordovém dokumentu se vztažením k jiným znakům může změnit na tučný a podobně.

Celá klasifikace pak spočívá na tom: a) Jaký z rysů převládá v daném znaku. b) Jaké akce může znak provést tak, aby ovlivnil rysy ostatních znaků. b) jaký je možný rozsah významů jednotlivých znaků. Díky tomuto analytickému nástroji je pak možné vydělit. Jednotlivé druhy znaků a to: Interaktory, tlačítka, aktory, kontrolery, objekty, duchy

48 Andersen 2001 str. 13.

49 tamtéž

(Ghosts) a layouty.⁵⁰ Jejich definice zde není nutné vyjmenovávat, jelikož leží mimo rámec této práce. Náčrt Andersenovi klasifikace slouží spíše jako ukázka toho, jak je možné pomocí sémiotické analýzy přistoupit k rozhraní a rozebrat ho na jednotlivé části tak, aby bylo možné tvořit o něma nějaké závěry a navrhnout řešení.

Stejně jako de Souza⁵¹ říká i Andersen, že evropská strukturalistická tradice je pro důkladnou analýzu počítačových znaků nedostatečná. Až zapojení peircovy koncepce přináší důležitý aspekt, a to fakt, že všechny počítačové znaky jsou vytvořeny pro někoho. Nestojí sami o sobě, ale jsou znaky vždy jako něco znamenající pro někoho v nějakém ohledu⁵². Zde je důležité si uvědomit, že něco znamenají pro programátory, ale nutně jsou jimi tvořeny tak, aby něco znamenali i pro jejich běžné uživatele. Proto je nutné nějak překlenout propast mezi programátorem a uživatelem. To je přesně to, k čemu je sémiotická analýza vhodná. Tento aspekt je důležitý i pro celé sémiotické inženýrství, jelikož komunikace mezi programátorem a uživatelem je zde základním tématem, jak bude ukázáno níže.

Při tvorbě rozhraní je také třeba dbát na to, že počítačové znaky se musí vstahovat nejenom k dalším znakům v rozhraní, ale i ke znakům mimo něj. To znamená znakům, které jsou součástí pracovního prostředí. Lidé užívající počítačový systém jako nástroj k práci musí být schopni bez obtíží přecházet mezi prací s rozhraním a komunikací se svými kolegy o své práci s rozhraním.

2.3.2 Další sémiotické přístupy

Nyní ještě ve stručnosti k některým dalším sémiotickým přístupům k HCI. Nejsou to samozřejmě jediné, slouží zde jako příklad dalších možných. Na závěr nám také umožní před-připravit půdu pro základní hypotézu sémiotického inženýrství.

2.3.2.1 Nake a Grabowski

Základním přínosem práce Friedra Nakea a Susanne Grabovwski je poznatek, že není jakkoli z pohledu sémiotiky možné srovnávat, nebo dokonce stotožňovat, procesy uvnitř lidského mozku s procesy odehrávajícími se uvnitř počítače.⁵³ Přiznávají, že jsou zde

50 Andersen 2001 s.15.

51 Viz níže

52 Viz výše uvedená peircova definice znaku

53 K tomuto tématu například i Searle 1994

určité rysy, které vedou k tomu, že jsou někteří vědci přesvědčeni o fungování analogií mezi myšlenkovou prací člověka a výpočetní prací počítače. Vzali si proto za úkol prozkoumat, v čem spočívá tato povaha a proč je pro některé lidi natolik matoucí.⁵⁴

Výše jsem přiblížil pohled P.B. Andersena, který říká, že hlavním přínosem sémiotiky k HCI je fakt, že je schopna sblížovat světy programátorů a uživatelů⁵⁵. Pohled na práci Nakea a Grabowski ovšem připomene, že toto propojení není absolutní a nelze zcela zaměňovat znaky, se kterými pracuje člověk se znaky, jež zpracovává počítač. Na základě tohoto fundamentálního rozdílu mezi počítačem a člověkem zavádějí Nake a Grabowski termín pseudo-komunikace pro pojmenování určitého specifického druhu komunikace, která vlastně komunikací v pravém slova smyslu není. Ukazuje se tak, že jediná možná komunikace se může odehrávat mezi lidmi a nikdy ne mezi člověkem a strojem. Ten zde slouží pouze jako prostředník (Z toho například vyplývá i specifický rys sémiózy, tak ji popisuje sémiotické inženýrství. O tom ale níže.)

Definují tzv. Algoritmický znak, což je znak, který může být na jedné straně zpracován počítačem, ale na druhé straně může být i znakem pro člověka. Na straně člověka je znak interpretován jako každý jiný znak. Tato jeho stránka je nazývána "znak jako znak." Na straně počítače ovšem k nějaké interpretaci závislé na momentálním rozpoložení, kontextu a stavu interpretujícího nedochází. Každý znak zde zapadá do předem daných funkcí naprogramovaných tak, aby s ním předem daným způsobem zacházeli. Je to "je to znak jako signál."⁵⁶

Počítačový znak (tedy signál) není možné definovat na základě standartního peircova sémiotického trojúhelníku. Podle autorů je redukován na svoji materiální podstatu, tedy vlastně pouze na representamen. Pokud bychom měli srovnat navzájem dva trojúhelníky definující klasický znak a znak jako signál, tak na straně reprezentamenu se v zásadě nic nemění. Materiální podoba je platná pro oba znaky. Pro člověka označuje representamen nějaký objekt, pro počítač však spíše operaci. Třetí část trojúhelníku, tedy tzv. Interpretans je u člověka tzv. intencionálním interpretantem, odráží se v něm nějaká intence interpretujícího a řetězec sémioze, do které je zapojen. U počítače mluvíme spíše o kauzálním interpretantu. Je přesným a přímým důsledkem nějakých předem daných okolností. Sám je pak součástí dalšího řetězení a větvení, které je výsledkem záměru programátora aplikace.

54 Nake a Grabowski 2001 str. 441

55 Viz výše

56 Nake a Grabowski 2001 str. 442

Kauzální interpretant a intencionální interpretant jsou tím, co od sebe navzájem vyděluje znak a signál. HCI je tak ze sémiotického pohledu možné chápat jako "párování dvou nezávislých a přesto k sobě vztažených procesů: plnohodnotného lidského zpracování znaků, které se odehrává spolu s omezeným počítačovým zpracováním signálu.⁵⁷". Lidé a počítače jsou součástí jednotného systému (human-computer system), uvnitř kterého dochází k velice specifické výměně mezi dvěma nesourodými partnery. Počítač ze své povahy nemůže jinak než kalkulovat a člověk naopak nemůže jinak, než interpretovat (více viz výše zmíněný Andersenův termín *compulsive interpreters*).

Mezi počítačem a člověkem nedochází tedy ke komunikaci, není to z principu možné. Dochází spíše k čemusi, co Nake a Grabowski nazývají pseudokomunikací. Je to něco co se zdá na povrchu vykazovat znaky běžné komunikace, až při hlubším pohledu zjistíme, že klasická komunikace se zde nemůže ani odehrávat.

2.3.2.2 Mihai Nadin – Sémiotické paradigma

Nadin je jedním z teoretiků HCI, kteří UI⁵⁸ chápou zejména jako prostředek mezilidské komunikace. Jeho základním východiskem při přístupu k HCI je až radikální tvrzení, že sémiotika je nutně základem jakéhokoli designu a tedy i designu v HCI.⁵⁹ Při každém designu dochází k uspořádávání jednotlivých znaků a jejich začleňování do vzájemných vztahů tak, aby tvořily celek. Důkaz je možné nalézt i v samotném slovu design (sign tedy Anglicky znak). Ze dvou hlavních sémiotických proudů preferuje Nadin, stejně jako většina ostatních, spíše ten peircovský, jelikož, jak už bylo zmíněno výše, se jedná o komplexní logický systém, který se neorientuje jenom na problematiku jazyka. Samotné rozhraní se neskládá pouze z textů, ale i z obrazů a podle Nadinova názoru i z ostatních prvků zprostředkovávajících člověku přístup k funkcím počítače⁶⁰. Je zde také podstatný pojem interpretantu, jelikož se klade důraz na fakt, že rozhraní je vždy tvořeno pro někoho, kdo ho nějak interpretuje a zachází s ním (uživatel).

Základním Nadinovým termínem je rozhraní (tedy interface). Chápe ho ale v širším smyslu, než v jakém je obecně teorií HCI přijímán. Definuje rozhraní následujícím způsobem: "Rozhraní je místem, kde se střetávají dvě odlišné entity, aby zde přišly v

57 tamtéž

58 User Interface – Uživatelské rozhraní

59 Nadin 1988 str. 269.

60 Nadin ve svém článku z roku 1988 na několika místech zdůrazňuje, že pojem rozhraní zahrnuje i periferie počítače a jeho okolí

kontakt, byly přivedeny k sobě – to znamená, aby komunikovali.“⁶¹ Je zřejmé, že stejně jako u předchozích autorů je zde kladen důraz na pojem komunikace a i zde se jedná hlavně o komunikaci člověka s člověkem skrze rozhraní a ne člověka se strojem. Nadin přímo zmiňuje, že rozhraní je místem, kde se spolu setkávají designér a uživatel.⁶² Rozhraním je tak zprostředkována komunikace mezi designérem a uživatelem. Tato myšlenka je zásadní i pro teorii Sémiotického inženýrství, jak uvidíme níže.

Jedná se tedy o přenos sdělení z jednoho na druhé tedy o nějakou mediaci. Rozhraní je podle Nadina médiem a sémiotika obecně je teorií mediace⁶³. V designu rozhraní spočívá mediace ve dvou úrovních. Je tu úroveň exaktní, vědecká (programování, výsledky kognitivní vědy, fyziologie) a také úroveň umělecká, na které se do rozhraní přináší vždy něco nového. Někjaký originální nápad designera, který (designer) se tak v rozhraní nějak projevuje. Sémiotika je pak schopna propojení obou dimenzí pomocí stejné sady nástrojů (viz Andersen).

Širší pojetí rozhraní vyžaduje komplexnější přístup k jeho tvorbě. Není možné se omezit pouze na návrh výstupů a vstupů, aby plnily požadovanou funkci. Je nutné dbát i na konzistenci a logickou provázanost jednotlivých prvků a to samotným ovládacím zařízením počínaje (myší, klávesnicí). To vše pak musí být citlivě začleněno do celého prostředí, ve kterém je interface užíván (resp samotné prostředí lze chápat jako interface) tak, aby nebránil komunikaci v něm, ale naopak sloužil, jako její prostředník (médiem). Je také důležité brát ohled na kulturní prostředí ve kterém je interface užíván. Významy znaků jsou totiž z velké části kulturně dané a proto je třeba s touto kulturní daností počítat.

Stejně jako předchozí autoři ukazuje Nadin, že je pomocí sémiotiky možné nejen tvořit nějaké závěry, ale i navrhovat správná řešení. Sémiotická analýza spočívá již v základním určení celku rozhraní jako znaku. Kde objekt je to, co je reprezentováno, tedy typ počítačového systému (textový procesor, tabulkový procesor atd.). Na straně representamenu pak leží samotné konkrétní programové ztvárnění objektu (např. MS Word, OpenOffice Calc). A vše uzavírá interpretace v podobě podmínek samotného užívání a hodnocení systému v kontextu. Je patrné, že zatímco tradiční pohled na rozhraní se zabývá pouze částí representamenu a objektu, sémiotická umožňuje komplexnější přístup.

Na úrovni jednotlivých prvků rozhraní lze nalézt prvky ikonické, indexikální a

61 Článek s. 284.

62 Tamtéž

63 Nadin 1988 s. 272.

symbolické přesně podle výše zmíněných definicí.⁶⁴ Je třeba dbát na to, aby tyto jednotlivé prvky byly ve vzájemné harmonii a nedocházelo mezi nimi k přílišnému prolínání. Je například chyba, pokud se v rozhraní objevují prvky symbolické smíšeně s ikonickými, jelikož každý z nich vyžaduje jiný způsob myšlení a daný výsledek může být pro uživatele matoucí.

Toto samozřejmě nejsou zdaleka jediní autoři, kteří se v současné době i v minulosti zabývají nebo zabývali sémiotickým přístupem k HCI. Mezi dalšími je možné jmenovat například ještě Davida R. Benyona a jeho pojetí počítačů jako informačních prostorů a HCI obecně jako problematiky navigace v informačních prostorech.⁶⁵ Významným činitelem v sémiotickém přístupu k HCI je v posledních letech tzv. SERG (tedy Semiotic Engineering Research Group). Tato skupina založená na teorii sémiotického inženýrství, které je předmětem této práce, přináší neustále nové a nové aplikace sémiotického přístupu k HCI, zejména co se týče aplikací jejich výzkumných metod, které budou představeny níže v této práci⁶⁶. Patří sem následující autoři: C.S. de Souza, S.D.J. Barbosa, R.O. Prates, E.J. da Silva a další.

Výše uvedené autory jsem ne zvolil jako zástupce sémiotického pohledu na HCI náhodně, ale proto že mi připadá, že v jejich pracích se začíná rýsovat specifický pohled na komunikaci v HCI, který je následně v Sémiotickém inženýrství převzat a dále použit jako základ teorie, která je dále rozpracována. Andersen pokládá základ sémiotickému chápání uživatelských rozhraní, přichází s myšlenkou, že sémiotika je díky své povaze a díky fenoménům se kterými se potýká, přirozeně vhodným nástrojem pro studium rozhraní. Jeho termíny „nutkaví mluvčí“ (compulsive speakers) a „nutkaví interpretátoři“ (compulsive interpreters) naznačují, že člověk nemůže jednat jinak, než nějak interpretovat zdělení a naopak nějaká vytvářet. Je v zásadě puzen ke komunikaci, která tak i v prostoru designu uživatelských rozhraní nutně probíhá. Zdánlivě se může zdát, že při této komunikaci jsou přítomni dva partneři a to člověk a počítač. Jak ale ukazují Nake a Grabowsky, mezi těmito dvěma nelze mluvit o standardní komunikaci, ale pouze o jakési pseudo-komunikaci. Opravdová komunikace pak probíhá jinde. Celé rozhraní slouží pouze jako prostředník, tedy podle Nadina „jako médium komunikace mezi lidmi. A jak Nadin

64 Nadin 1988 s. 272 a dále

65 BENYON, David, *Navigating information spaces*

66 Např. De Souza a Cypher 2008; De Souza 2010; De Souza, Prates, Barbosa 2000; Prates, De Souza 2000

ukazuje nejen mezi lidmi v rámci konkrétního kontextu použití, ale i mezi designerem a uživatelem. Poslední zmíněné je právě základem sémiotického inženýrství. Podle něj je rozhraní tzv. „designerův zástupce“ (designer's deputy⁶⁷), který za designera promlouvá. K tomu ale více v následující kapitole.

2.4 Sémiotické inženýrství v HCI

2.4.1 Historie

Ačkoli jsem se výše zmínil, že je Sémiotické inženýrství mladý obor, není to ale úplně pravda. Jeho počátky sahají do konce osmdesátých let minulého století a jsou spjaty zejména s prací Clarisse Sieckenius De Souza, která je profesorkou university v Rio de Janeiro. De Souza se k problematice HCI dostala přes studium lingvistiky a posléze NLP⁶⁸ a umělé inteligence. S ní v průběhu let spolupracují další výzkumníci jako Simone D. J. Barbosa, Raquel Oliveira Prates, Carla Faria Leitao a další.

První ucelená publikace ovšem vychází až v roce 2005 pod názvem *The Semiotic Engineering Of Human Computer Interaction*⁶⁹. De Souza v ní shrnuje celou teorii včetně jednotlivých sémiotických teoretických předpokladů. Ze jména z této práce a z několika dalších článků z devadesátých let budu vycházet v následující části. Další publikace vyšla v roce 2009 pod názvem *Semiotic Engineering Methods For Scientific Research In HCI*⁷⁰. Podle slov autorek je určena spíše pro výzkumníky a studenty oboru HCI a zabývá se do větších podrobností praktickým využitím teorie pro výzkum.

Nutno podotknout že v českém prostředí není tento přístup příliš rozšířen a takřka, kromě několika vlaštovek⁷¹, neexistují žádné komplexnější texty, které by se teorii věnovali.

2.4.2 Poslední exkurz

De Souza v úvodu své knihy zmiňuje několik koncepcí z oblasti teorie pragmatiky, které je dobré mít v paměti při uvažování rozhraní jako prostředku komunikace. Jelikož,

67 Odkaz na definici v SI 2005

68 Natural language processing

69 De Souza 2005

70 De Souza a Leitao 2009

71 Viz. například Vtípil 2009

jak uvidíme níže, člověk má při práci se strojem tendenci představovat si, že komunikuje s někým a ne s něčím, proto je třeba při designu rozhraní dodržovat určité zásady komunikace tak, aby byl její průběh co nejhladší.

2.4.2.1 Řečové akty

S termínem řečové akty přicházejí jako první Austin⁷² a Searle.⁷³ Ve stručnosti je základem této teorie myšlenka, že naše promluvy nejsou jenom nějakými informačními větami, které jsou vysloveny a zaniknou, ale že mají reálnou sílu něco ve světě měnit. Searle akty zařazuje do pěti různých kategorií.

- 1) Asertiva – K této kategorii je možné zařadit např. jednotlivá informační zdělení systému.
- 2) Direktiva – Jejich účelem je přinutit příjemce sdělení k nějakému činu
- 3) Deklarativa – Promluvy, které nějak mění stav okolního světa. Typickým příkladem je: „Odsuzuji vás k odnětí svobody.“
- 4) Komisiva – Jsou obecně sliby, které mluvčího k něčemu zavazují.
- 5) Expresiva – Chtějí přilákat pozornost posluchače

Searlova teorie řečových aktů stojí za prvními uživatelskými rozhraními používajícími přirozený jazyk jako prostředek komunikace. Je nutné spolu s De Souza⁷⁴ podotknout, že přísné lpění na teorii řečových aktů nepřináší příliš příjemný uživatelský prožitek. Je třeba přidat ještě něco dalšího.

2.4.2.2 Komunikační maximy

Maximy komunikace jsou jakési zásady, které by měla komunikace dodržovat proto, aby probíhala bez překážek. Je samozřejmostí, že v běžné konverzaci face-to-face jsou tyto maximy neustále porušovány, ale to je něco, co by se nemělo stávat v případě, že proti sobě nemáme jiného člověka, ale rozhraní počítače. Při užívání počítače nepředpokládáme, že si z nás počítač dělá legraci, nebo že přehání.

Paul Grice definuje sadu následujících konverzačních maxim:

72 Více např. AUSTIN, Langshaw J. *Jak udělat něco slovy*. Praha : Filosofia, 2000. 172 s.

73 Více např. SEARLE, John R. *Speech acts: an essay in the philosophy of language*. Cambridge : Cambridge University Press, 1969. 203 s.

74 De Souza 2005 s. 60.

- 1) Maxima kvantity – Nikdy neříkej více, nebo méně, než je třeba.
- 2) Maxima kvality – Říkej pouze to, čemu věříš, že je pravda.
- 3) Maxima relevance – Mluv pouze o tom, o čem se právě mluví.
- 4) Maxima způsobu – Mluv tak, aby bylo zřejmé, co chceš říci.

Greicovy maximy dále rozšiřuje například Leech a formuje tzv. maximy slušnosti⁷⁵, které dále rozvádějí řečové akty a greicovy maximy, aby odpovídaly zásadám slušné lidské komunikace.

- 1) Maxima taktu – Při udělování rozkazů nebo slibování minimalizuj dopad na jednotlivé mluvčí.
- 2) Maxima šlechetnosti - Při udělování rozkazů nebo slibování je slušné minimalizovat zisk mluvčího.
- 3) Maxima uznání – Při vyslovování něčeho, co ovlivňuje mluvčí, je třeba minimalizovat, nebo zcela zamezit snižování věcí, jež s nimi souvisí.
- 4) Maxima skromnosti – Je slušné minimalizovat chválení věcí, které souvisí s námi.
- 5) Maxima souhlasu – Je slušné minimalizovat nesouhlas v promluvě.
- 6) Maxima pochopení – Je slušné minimalizovat antipatie k jednotlivým mluvčím.

Na výše zmíněných Leechových maximách je zřejmé, že jsou značně poplatné britskému kulturnímu prostředí, ze kterého vznikly a v jiných prostředích by pravděpodobně vypadali jinak. Je to jenom důkazem toho, co bylo několikrát zmíněno výše a to, že každé rozhraní musí co nejvíce odpovídat prostředí, do kterého je zasazeno.

Sémiotické inženýrství je založeno zejména na analýze komunikace a to komunikace specifického typu. Je proto dobré mít v paměti, že zde existují určitá pravidla popisující, jak vést komunikaci, aby byla co nejúčinnější.

2.4.3 Vymezení Sémiotického inženýrství

Než přejdu k podrobnějšímu popisu teorie SI⁷⁶ zastavím se na chvíli u toho, jak se

⁷⁵ Celé jméno a odkaz někam dál

⁷⁶ Nádále budu Sémiotické inženýrství označovat touto zkratkou

vymezuje vůči ostatním teoriím. Jmenovitě vůči již zmíněnému UCD. Tím se také dostanou do popředí některé základní koncepty, které budou rozebrány níže.

UCD se snaží na základě systematického sběru informací o uživateli dosáhnout co nejlepší představy o tom, co uživatel potřebuje, jaké jsou jeho cíle (viz. výše) a očekávání. Postupuje se pomocí propracovaných metod sahajících od empirických výzkumů, přes modelování uživatelů, až po rozsáhlé testování. Výsledkem je jakýsi obraz programu, který je ideální odpovědí na všechny uživatelovi požadavky. Jelikož je ale tento systém velice časově náročný a nákladný, vznikají na základě výzkumů určité návody a pravidla, kterými by se měli designeři řídit tak, aby jejich produkty odpovídaly požadavkům UCD, tzn., požadavkům jejich potenciálních uživatelů.

Jeden ze zakladatelů UCD D. Norman chápe proces užívání interaktivního programu, jako probíhající v následujících sedmi fázích.

- 1) Stanovení obecného cíle interakce.
- 2) Tvorba bezprostředních důvodů pro užití rozhraní.
- 3) Formování plánu akcí tak, aby bylo dosaženo záměru.
- 4) Vykonání akcí
- 5) Pozorování odpovědí systému na vykonané akce
- 6) Interpretace pozorovaných výsledků.
- 7) Zhodnocení výsledků s přihlédnutím k obecnému cíli a záměrům, které vedli k použití.⁷⁷

Autoři SI kritizují výše uvedený přístup z několika úhlů pohledu.

Z teorie SI vyplývá, že není možné přesně naplánovat, jak bude probíhat uvažování uživatele a jaké budou jeho akce. Jako příklad uvádějí autoři uživatele MS Wordu, kteří ho často nepoužívají jenom jako textový editor, ale i jako primitivní způsob uchování informací.⁷⁸ Je tedy třeba se zaměřovat ne pouze na přesné plánování interakcí, ale počítat také s neustálou změnou přístupu uživatele k rozhraní. Výstižně to autoři popisují v úvodu svého článku z roku 1993: „Zatímco Kognitivní věda ukazuje designerovi cíl, na který má mířit, sémiotika mu poskytuje cenné nápovědy k tomu, aby úspěšně střílel.“⁷⁹

77 Norman 1986

78 De Souza 1993 s. 757.

79 De Souza 1993 s. 754.

UCD se zaměřuje pouze na uživatele (jak je zřejmé z názvu) a opomíjí roli designera, nejen v procesu tvorby rozhraní, ale také v čase samotné interakce. UCD připraví k dokonalosti vykreslený obraz toho, jak má rozhraní vypadat a předloží ho uživateli, aby se s ním nějak potýkal. Oproti tomu sémiotické inženýrství ve svém základu počítá se zapojením designera i ve fázi užívání. Ten je tak skrze svého zástupce (rozhraní) neustále připraven s uživatelem komunikovat. To znamená, podávat mu sdělení o významu a jednotlivých funkcích programu. Z pohledu sémiotického inženýrství v sobě každé uživatelské rozhraní obsahuje následující základní sdělení poslané skrze rozhraní designerem k uživateli:

„Here is my understanding of who you are, what I've learned you want or need to do in which preferred ways and why. This is the system that I have therefore designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision.“⁸⁰

V pozdějších textech⁸¹ nazývá De Souza tyto věty tzv. „metakomunikační šablonou“ (metacomunication template), kterou by se měla řídit veškerá metakomunikace v rozhraní (Její význam se ukáže zejména v metodologické části práce).

Designer se z pohledu SI nesnaží pouze porozumět uživateli, ale snaží se mu své porozumění i nějak komunikovat. Schéma vývoje designu SI by se dalo vydělit do následujících fází (srv. s cooperovým procesem vzniku UCD designu, nebo nomanovými fázemi užívání).⁸²

- 1) Designer studuje aktivity uživatelů v jejich prostředí.
- 2) Skrze počítačovou technologii ukazuje uživatelům, jak by se jejich činnosti měli změnit, aby dosáhli svých cílů.
- 3) Uživatelé přijímají a postupně v průběhu užívání rozhraní odkrývají designerovo sdělení.
- 4) Uživatelé ve finále plně chápou designerovo sdělení a adekvátně na něj reagují.

Zaměření na uživatele a zároveň designera v rámci jednoho teoretického přístupu je něco, co UCD chybí. Bere se v něm úvahu pouze uživatel a to natolik přísným způsobem,

80 De Souza 2005 s. 84.

81 Např. De Souza 2009

82 De Souza 2005 s. 24.

že paradoxně hrozí, že výsledný produkt bude pro uživatele nesrozumitelným. Sémiotické inženýrství klade důraz na neustálou komunikaci ve všech jejích možných podobách tak, aby bylo rozhraní uživateli vždy srozumitelné a nedocházelo k selháním komunikace.

Ve své práci z roku 2005 jmenuje De Souza i další problémy, kterým UCD může čelit. Na základě množství výzkumů byly vytvořeny určité postupy a manuály pro designery, aby věděli, jak mají postupovat. Technologický pokrok je ale natolik rychlý, že tyto postupy rychle zastarávají a není možné provádět stále dokola nákladné studie a empirické výzkumy. Tyto výzkumy jsou také natolik nákladné, že zadavatelé nejsou ochotni za ně platit a musí docházet ke kompromisům, které se samozřejmě odráží na kvalitě. Dochází také k přílišnému lpění na vytvořených nástrojích a ty jsou prostě neustále replikovány namísto toho, aby byly chápány pouze jako nástroje pomáhající činit určitá rozhodnutí v určitých situacích tak, jak bylo původně zamýšleno. Přílišné trvání na zažitých pravidlech může také zatlačovat do pozadí některé nové myšlenky.

Sémiotické inženýrství oproti tomu nenabízí nějakou sadu fixních nástrojů ale vytváří způsob pohledu na rozhraní, který je diametrálně odlišný. Hlavní důraz je zde kladen na komunikaci a na užívání jednotlivých stavebních prvků rozhraní tak, aby bylo komunikace úspěšně dosaženo. Klade důraz na uživatele a na designera v rámci jednoho celku. Na základě své teorie pak vytváří metody sloužící k praktickému hodnocení procesu designu⁸³.

Toto je z pohledu cíle této práce zajímavá námitka. Jak se zdá, odmítá De Souza tvorbu nějakých pravidel a vodítek. Jsou ale dva hlavní důvody, proč si myslím, že je i přes tento nesouhlas možné a užitečné nějaká takováto pravidla získat z teorie sémiotického inženýrství.

- 1) Jak už jsem zmínil na začátku. Výstupy sémiotického inženýrství mají jinou povahu než výstupy UCD a proto i pravidla z něho vzniklá budou mít nutně jinou povahu. Budou daleko méně konkrétní a proto nebudou tolik podléhat času. Navíc zde nehrozí to, že by přílišné lpění na nich nějak brzdilo vývoj a bránilo novým myšlenkám, jelikož jejich potenciálem je naopak nutit designera vymýšlet nové způsoby, jak co nejlépe komunikovat s uživatelem. Daná pravidla pouze poskytují podnět k tomuto zamýšlení se.
- 2) Námitky proti tvorbě pravidel z teorie jsou v akademickém prostředí

83 O nich viz následující kapitoly

pochopitelné, jelikož pro člověka, který se oborem dlouhodobě teoreticky zabývá je obtížné přijmout, že teorie bude nějak okleštěna, aby se jí dostalo jednoduchého praktického uplatnění i přesto, že ona sama o sobě je značně komplexní a není možné v ní něco ubírat a zkracovat. V prostředí reálné praxe jsou ale právě tato pravidla a vodítka tím, co je potřeba, jelikož málo který programátor má čas a vůli se zabývat podrobným studiem teorie a málo která společnost má prostředky na to, aby mohla najímat někoho, kdo se teorií a výzkumem bude zabývat. Je proto užitečné mít k ruce sadu vodítek, kterých je možné se držet i za cenu určité redukce.

2.4.4 Základní pojmy SI

Jak je patrné z výše uvedeného, hraje hlavní roli v teorii SI vztah designer-uživatel realizující-se skrze rozhraní. Tyto tři prvky se spojují v jednotném teoretickém modelu, což není v rámci teorie HCI příliš obvyklé. Většinou je brán v úvahu pouze uživatel případně uživatel a rozhraní, ale ne designer.

Tento nový pohled přináší i nové možnosti pro design. Designeři musí v průběhu tvorby přemýšlet o své úloze v rozhraní a o tom, jak budou v průběhu užívání programu komunikovat s uživateli. Jinými slovy musí promýšlet své sdělení a to nejen ve smyslu obsahu, který budou sdělovat, ale i jak ho budou sdělovat. Musí užívat jednotlivé znaky rozhraní tak, aby tvořily srozumitelný jazyk a tímto jazykem musí předávat co nejsrozumitelnější sdělení (na pomoc si mohou vzít například výše zmíněné maximy). Designer zde zaujímá roli člověka vytvářejícího nový jazyk (znakový systém) a tento jazyk musí uživatele naučit skrze rozhraní, které je zároveň pomocí daného jazyka vybudováno. Nemá možnost v průběhu užívání rozhraní s uživatelem komunikovat a proto musí veškerá svá možná sdělení (tj. veškeré možné konverzace o rozhraní odehrávající se mezi ním a uživatelem) zakomponovat již ve fázi tvorby.

Tento pohled neslouží pouze jako vodítko pro designery v procesu tvorby, ale může být užit i při analýze již hotového rozhraní. Podle De Souzy je možné zaměřit se například na následující faktory:⁸⁴

- 1) Jaké znaky jsou užity při konstrukci znakového systému rozhraní a jak jsou úspěšné při předávání sdělení od designera k uživateli?

84 De Souza 2005 s. 31.

- 2) Jak je zakódován účel rozhraní?
- 3) Jak je uživatelům umožněno tento účel použít ve svůj prospěch?
- 4) Jaké sémiotické procesy jsou v uživateli rozhraním spouštěny, které z nich jsou v rámci rozhraní očekávány a je na ně reagováno a které ne?
- 5) Pokud se sémiotické procesy uživatele vyvinou jinak, než bylo designerem očekáváno, jak je toto selhání ošetřeno?

2.4.4.1 Metakomunikace

Základním pojmem SI je metakomunikace. Tedy komunikace o probíhající komunikaci. Designér je v rozhraní jakoby neustále přítomen a sděluje uživateli důležité informace o rozhraní. Nejviditelnějším příkladem jsou např. různé formy nápovědy, kdy je uživateli explicitně sdělováno, co který prvek rozhraní znamená a jaká je jeho funkce. Metakomunikační prvky se ale nacházejí zakomponované i v rozhraní samotném pomocí jednotlivých statických a dynamických znaků (viz níže v kapitole o Sémiotické inspekční metodě). Autoři SI zdůrazňují, že tato komunikace o komunikaci se odehrává nezávisle na tom, jestli je designerem a uživatelem reflektována.⁸⁵ Není tedy něčím, co je do rozhraní třeba nově zakomponovat na základě teorie SI, ale naopak něčím, co je vždy přítomno. SI ji pouze zviditelňuje a tak umožňuje designérům, aby se na ni aktivně soustředily.

Designer ovšem, jak už bylo výše zmíněno, nemůže být nikdy přítomen fyzicky v čase interakce, proto za sebe vysílá rozhraní jako svého zástupce (viz níže), který hovoří za něj.

Myšlenka metakomunikace skrze rozhraní není, jak autoři poukazují, nic nového.⁸⁶ Rheinfrank a Evenson ve své práci z roku 1996 například tvrdí, že zatímco přirozené jazyky jsou určeny k tomu, aby sdělovali myšlenky, jazyky designu (tedy jazyky umělé) jsou určeny k tomu, aby vytvářeli objekty vyjadřující, co sami jsou a k čemu slouží.⁸⁷ Tomuto pojetí odpovídá i termín *affordance*⁸⁸, který popisuje to, jak objekty sami za sebe vyjadřují, k čemu slouží. Například tlačítka v rámci programu jsou vymodelována jakoby plasticky i přesto, že na ploché počítačové obrazovce jejich plasticita nemá žádný význam. Designer nám tak dává najevo, že je jeho záměrem, abychom tlačítka používali tak, že na

⁸⁵ De Souza 2009 s. 17.

⁸⁶ De Souza 2009 s. 18.

⁸⁷ Více viz Rheinfrank a Evenson 1996

⁸⁸ Viz např. Norman 1993

něj na něj najedeme myší a stiskneme. Pokud je pak na tlačítku ještě například ikona odpadkového koše, říká nám nejen to, že se jedná o tlačítko, ale i to, jakou akci tlačítko vyvolá.

De Souza a Letaio definují předmět výzkumu Sémiotického inženýrství jako: „soubor všech, možných počítačově kódovaných konverzací, které může designerův zástupce vést s uživatelem v době interakce a žádných jiných.“⁸⁹ To, co je podstatné pro teorii SI, je metakomunikace probíhající v čase interakce, ta je předmětem zkoumání. Ne už ale konverzace designer – uživatel probíhající ve fázi designu. Designer by neměl vézt v průběhu vytváření návrhu konverzací s uživatelem (ať už reálnou, nebo teoretickou) a na základě ní tvořit design (jak to dělá např UCD), ale měl by brát v potaz všechny možné konverzace, které se mohou vyskytnout v čase užívání.

2.4.4.2 Designerův zástupce

Designerův zástupce (Designer's deputy) je ústředním pojmem teorie sémiotického inženýrství. Jak už bylo řečeno, uživatelské rozhraní tvoří prostředníka mezi uživatelem a designerem. Designer vysílá rozhraní k uživateli jako svého zástupce a posílá skrze něj sdělení o tom, jak má být s jeho zástupcem nakládáno a k čemu slouží.

Teorie rozhraní jakožto designerova zástupce je založena na pojetí softwaru jako intelektuálního artefaktu. Artefakt je nějaký lidský výtvar, který může nabývat konkrétní podoby (kladivo, stůl), ale také abstraktní (například různá preventivní opatření). Specifickým typem je pak artefakt intelektuální, který v sobě zahrnuje následující rysy⁹⁰:

- 1) Má v sobě zakódováno určité porozumění nějaké situace, kterou je potřeba řešit.
- 2) Dále zahrnuje sadu nástrojů pro řešení dané situace.
- 3) Porozumění situaci a nástroje jejího řešení jsou v rámci artefaktu zakódovány v nějaké jazykové podobě. S tím, že termín jazyk je zde třeba chápat v rozšířeném smyslu jako jakýkoli systém symbolů ať už verbálních, nebo obrazových, na který je možné aplikovat nějaká sémantická pravidla.
- 4) Účel tohoto artefaktu je naplněn až v okamžiku, kdy jsou uživatelé schopni, za pomoci předloženého jazykového systému, tento účel artikulovat. Jinými slovy – musí rozumět jazyku rozhraní a musí rozumět tomu, co je jim o tomto rozhraní

89 De Souza 2009 s. 19.

90 De Souza 2005 S.10.

oním jazykem sdělováno a naopak musí být schopni tímto jazykem sdělovat rozhraní své požadavky.

Designerova zástupce je tedy možné chápat jako intelektuální artefakt. Odráží se v něm designerovo pochopení a interpretace určité sady problémů, které potřebuje uživatel vyřešit. Např. v obyčejné kalkulačce se může odrážet pochopení toho, že uživatel potřebuje počítat úlohy, na které mu jeho schopnosti nestačí. V grafickém editoru pak designerovo specifické pochopení toho, co uživatel potřebuje při editaci fotografií (Fakt, že je toto pochopení u každého designera různé dobře ilustruje škála programů, které jsou na trhu k dispozici).

Spolu s tímto pochopením nám designér programu zároveň předkládá řešení našich problémů a to takovým způsobem, abychom jasně pochopily, jaký je jeho záměr (co nám chce sdělit). Designerův zástupce za něj hovoří⁹¹ v jeho nepřítomnosti, je proto nutné, aby hovořil jasně, bez možnosti vzniknutí omylu a aby hovořil zcela k věci. Tento jednorázový všezahrnující vzkaz vyslaný skrze zástupce směrem k uživateli by měl obsahovat jenom informace, které jsou relevantní, neměl by hovořit o ničem jiném (viz výše zmíněné maximy). Proto se musí designer v průběhu své práce neustále ohlížet zpátky a klást si otázku: „Jak dobře bude tento zástupce sdělovat uživatelům, co mám na mysli (včetně toho, co po by podle mě měli dělat)“.⁹²

Sémiotické inženýrství je tedy podle výše uvedených termínů teoretickou disciplínou, která se primárně zabývá metakomunikací odehrávající se v rámci uživatelského rozhraní. Tato metakomunikace probíhá skrze jednorázové sdělení vyslané designerem k uživateli pomocí jeho zástupce. Je třeba si uvědomit, že předmětem SI je pouze toto a je schopné řešit problémy a navrhovat řešení, které souvisí s touto metakomunikací. V žádném případě tedy nenahrazuje dosavadní teorie designu uživatelských rozhraní. Přináší do nich ale nový vhled a významně je doplňuje, tam kde selhávají. V následující části ukážu teorii SI v HCI o něco podrobněji.

2.4.5 Ontologie sémiotického inženýrství

Ontologie SI se skládá ze čtyř částí. Procesů signifikace, procesů komunikace,

91 De Souza 2005 s. 90.

92 Tamtéž s. 91.

jednotlivých mluvčích zahrnutých v procesu signifikace a prostoru, ve kterém se se odehrává design vycházející s teorie SI.

2.4.5.1 Procesy signifikace

Znaky, pomocí kterých probíhá metakomunikace přenášející designerův jednorázový vzkaz skrze jeho zástupce, nemusí být nutně textové a mohou nabývat jakýchkoli audiovizuálních podob uspořádaných do nějakého designerem vytvořeného systému. Je zřejmé, že Peircova širší koncepce odpovídá teorii SI lépe, ale i ona má určité limity. De Souza také představuje svoji vlastní klasifikaci znaků, tak aby odpovídala potřebám Sémiotického inženýrství a oblasti HCI. Dělí je na statické znaky, dynamické znaky a metalingvistické znaky.⁹³

Statické znaky jsou ty znaky v rozhraní, které zůstávají neměnné v průběhu interakce. Jedná se například o samotné rozložení oken, polohu jednotlivých menu a jejich položek, ikony a jejich vzhled. Jsou to znaky, které interpretujeme bez ohledu na jejich kauzální vztahy k jiným znakům. Dynamické znaky naopak mají vztah ke konkrétní fázi interakce, mění se v průběhu naší činnosti. Obecně je možné říci, že slouží jako průvodci interakce. Jsou to jednotlivé akce a změny stavů odehrávající se se statickými znaky. De Souza říká, že tyto dva druhy znaků jsou vzájemně propojeny: „Statické znaky stimulují uživatele k práci s artefaktem [...] Dynamické znaky stimulují jejich očekávání.“⁹⁴ Posledním typem znaků jsou znaky metalingvistické, ty jsou v rozhraní užity k tomu, aby něco vypovídaly o ostatních znacích. Může to být například integrovaný systém nápovědy, nebo i různá okna s upozorněními a informacemi o tom, co se právě děje. De Souza upozorňuje na to, že designeři by měli být při tvorbě těchto znaků velice opatrní, měli by sloužit k podpoření sémiózy uživatele správným směrem a v žádném případě by ji neměli násilně přerušovat.

To podstatné vyplývající pro SI z teorie Peircovi je fakt, že za prvé to, že něco znamená něco pro někoho je ovlivněno přecházejícími znalostmi a zkušenostmi (viz také Eco) a za druhé, to, že proces tvorby a interpretace znaků probíhá neustále, není nikdy uzavřený. To je také podstatou Peircova pojmu abduktivního myšlení, který byl podrobněji vysvětlen výše. Shrnutím celé této charakteristiky znakového trojúhelníku je teorie sémiózy popisující neustále se vyvíjející povahu znaku. Důležitou rolí pro sémiotické

93 De Souza 2010 s. 19.

94 tamtéž

inženýrství hraje i Ecova teorie znakové produkce, jejímž základem je mimo jiné myšlenka neustálé produkce a kulturní podmíněnosti znaků a kódů.

Zde leží pro designery aplikací zásadní problém, jelikož proces sémiózy nelze v žádném případě předem odhadnout, navíc je u každého uživatele jiný. Situace v HCI je tak komplikovaná jelikož se tu střetává designerova sémióza o tom, jak nejlépe připravit půdu pro uživatele a jak ten bude pravděpodobně reagovat, se sémiózou uživatele, který vždy používá program v nějaké konkrétní situaci a vychází z nějakého předchozího kontextu, svých vlastních zkušeností a dipozic. Tato uživatelova sémióza je předem neodhadnutelná. Je proto úkolem designera snažit se ji co nejvíce předpovídat a případně směřovat správným směrem. Zároveň musí počítat se situacemi, kdy se jeho sémióza se sémiózou uživatele zcela rozejde a být na ně v rámci rozhraní připraven, aby nedocházelo ke komunikačním zkratům.

Pro úplnost celého obrazu je třeba si uvědomit, že tyto dvě sémiózy se spolu střetávají posunuty v čase prostřednictvím designerova zástupce, který žádnou takovou vlastnost nemá a jeho zpracování znaků je omezeno pouze na to, co do něj bylo předem vloženo. Zde je možné odkázat na výše zmíněnou teorii Nakea a Grabowski, která tento rys rozvádí do větších podrobností. Výsledek designerovy sémiózy je tak poslán uživateli skrze rozhraní, které je v tomto směru značně omezené. Designer se proto musí snažit vytvořit ho tak, aby bylo schopno reagovat na co největší možné rozpětí sémiózy uživatele.

2.4.5.2 Procesy komunikace a jednotliví mluvčí

Procesy komunikace se v rámci uživatelského rozhraní odehrávají mezi třemi partnery: designerem, uživatelem a designerovým zástupcem. Rozhraní z pohledu teorie Sémiotického inženýrství přivádí všechny tři do komunikace v místě a čase interakce. Tím se SI daří překlenovat rozdělení, které je implicitně zahrnuto ve všech ostatních teoriích HCI. Je to rozdělení uživatel vs. designér a uživatel vs. systém. Sémiotické inženýrství dělá z těchto dvou jeden komplexní problém designer, uživatel, systém.⁹⁵ Zatímco většina teorií má tak tendenci řešit spíše otázku komunikace mezi uživatelem a systémem, SI postupuje jinak. Říká, že postavu designera nelze z interakce vyloučit. Rozhraní vlastně slouží pouze jako médium pro přenos výsledku designerovy sémiózy o potřebách a problémech

95 De Souza 2009 s.20.

uživatele směrem k uživateli.⁹⁶ Asi každý zažil situaci, kdy mu mimoděk unikla při práci s počítače například věta typu: „No to si ze mě děláš legraci“. Nemluvíme takto se systémem, ale reagujeme na designera a jeho sdělení skryté v rozhraní. Nemáme ho sice k dispozici fyzicky na místě, ale máme přítomného jeho zástupce, který je oprávněn za něj jednat.

Mezi těmito partnery tedy probíhají obecně dva druhy komunikace odehrávající se na různých úrovních. Je to metakomunikace mezi designerem a uživatelem a komunikace mezi uživatelem a systémem. Metakomunikace probíhá pouze ve směru designér – uživatel a to na základě jednorázového sdělení, které je v systému zakomponováno a je uživatelem postupně odkrýváno v průběhu jeho interakce a komunikace se systémem. Samotná komunikace se systémem je pak ovlivňována a usměrňována na základě této metakomunikace. „Komunikace mezi uživatelem a systémem je nízko-úrovňovým komunikačním procesem, který je určený vyšší úrovni metakomunikace.“⁹⁷

Dalším důležitým prvkem, který je ještě třeba zmínit, je již výše uvedená povaha procesu komunikace, jak ji uvádí Umberto Eco ve své teorii znakové produkce⁹⁸. Komunikace je procesem, ve kterém jsou užívány stávající znakové systémy a kódy. To se ale nikdy neděje podle předem daných striktních pravidel. Naopak komunikace je živý proces, při němž dochází k neustálému porušování a kreativní modifikaci daných kódů. Je proto zřejmé, že nikdy není možné, aby designer přesně předpokládal všechnu možnou komunikaci odehrávající se mezi uživatelem a systémem a už vůbec není možné, aby automaticky předpokládal, že daná komunikace se bude odehrávat přesně podle pravidel stanovených v rámci kódu, který vytvořil a předal skrze rozhraní směrem k uživateli. Je ale třeba, aby s něčím takovým počítal a byl schopen na případné odchylky reagovat. Tedy, aby jeho jednorázové metakomunikační sdělení bylo relevantní i v případě odchylek od předpokládané cesty a nebylo nahrazeno jenom trapným mlčením.

2.4.5.3 Prostor SI

Pojem prostor je zde chápán ve smyslu prostoru, ve kterém by se měl designer pohybovat tak, aby jeho rozhraní fungovalo, jak fungovat má, ve smyslu výše popsaného přístupu. K jeho vytýčení z pohledu probíhající komunikace si bere de Souza na pomoc

96 De Souza 2005 s.96.

97 Tamtéž s.97.

98 Eco 2004

Jakobsonův model komunikace.⁹⁹ Z pohledu prostoru designéra, ve kterém se může pohybovat při tvorbě nových znaků a kódů je pak využita ecova Teorie znakové produkce.

Jakobson používá při popisu struktury komunikace termíny vysílatel, příjemce, zdělení, kód, kanál, kontext.¹⁰⁰ Jeho model je možné ve stručnosti popsat následujícím způsobem. Vysílatel vysílá svůj vzkaz směrem k příjemci pomocí nějakého kanálu a používá při tom pro přenos nějaký kód. To vše se odehrává v konkrétním kontextu, který celé komunikační situaci dodává smysl. Na základě tohoto rozdělení pak určuje 6 různých funkcí jazyka, které nalézají v komunikaci uplatnění. Můžeme je sledovat i v prostředí uživatelského rozhraní.

- 1) Expresivní funkce – Soustředí se na vysílatele vzkazu. Ten pomocí ní vyjadřuje informace o sobě, o svém postoji nebo stavu. Sem spadají informace systému typu: „Nemohu zpracovat vaši žádost“ a další.
- 2) Konativní funkce – Pomocí ní ověřuje vysílatel zda-li je příjemce na příjmu. Vztahuje se tedy spíše k příjemci.
- 3) Referenční funkce – se užívá, když mluvíme o něčem. Odkazuje ke kontextu. Např. „Na pevném disku nezbyvá volné místo.“
- 4) Fatická funkce – se soustředí na kanál. Vysílatel pomocí ní většinou ověřuje, jestli je příjemce dostupný skrze daný kanál (například různé testy síťové komunikace)
- 5) Metalingvistická funkce – se vztahuje ke kódu. Například když nám designer skrze svého zástupce ukazuje, jak máme jeho kódů rozumět.
- 6) Poetická funkce – se vztahuje ke sdělení samotnému. Můžou to být například různé vizualizace doprovázející výstup počítačového programu.

Jakobsonův model pomáhá designerovi myslet na jednotlivé prvky komunikace tak, aby tvořily soudržný celek a nedocházelo k selháním v přenosu. Musí si vždy uvědomovat: Jakou část svých postojů, zkušeností, názorů atd. Chce přenést. Jaká část toho, jak chápe postoje, názory (etc.) uživatele by měla být sdělena skutečnému uživateli. Jakou část uživatelského kontextu (kulturního, profesionálního) je třeba brát v rámci rozhraní v potaz. Jaké výpočetní kódy budou použity, proto aby byla komunikace co nejefektivnější. Jaký je k dispozici kanál pro přenos sdělení. A v neposlední řadě, jaké bude sdělení samo o sobě.

99 De Souza 2005 s. 66.

100 Jakobson 1960

Tedy co chce designer skrze rozhraní uživateli vlastně říci.¹⁰¹

V případě trojjediné komunikace mezi designerem uživatelem a systémem se jedná o specifickou situaci. Předně designér a uživatel se zde nenalézají ve stejném kontextu. Designer spíše předpokládá uživatelský kontext (to je komplikováno jak jeho, tak uživatelským procesem sémiózy) a pro něj daný program navrhuje. Kanálem, skrze který je sdělení posíláno, je počítač, sdělení musí být zakódováno tak, aby mohlo být skrze něj posláno. Designerovo chápání uživatelského kontextu musí být zapsáno do počítačového kódu a je tak na výstupu omezeno, jelikož uživatel nemá možnost komunikovat přímo s designerem, ale s jeho prostředníkem, který sdělení značně oklešťuje. De Souza tak ukazuje, že počítač není pouze mediem komunikace a metakomunikace, ale dá se říci, že do značné míry určuje i podobu daného kódu, možných kanálů výsledného sdělení a dokonce i kontextu komunikace. Říká že: „...tuto formulaci je možné rozšiřovat mnoha směry, počítače mohou být kódem, kanálem, sdělením i vysílatelem ...“¹⁰² McLuhanovo prohlášení o tom, že medium je poselstvím je tím značně rozšířeno.

Jakobsonova teorie pomáhá vytyčit prostor pro úspěšnou komunikaci. Ecova teorie je pak užitečná při vytyčení prostoru pro efektivní tvorbu znaků a kódů, které budou v průběhu komunikace předávány. Jeho typologie produkce znaků již byla výše zmíněna v sémiotickém úvodu. Je vypracována na základě 4 kategorií.

- 1) Vykonaná práce nutná k vytvoření výrazu
- 2) Poměr typ-token
- 3) Typ přetvářeného kontinua
- 4) Modus a komplexita artikulace

Na jejich základě vypracovává De Souza čtyři pravidla pro správnou tvorbu znaků v HCI.¹⁰³

- 1) Designér by měl vytvářet znaky tak, aby co nejvíce odpovídaly systému ustálenému v kontextu, do kterého je využívání systému situováno. Měl by se snažit co nejméně vytvářet znaky zcela nové a co nejvíce sahat do zásoby, která je mu daným kontextem dána k dispozici.

101 De Souza 2005 s. 87.

102 De Souza 2005 s.89.

103 De Souza 1993 s.764.

- 2) S tím souvisí i druhé pravidlo. Měl by co nejvíce vybírat takové tokeny, jaké jsou standardně dle konvence používány pro označení daných typů (ratio facilis). Neměl by se snažit nalézat nové tokeny, které sice vzhledem odpovídají tomu, co označují, ale nejsou ve společnosti zakořeněny.
- 3) Znak rozhraní, které referují k objektům týkajícím se počítače, jeho řešením problémů a procesům by měli být homomateriální a znaky týkající se vstupu a výstupu by měli být heteromateriální.
- 4) Designeři by měli vždy používat znaky systémů, které jsou založené na systematických pravidlech. Všechny procesy, které se odehrávají na všech úrovních programu jsou založeny na nějakých pevně daných pravidlech, jelikož tak umožňují přesnou interpretaci daných znaků. Pokud bude designer užívat takovéto na pravidlech postavené systémy sníží tím pravděpodobnost nedorozumění.

2.4.6 Epistemologické následky Sémiotické inženýrství

Než přejdu k představení metod sémiotického inženýrství, zastavím se ještě na chvíli u pojmu význam (meaning) a změny jeho pojetí v rámci SI jako teorie HCI.

Sémiotické inženýrství přináší do HCI zásadní rozdělení na význam uživatelský a význam designerův. Tyto dva se k sobě sice můžou blížit, ale nikdy nebudou schodné a není ani možné, aby designer předpokládal, že jeho význam je shodný s významem uživatele. De Souza poukazuje na to, že dřívější teorie UCD se zaměřují převážně na uživatelský význam a snaží se ho všemi možnými způsoby předem odhadnout, případně změřit jeho blízkost k designerovu významu.¹⁰⁴ Sémiotické inženýrství ovšem ukazuje, že není možné uživatelský význam brát jako měřítko čehokoli, jelikož ho nelze nijak přesně definovat. Nelze s jistotou říci, jaký bude uživatel přiřazovat význam jednotlivým prvkům rozhraní a nelze ho tedy ani použít jako měřítko úspěšnosti tohoto rozhraní. Je vhodnější předem počítat s tím, že bude docházet k rozcházení se významů a tak i k situacím selhání. Rozhraní by na toto mělo být připraveno a nemělo by se spoléhat na dokonalý model uživatele.

Jediné, co lze v otázce významu zjišťovat je rozdíl mezi významy znaků v rozhraní a jejich kulturně zakořeněnými významy. Kultura nám totiž slouží jako zásobárna významů a to, jak v případě, že znaky tvoříme, tak i v případě, že je nějak interpretujeme. Proto by

104 De Souza 2005 s.99.

měl designer vytvářet rozhraní tak, aby významy v něm obsažené byly co nejbližší významům, které jsou kulturně dané.

Sémiotické inženýrství tak přináší do oboru radikální změnu v tom, co je možné s určitostí vědět a co ne. Je opět třeba mít na mysli, že prostředníkem mezi designerem a uživatele je počítač. Tedy zatímco význam uživatele nikdy neznáme, můžeme vždy s určitostí tvrdit, jaký bude význam, který znakům v rozhraní přiřazuje počítač. Designerův zástupce je jediným ze členů komunikace v HCI jehož způsob přiřazování významu je možné předem odhadnout a naprogramovat, je to to jediné co jsme schopni opravdu vědět. Proto snaha o nějaké exaktní uchopení významu měla směřovat zejména sem.

2.4.7 Metody sémiotického inženýrství

Nyní se dostáváme k podstatné části této práce. Jejím úkolem je představit metody teorie sémiotického inženýrství, které složí k analýze rozhraní z hlediska jeho schopnosti vést efektivní metakomunikaci s uživatelem. Tyto metody a zejména výzkumy, již v minulosti provedené na jejich základě, poslouží jako jeden z nástrojů pro derivaci obecných pravidel.

Ve své práci z roku 2005 stanovuje De Souza základní metodologická vymezení oboru sémiotického inženýrství.¹⁰⁵ a) Jeho teorie není prediktivní teorií, tj. není schopno poskytnout takovou metodu, která by předvíдалa jakékoli výsledky. To plyne i z výše zmíněné neurčité povahy procesů odehrávajících se v HCI. b) Sémiotické inženýrství by se mělo vyjadřovat o pozorovatelných fenoménech. c) Sémiotické inženýrství by mělo poskytovat takové metody, které budou schopné zformulovat problémy v HCI a navrhnout jejich řešení.

V této době jsou výzkumníkům v oboru k dispozici dvě metody sémiotického inženýrství SIM¹⁰⁶ a CEM¹⁰⁷. Jejich účelem je umožnit člověku, který provádí analýzu rozhraní, odhalit, jakým způsobem zakódoval designer svůj záměr a své pochopení uživatele do rozhraní a zdali je komunikace tohoto zájmu a porozumění vzhledem k uživateli srozumitelná a organizovaná. SIM zkoumá, jak jsou jednotlivé metakomunikační znaky v rozhraní zakomponovány a CEM pak na základě pozorování uživatelů zjišťuje, nakolik je tato metakomunikace úspěšná. Obě metody, zejména pokud jsou využity při

105 De Souza 2005 s. 104. - 105.

106 Semiotic Inspection Method (Metoda sémiotické inspekce) – nadále budu používat zkratku SIM

107 Communicability Evaluation Method (Metoda hodnocení komunikativnosti) – nadále budu používat zkratku CEM

výzkumu společně, jsou schopny poskytnout cenné výsledky. CEM například odhalí některé nedostatky v metakomunikaci a SIM je schopna na základě své analýzy zjistit, proč k nim dochází. Je tak možné navrhnout řešení tak, aby k selháním nedocházelo.

2.4.7.1 Metoda sémiotické inspekce (SIM)

Jejím cílem je rekonstruovat metakomunikaci tak, jak probíhá v rozhraní. Analýza se provádí postupně na jednotlivých úrovních znaků. Na jejím konci je možné popsat „komunikativní potenciál jednotlivých interakcí“¹⁰⁸ a to včetně možné identifikace míst, ve kterých bude tato komunikace selhávat, například díky inkonzistenci jednotlivých sdělení. Nyní si ve stručnosti projdeme, jak probíhá analýza pomocí této metody.

Základním vodítkem je tzv. metkomunikační šablona tak, jak byla výše citována. Na jejím základě je třeba pokusit se odpovědět na otázky, týkající se rozmístění metakomunikace v rozhraní. Tyto otázky jsou:

- 1) Co je obsahem designerova sdělení?
- 2) Komu je designerovo sdělení adresováno?
- 3) Jak tuto svoji komunikaci přenáší designér do znaků rozhraní?
- 4) Jaká je designerem předpokládaná komunikace uživatele se systémem?
- 5) Jak, kde, kdy a proč se podle designerova očekávání pustí uživatel do komunikace se systémem.¹⁰⁹

V přípravné fázi je nejprve nutné zvolit si vhodně otázku, na kterou nám má SIM analýza odpovědět. De Souza poukazuje na to, že by mělo jít o otevřenou otázku a ne takovou, jaká vyžaduje jednu konkrétní odpověď. Příkladem takové otázky může být: „Jaké komunikační strategie nejlépe pomáhají uživateli vyhnout se opakování chyb v rozhraní.“ Tedy neptáme se na otázku týkající se konkrétního zkoumaného rozhraní, ale spíše na nějaký širší problém. De Souza zdůrazňuje, že SIM sama o sobě není metodou technickou, ale spíše metodou vědeckou. Měla by proto sloužit spíše teorii, než praxi. To ovšem neznamená, že nemůže posloužit jako doplnění jiné metody (např. CEM) a pomoci, tak upřesnit její poznatky o konkrétním rozhraní.

108 De Souza 2009 s.26.

109 De Souza 2005 s. 26

Po té co je nalezen a definován problém, který bude určovat celé směřování dané analýzy, je třeba zvolit vhodný produkt, který poslouží jako ilustrace zmíněného problému, a také provést jakousi jeho předběžnou inspekci tak, aby byl zasazen do nějakého konkrétnějšího kontextu. Je třeba se předběžně ptát: a) Kdo jsou cílový uživatelé systému. b) Jaké jsou cíle těchto uživatelů, na které je produkt zaměřen.

V přípravné fázi je ještě třeba připravit scénář interakce, který bude vodítkem pro analýzu. Není možné prozkoumávat všechny možnosti daného systému, je třeba si zvolit jednu z činností, kterou systém uživateli umožňuje, aby vyhovovala předem položené otázce. V případě výše položené otázky to může být například triviální problém vytvoření záložní kopie dokumentu v OpenOffice. Daný scénář bude posléze zkoumán v celé své komplexitě, včetně využití nápovědy, případně prozkoumání všech možných postupů pro jeho splnění, včetně těch, které systém přímo nepreferuje.

Analýza poté postupuje v následujících krocích:

- 1) Analýza metalingvistických znaků
- 2) Analýza statických znaků
- 3) Analýza dynamických znaků
- 4) Srovnání designerovy metakomunikace, tak jak vychází z předchozích analýz.
- 5) Závěrečné zhodnocení komunikační schopnosti systému.

V prvních třech krocích se analýza zaměřuje na důkladné prozkoumání znaků, skrze které je nám designerem předáváno metakomunikační sdělení. V každé fázi je následována výše zmiňovaná metakomunikační šablona. Analytik se tak v každém okamžiku zaměřuje na to, jak designer dává skrze svého zástupce najevo: Jaké je jeho porozumění tomu, kdo jsou uživatelé jeho systému. Jak pochopil to, co uživatel od systému požaduje, jak nejlépe toho dosáhnout a proč. Jaký je systém, který pro něj na základě pochopení jeho požadavků vytvořil a jak by ho měl uživatel používat, aby co nejvíce odpovídal jeho požadavkům.

V první úrovni jsou rozebírány znaky metalingvistické, tedy takové, skrze které se designer o svém produktu přímo vyjadřuje. Popisuje nám ho a ukazuje, jak ho máme správně používat. Na této úrovni je metakomunikace nejzřejmější. Jsou to například různé podoby on-line a off-line nápovědy. V této fázi je třeba se držet okruhů nápovědy, které

přímo souvisí se zvoleným scénářem. Problémem na této úrovni může být například to, že v případě našeho scénáře musíme hledat nápovědu v různých navzájem nesouvisejících okruzích.

Při analýze statických znaků si všímáme výhradně znaků, ze kterých je vystaveno rozhraní samo, takových, které zůstávají po čas interakce neměnné. V této fázi je třeba dávat pozor na to, abychom oddělovali statické znaky od dynamických kauzálních procesů. Měli bychom si všimnout pouze jejich statické stránky a jak ta je v souladu s metakomunikační šablonou. Důležitý je také ohled na to, co bylo zjištěno v minulé části. Jak dané statické znaky odpovídají tomu, co se říká v nápovědě programu. Dalším problémem, na který je třeba se soustředit, může být například roztržitost jednotlivých znaků v rámci rozhraní, bez jejich viditelné souvislosti¹¹⁰.

V poslední fázi analýzy distribuce metakomunikace v rozhraní se soustředíme na dynamické znaky, tedy na různé vizualizace procesů a změn, ale také na funkce a děje odehrávající se v rozhraní. Dynamické znaky mohou buď podporovat, nebo vyvracet představu uživatele o možných funkcích statických znaků v rozhraní. Proto je třeba se zaměřit na to, jak tyto znaky odpovídají znakům statickým. Je třeba také dávat pozor na to, jestli jednotlivé statické znaky reprezentující dynamické možnosti zůstávají konsistentní po celý průběh interakce. Díky dynamickým znakům je upevňována určitá cesta uživateli sémiózy. Příkladem selhání může být například otevření souboru v programu přetažením myši do jeho okna. Pokud je možné soubor v programu otevřít je nám to dáno najevo změnou tažené ikony. Pokud ale přetáhneme soubor ne do okna programu, ale pouze na ikonu programu na liště (v MS Windows). Ikona se sice změní, ale po uvolnění tlačítka se objeví upozornění, že soubor nelze přetažením na lištu otevřít. Je zřejmé, že zde metakomunikace selhává, jelikož vztah mezi dynamickým a statickým znakem nefunguje správně.

V další fázi přichází vzájemné srovnání jednotlivých metakomunikačních sdělení vyabstrahovaných v předchozí analýze znaků. Cílem je zde nalézt metakomunikační vzorce, jak se prolínají skrze jednotlivé vrstvy znaků. Je třeba zaměřit se zejména na místa, kde dochází k nějaké inkonzistenci a tím k potencionálnímu přerušení plynulé komunikace. Jednotlivá metakomunikační sdělení nemusejí být v žádném případě identická, a ani nemohou, jak to vyplývá z rozdílné povahy znakových systému, ve kterých se objevují. V žádném případě by se ale neměla navzájem popírat, nebo tvrdit něco jiného. V této fázi

110 De Souza 2009 s.52.

analýzy je proto třeba se zaměřit na rozhraní jako celek, jak v jeho rámci působí jednotlivá sdělení. De Souza upozorňuje, že interaktivní systémy mají každý unikátní znakový systém, který je výtvozem designera. Je třeba zde obzvláště dbát na to, aby se v tomto systému nevyskytovali nepravidelnosti, které by zákonitě vedli ke zmatení uživatelů.¹¹¹

De Souza navrhuje sadu otázek, které mohou sloužit jako vodítko pro zhodnocení daných metakomunikačních znaků.

- 1) Je pravděpodobné, že uživatel bude interpretovat znak jinak, než bylo zamýšleno? Jak a proč?
- 2) Bude tato interpretace stále v souladu se záměrem designu?
- 3) Připomíná tento způsob interpretace jiné způsoby použité při analýze rozhraní? Jaké a proč?
- 4) Je možné díky sémiotické analýze získat nějaké obecnější třídy dynamických a statických znaků? Jaké?
- 5) Je možné nalézt statické, nebo dynamické znaky, které jsou zařazeny do nesprávných tříd? (Těch které byly definovány v předchozím bodě). Může takovéto chybné zařazení ovlivnit komunikaci v systému? Jak?¹¹²

V poslední fázi dochází k rekonstrukci celého metakomunikačního sdělení. K rekonstrukci systému znaků, který je v rozhraní požit a k následnému zhodnocení toho, jestli dané rozhraní efektivně slouží, jako prostředník designera pro komunikaci s uživatelem. To zejména s ohledem na původně položenou obecnou otázku.

2.4.7.2 Metoda hodnocení komunikativnosti (CEM)

Výše zmíněná metoda sémiotické inspekce spočívá v důkladné analýze rozhraní odborníkem v oboru HCI, který vychází ze svých zkušeností a znalostí dané metody. Její záběr je proto velmi široký. Člověk zabývající se analýzou může podrobně zkoumat jednotlivé části rozhraní a testovat různé scénáře. CEM je oproti tomu založena na pozorování a analýze chování skutečných uživatelů při používání programu. Je tedy svým způsobem omezena, jelikož je zde zkoumána jenom jedna interakční cesta odehrávající se

111 De Souza 2006

112 Tamtéž s.153.

v omezeném čase (cca 30 minut). Důležitá ale není ani tak šířka záběru, jako hloubka analýzy. Zatímco metoda SIM zkoumá, jaké je metakomunikační sdělení designera v celé jeho šíři, CEM se věnuje tomu, jak je toto sdělení přijímáno uživatelem.

Jak bylo řečeno výše, důležitým pojmem je pro teorii SI tzv. sémióza, tedy neustálé řetězení přiřazování významů a proud asociací, který probíhá jak na straně uživatele, tak na straně designera. Ideálním stavem je, pokud uživatelova sémióza probíhá přesně tak, jak bylo designerem očekáváno. To je samozřejmě případ, kterého není nikdy absolutně dosaženo. CEM na základě pozorování chování uživatelů hledá místa, kde je uživatelova sémióza nějak odkloněna od ideálního průběhu, případně zastavena díky neporozumění designerovu vzkazu. Tedy místa, kde metakomunikace selhává a nefunguje tak jak má.

SIM a CEM se v žádném případě nenahrazují, spíše doplňují. Jedna popisuje šíři sdělení a druhá to, jak je sdělení přijímáno. Přičemž SIM je schopna identifikovat místa, kde bude pravděpodobně docházet k chybám.

Ústředním termínem CEM je tzv. „komunikativnost“ (communicability), kterou ovšem není možné chápat pouze jako nějaký vágní pojem. Autoři sémiotického inženýrství ji definují následujícím způsobem:¹¹³

„communicability is the distinctive quality of interactive computer-based systems that communicate efficiently and effectively to users their underlying design intent and interactive principles.“¹¹⁴

Tuto definici posléze ještě doplňují a zpřesňují jako:

„... designers deputy capacity to achieve full metacomunication, conveying to user the gist of the original designer's message. (...) Communicability applies to both interpretative and expressive codes that the designers deputy handles for generating and interpreting messages during situated interaction with users“¹¹⁵

Je vidět, že je zde kladen důraz na několik faktorů. V prvním sledu poněkud vágní požadavek umožnění efektivní komunikace je posléze rozšířen ta to, že se jedná o vlastnost designerova zástupce která se realizuje jak na úrovni interpretativních, tak na úrovni expresivních kódů. Tzn. Na úrovni komunikace program → uživatel, kdy je nám skrze

113Obě definice ponechávám záměrně v originálním znění

114 De Souza 2005

115 De Souza 2009 s.114.

rozhraní dáváno designerem najevo, jak a proč máme s rozhraním zacházet a zároveň i na úrovni komunikace uživatel → program, kdy jde o porozumění systému tomu, co mu chceme sdělit. Tzn. Při pohledu z jiného úhlu, jde i o porozumění uživatele kódům, které mu byly designerem připraveny proto, aby mohl komunikovat s jeho zástupcem.

Samotné testování komunikativnosti spočívá opět v několika fázích. Je to fáze přípravy, fáze štítkování (tagging), fáze interpretace a fáze tvorby sémiotického profilu. V první části je opět třeba vybrat nějaké konkrétní rozhraní a provést jeho zběžnou inspekci, abychom byly schopni vybrat jeho cílové uživatele a hlavní úkoly. Posléze je třeba zvolit nějakou konkrétní funkci, která bude testována. Testování probíhá pozorováním 3 – 10 uživatelů při plnění daného cíle v rámci rozhraní. Pozorování může probíhat v reálném čase případně z nahrávky interakce. Důležitou částí jsou také rozhovory s uživateli, které pomohou objasnit některé jejich volby a motivace při užívání programu. Je možné testování provádět jak na hotovém programu, tak i na papírovém prototypu. Papírové prototypy jsou v mnoha ohledech vhodnější, jelikož zde uživatel musí daleko více verbalizovat to, co právě dělá a proč.

Výsledný záznam je potom analyzován a hledají se jednotlivá selhání v komunikaci, která jsou posléze označena třinácti možnými štítky podle charakteru selhání ke kterému došlo. Nejprve zde uvedu jejich výpis s českým překladem a posléze každý z nich rozeberu podrobněji. Jsou to:

- 1) „Vzdávám to.“ - „I give up.“
- 2) „Takhle se to zdá v pořádku.“ - „Looks fine to me.“
- 3) „Děkuji ale nechci.“ - „Thank, but no thanks.“
- 4) „To můžu udělat i jinak.“ - „I can do otherwise.“
- 5) „Kde to je?“ - „Where is it?“
- 6) „Co se stalo?“ - „What happend?“
- 7) „Co teď?“ - „What now?“
- 8) „Kde to jsem?“ - „Where em I?“
- 9) „Sakra!“ - „Oops!“

- 10) „Takhle to nepůjde.“ - „I Can't do it this way.“
- 11) „Co je to?“ - „What is this?“
- 12) „Pomoc!“ - „Help!“
- 13) „Proč to nedělá to co má?“ - „Why doesnt it?“

Jednotlivé kategorie selhání jsou označeny výrazy z přirozeného jazyka tak, jak by je pravděpodobně označil sám uživatel. Tento způsob pomáhá výzkumníkovi, který provádí jejich analýzu nelézt je a správně identifikovat.

Jednotlivá selhání vycházejí obecně z inkonzistence mezi ilokucí a perlokucí, jak uživatele, tak i designerova zástupce. Jinými slovy inkonzistence mezi tím, co je „říkáno“ a s jakým záměrem a tím, jak se to reálně projeví na stavu věcí.¹¹⁶ Autoři rozlišují mezi globální ilokucí a perlokucí, která se týká vyšší úrovně komunikace o obecných cílech a strategiích a lokální ilokucí a perlokucí, kde se jedná o komunikaci o řešení jednotlivých konkrétních problémů. Podle toho, jestli selhává vztah mezi lokální, nebo globální ilokucí a perlokucí, jsou jednotlivá selhání dělena do tří skupin podle závažnosti. Nyní je postupně vyjmenuji od nejzávažnějších po nejlehčí. U každého ve zkratce uvedu, jakého typu selhání se týká, jaký je jeho obecný projev a příklad klasického symptomu, jak se může projevit.

Nejzávažnější skupinu selhání tvoří ta, kde není globální ilokuce konzistentní s globální perlokucí. To znamená, když se uživatel nechová podle celkové strategie, která je mu designerem navržena. Patří sem:

a) „Vzdávám to!“

- Jde o rozpor mezi globální ilokucí a perlokucí, kterého je si uživatel vědom.
- Uživatel přiznává svoji neschopnost splnit svůj cíl. Stručně by se dalo říci, že mu dojde nálada se snažit.
- Typickým příznakem je ukončení interakce.

b) „Takto se to zdá v pořádku.“

- Ještě závažnější selhání než předchozí, jelikož uživatel si není vědom toho, že k selhání došlo a stále pokračuje v interakci. Nutně ale špatným směrem.

¹¹⁶De Souza 2005 s.122.

- Uživatel je přesvědčen, že vyřešil svůj problém, ale přehlédl chybu, ke které během toho došlo.
- Nemá žádný konkrétní symptom, spíše vede v průběhu interakce k řadě dalších selhání vycházejících z toho, že uživatel sleduje špatnou cestu.

Další skupinou selhání jdou ta, při kterých uživatel sice pochopí celkový záměr designera, ale ilokuce a perlokuce se míjejí v konkrétních úkolech.

a) „Kde to je?“

- Uživatel hledá způsob, jak sdělit svoji ilokuci designerovu zástupci. Jeho sémióza se dočasně zastavuje.
- Uživatel v rámci své sémiózy očekává, že uvidí znak, který mu umožní pokračovat tak, jak si myslí, že by pokračovat měl, ale nenachází ho.
- Typickým symptomem je zdánlivě bezcílné otevírání různých nabídek a hledání v nich.

b) „Co se to stalo.“

- Uživatel nechápe designerovu ilokuci. Jeho sémióza se zastavuje.
- Uživatel nevidí žádný výstup operace, kterou provádí. Proto ji několikrát zdánlivě bezvýsledně opakuje. Případně výstup vidí, ale nechápe ho, jako výstup dané operace.
- Typický symptom je např. spouštění stejné funkce několikrát za sebou.

c) „Co teď?“

- Uživatel nemůže nalézt záměr pro svou ilokuci. Jeho sémióza je opět zastavena.
- Neví co má v nastalé situaci dělat a proč.
- Náhodné pokusy o interakci, které zjevně nesledují žádný přesný účel.

d) „Kde to jsem.“

- Ilokuce pronesená sice s dobrým úmyslem, ale ve špatném kontextu. Uživatel se musí zastavit a znovu svou ilokuci promyslet.

- Uživatel se pokouší vykonávat nějakou funkci, ale v jiném kontextu.
 - Snaha o provádění nějaké činnosti v nesprávném modu aplikace (např. snaha formátovat text v náhledu tisku).
- e) „Sakra!“
- Uživatel se sice pohybuje ve správném kontextu, ale jeho ilokuce je špatná. Musí ji proto změnit.
 - Uživatel udělá chybu, kterou si ovšem vzápětí uvědomí.
 - Typickým symptomem, je například stisknutí tlačítka zpět.
- f) „Takhle to nepůjde.“
- Uživatelova ilokuce je špatná v rámci delšího sledu kroků, je ji proto třeba znovu formulovat.
 - Uživatel si najednou uvědomí, že postupoval špatně a musí se vrátit na začátek.
 - Stejně jako v předchozím případě může být symptomem stisknutí tlačítka zpět ovšem tentokrát několikanásobné, vracející se k výchozímu bodu celého procesu.
- g) „Co to je?“
- Uživatel zkoumá designerovu ilokuci a očekává od něj nějaké metakomunikační sdělení.
 - Uživatel čeká, že mu designer nějak napoví, jak má postupovat dál.
 - Typicky například najetí myši na ikonu v očekávání, že se objeví okénko s nápovědou (tzv. tooltip)
- h) „Pomoc!“
- Uživatel se snaží pochopit designerovu ilokuci a vědomě se uchyluje k jeho metakomunikačním sdělením.
 - Uživatel neví, co má v daném okamžiku dělat a záměrně se uchyluje k on-line, nebo offline nápovědě.
 - Symptomem je například stisk tlačítka F1 (v systému Windows vyvolává

nápovědu).

i) „Proč to nedělá co má?“

- Uživatel nechápe designerovu ilokuci a snaží se sám nalézt její smysl.
- Uživatel zpravidla opakuje krok po kroku cestu, která vedla k nějakému neúspěchu a snaží se pochopit, proč k němu došlo.
- Uživatel například znovu načte původně uložený dokument a opakuje již jednou vykonanou činnost.

Poslední skupinou selhání jsou ta, při kterých nedochází k žádné inkonzistenci mezi lokální, ani globální ilokucí a perlokucí a přesto dojde k selhání komunikace.

a) „Můžu to udělat i jinak.“

- Uživatel nerozumí úplně ilokuci designera týkající se úkolu, který právě vykonává, proto ho vykonává „po svém“.
- Uživatel nechápe, že dělá něco špatně. Chyba se může objevit až po delší době.
- Uživatel dosáhne svého cíle, ale ne optimálním způsobem. Například může jít o typické formátování wordového dokumentu pomocí klávesy tab a enter a ne pomocí k tomu určených funkcí.

b) „Děkuji, ale nechci.“

- Uživatel rozumí designerově ilokuci, ale i přesto se rozhodne jednat jinak.
- V zásadě nedochází k žádnému nedorozumění, komunikace je neúspěšná, protože uživatel nesouhlasí s designerem.
- Může například jednat o využití vlastností operačního systému namísto vlastností programu.

V další části analýzy dochází k podrobnějšímu zkoumání výskytu jednotlivých štítků. Důraz je kladen zejména na:

- 1) Frekvenci výskytu jednotlivých selhání. Pokud nějaký štítek vyskytuje častěji, vypovídá to o komplexním problému celého rozhraní. U některých typů se může

jednat pouze o selhání znakového systému (např. Co je to? Kde je to?). U těchto je pravděpodobné, že postupem času vymizí s tím, jak se uživatel bude se systémem více seznamovat. Některé ale naznačují, že je problém v celém metakomunikačním sdělení. Například v případě „Mohu to udělat i jinak.“, nebo „To vypadá v pořádku“ se jedná o selhání designera zástupce, který není schopen navést uživatele na správnou cestu.

- 2) Kontext výskytu jednotlivých příznaků selhání. Autoři například zmiňují pravidelný výskyt štítku „Kde to jsem?“ ve stejném kontextu. Ukazuje, že designer nebyl schopen správně předpovědět způsob, jakým se bude s velkou pravděpodobností ubírat uživatelova sémióza¹¹⁷.
- 3) Rozpoznání určitých vzorců navzájem se následujících selhání. Určité sekvence selhání ukazují, že sémióza uživatele není designerovým zástupcem podporována správným směrem. Ačkoli například výskyt jednotlivých štítků nemusí znamenat fatální selhání. Výskyt určité jejich sekvence už ano.
- 4) Úroveň selhání s ohledem na to, jakým způsobem ovlivňují splnění cíle rozhraní. Jestli se jedná o selhání (a) na strategické úrovni, tedy na vyšší úrovni celkového účelu, který má daný systém plnit, nebo (b) na taktické úrovni, tedy při provádění jednotlivých parciálních cílů, nebo (c) na operativní úrovni, tedy na úrovni jednotlivých operací. Zde mohou napovědět předchozí body. Například určité sekvence selhání mohou značit problém na taktické úrovni. Jiné spíše na strategické.

Poslední částí CEM analýzy je pak vytvoření sémiotického profilu rozhraní. Člověk, který hodnocení provádí se postaví do role designera a pokusí se odpovědět na následující otázky.

- 1) Kdo si myslím, že jsou uživatelé mého produktu? - Odpověď na tuto otázku ve světle analýzy jednotlivých selhání může odhalit rozpor mezi tím, jak si designer původně představoval své uživatele, a tím jací jsou uživatelé doopravdy. Pokud designer například vytvářel rozhraní pro profesionální uživatele, ale při analýze se objevuje zvýšený výskyt štítku „Pomoc!“ je zřejmé, že skutečná cílová skupina je jiná.
- 2) Jak jsem pochopil potřeby uživatelů, pro které produkt navrhují? - Tato otázka

117 De Souza 2005 s.139.

ukazuje nakolik je realizována škála perlokucí obsažená v ilokucích, tedy jinými slovy nakolik odpovídají jednotlivé funkce programu potřebám uživatelů.

- 3) Jaké jsou preference uživatelů s ohledem na jejich potřeby. - Tato otázka pomáhá uvědomit si, zda-li byl zvolen správný znakový systém a systém pro vyjádření jednotlivých ilokucí. Například častý výskyt štítku „Děkuji ale nechci.“ Znamená, že designer nezvolil správný způsob komunikace s uživatelem svého programu a ten se uchyluje k vlastním řešením.
- 4) Jaký je tedy systém, který jsem pro uživatele vytvořil a jak ti by ho měli používat? - Tato otázka pomůže odpovědět na to, jak se designerovi daří komunikovat jednotlivé funkce rozhraní. Například častý výskyt selhání typu: „Co je to?“, „Pomoc!“, „Co se to stalo?“. Ukazuje, že tato komunikace nefunguje v celku rozhraní zprávně.
- 5) Jaká je vize mého designu? - Tedy jak se designerovi daří sdělit uživateli celkovou vizi toho, co mu předává.

Ve fázi sémiotického profilování přicházejí ke slovu výše zmíněné teorie. Ecovy parametry teorie znakové produkce, jakobsonovy komunikativní funkce, nebo gricovy maximy. Za pomoci těchto teorií může designer poznat, kde pramení jednotlivá nedorozumění. Člověk provádějící analýzu může postupně procházet rozhraní a kontrolovat, zdali některé jeho části nevybočují z optimálního prostoru, který je designu rozhraní výše uvedenými teoriemi vymezen. Může také, na jejich základě navrhnout konkrétní řešení.

3 Pravidla sémiotického inženýrství

3.1 Postup tvorby

Pro získání konkrétních pravidel jsem se rozhodl přistoupit k jednotlivým částem celku sémiotického inženýrství samostatně a z každé z nich vybrat několik doporučení. Jako první jsem vybíral z teorie SI, z jednotlivých částí její ontologie, jak byly výše popsány. Tato pravidla, díky čistě teoretické povaze svého zdroje, jsou spíše obecná a není u nich třeba nějaká generalizace. Další sadu jsem vybral z metod. Soustředil jsem se na chyby, které dané metody označují. Vycházel jsem zejména z jednotlivých kategorií selhání podle metody CEM a z nejčastějších problémů odehrávajících se mezi metalingvistickými, dynamickými a statickými znaky. Posléze jsem je invertovat a ke každému selhání uvedl způsob, jak se mu vyhnout. V mnoha případech se tato vzniklá řešení překrývala, proto umožnila vytvořit sadu obecnějších vět. Podobně jsem postupoval při výběru posledního souboru, kde jsem čerpal z publikovaných výsledků praktických aplikací metod sémiotického inženýrství. Vycházel jsem opět z konkrétních chyb, které byly autory označeny a snažil se na ně nalézt obecnější řešení. Zejména v této fázi by pravděpodobně bylo možné nalézt více výsledku, například na základě vlastního výzkumu.

Co se týče formulace, zvolil jsem oslovení v pátém pádě tak, aby se pravidla podobala například greicovým maximám. Snažil jsem se také co nejvíce vyhnout odborným výrazům z teorie sémiotického inženýrství a sémiotiky obecně, jelikož cílovou skupinou jsou lidé, u kterých se v tomto oboru nepředpokládá žádný přehled. Cílem zvoleného způsobu formulace je neodradit posluchače hned začátku, ale poskytnout mu podnět k přemýšlení o jeho práci.

V souboru osmnácti vzniklých pravidel se přirozeně vydělila pravidla týkající se komunikace samotné a ta týkající se tvorby znakového systému, pomocí kterého je komunikace vybudována. Přesně podle ecova rozdělení na tvorbu znakových systémů a komunikaci v jejich rámci.

Vyjmenuji nyní jednotlivá pravidla a ke každému uvedu pouze ve zkratce část teorie, které se týká. Pravidla budou v následujícím výčtu seřazena tak, jak vycházela nejprve z teoretické části posléze z metodologické části a ve finále doplněna z praktických poznatků. Je třeba ještě podotknout, že daná pravidla slouží spíše designerům-programátorům malých projektů, kde rozhodnutí spočívá na malé skupince lidí a je úzký

vztah mezi tím kdo problém řeší, navrhuje vzhled a programuje. V mnoha případech může jít o jednu a tu samou osobu.

3.2 Pravidla

3.2.1 Pravidla odvozená z teoretické části

Níže vyjmenovaná pravidla jsou vybrána na základě ontologie sémiotického inženýrství. Vycházejí z definic jednotlivých pojmů a vymezení prostoru designu HCI podle SI.

Rozhraní tvoříš proto, aby jsi sdělil(a) svůj originální pohled na daný problém. Pochlub se tím, že je to tvoje práce.

Zde se naráží na celou definici rozhraní v rámci SI. Tedy rozhraní jako designerova zástupce, skrze kterého designer promlouvá. Sděluje, jak pochopil daný problém a jaké na tento problém vymyslel řešení. To, je podle mého názoru jedna ze základních věcí, které by si měl designer-programátor uvědomit. Nevypouští do světa nástroj, ale i své originální pojetí problému a jeho řešení. Apel na to, aby se svým výsledkem chlubil a ukázal, že je to jeho práce, má za úkol povzbudit vědomé projevení-se designera v rozhraní. Pokud se bude chtít chlubit, tak nejspíše tak, že popíše jakým způsobem problém uchopil a s jakým originálním nápadem přišel. Tento apel může svým způsobem již od počátku tvorby naladit designera správným směrem a podpořit metakomunikaci mezi ním a uživatelem.

Nepodceňuj nápovědu. Spolu s tím, jak budeš program vytvářet zanášej do ní všechny jeho možnosti. Na začátku nápovědy dej uživatelům jasně najevo, pro koho je určen, jaké úkoly má plnit a proč.

Zde se naráží na tu nejexplicitnější formu metakomunikace probíhající skrze offline nápovědu. V té se může designer nejvíce projevit. Nápověda do velké míry ovlivňuje to, jakým způsobem bude uživatel k danému rozhraní přistupovat. Pokud je vyčerpávající ale zároveň přehledná, umožňuje uživateli nalézt všechny možnosti, které do ní designer zakomponoval a umožňuje mu plně pochopit jazyk, kterým k němu mluví. Pokud bude nápověda vznikat paralelně se samotným produktem, bude tím zajištěno, že designer v ní například neopomene nějakou funkci zmínit. Proces návrhu nápovědy spolu s návrhem rozhraní může také sloužit jako reflexe doposud odvedené práce a může designerovy již ve

fázi tvorby pomoci odhalit budoucí neshody (respektive jim předejít).

Uživatel musí být samozřejmě ochoten investovat do pročítání nápovědy svůj čas. Ne každý je, a proto, ačkoli je nápověda důležitá, není jediným způsobem metakomunikace, na který je třeba se zaměřit. V každém případě ale, pokud bude uživatel odkázán na prohlížení webových fór a hledání rad v internetu, nikdy nebude s daným rozhraním spokojena a nebude schopen s ním efektivně pracovat. Designer by také měl hned ze začátku dát najevo, k čemu je daný program určen a pro jakou cílovou skupinu. Například expert ve zpracování fotografií má jiné požadavky a je zvyklý na jiný způsob práce, než amatér.

Nespoléhej na to, že uživatel bude jednat přesně, tak jak si naplánoval. Počítej s tím, že bude experimentovat. Nebraň mu v tom, ale dávej mu vždy najevo, jaká je optimální cesta. Umožni mu ale také používat zkratky.

Tématem tohoto pravidla je proces sémiózy. Z výše uvedené teorie vyplývá, že nikdy není možné podchytit proces uvažování odehrávající se v uživateli v čase interakce. Není proto možné spoléhat na to, že uživatel bude dělat přesně to, co se od něho očekává. Ecova teorie znakové produkce říká, že nedílnou součástí komunikace je porušování daných kódů a experimentování s nimi. Toto by měl tvůrce programu předpokládat. Nemusí uživateli nutně bránit v tom, aby experimentoval, ale vždy by mu měl dávat nenásilně najevo, jaká je optimální cesta a jaké jsou jeho možnosti v daném kroku. Je třeba uživateli poskytnout vodítka pro orientaci nejenom v prostoru rozhraní, ale i v čase, tedy v procesu interakce. Nemělo by se stávat, že se uživatel nechá svoji sémiózou někam zavést a dostane se do situace, ze které nejde cesta dál a musí se vrátit zpět (k tomu viz pravidlo níže).

Než začneš tvořit, rozmysli si základní prvky ze kterých budeš stavět a rámcově navrhni jejich vzhled a to jak budou reagovat na interakci a s jakým typem funkcí budou pracovat. Z nich pak skládej rozhraní jako stavebnici.

Tato rada je podle mě velice důležitá. Týká se samotné tvorby znakového systému ještě předtím, než je použit v praxi. Pokud si vývojář předem připraví jednotlivé prvky (tlačítka, alert boxy atd.), rozmyslí si, jak budou rámcově vypadat a k jakým funkcím budou jednotlivé typy sloužit a až poté, co bude mít daný soubor hotov přejde k tvorbě tokenů v daném rozhraní. Nejde o to tvořit konkrétní znaky označující konkrétní rozhraní, ale spíše tu část znaků, která odkazuje k jejich vlastní funkci, která něco říká o oněch

znacích samotných, o jejich možnostech. Tedy o metakomunikační část jednotlivých prvků rozhraní. Výsledek bude potom konzistentnější, co se týče jednotlivých znaků. Nemělo by pak docházet k problémům v komunikaci s uživatelem plynoucích z nesouhlasu mezi statickými a dynamickými znaky v rozhraní. Základem je představa, že designer tvoří pro uživatele nějaký znakový systém a ten pak používá k tvorbě rozhraní.

Při tvorbě jednotlivých dílů stavebnice využívej maximálně zvyklostí operačního systému na kterém stavíš. Nesnaž se za každou cenu vymýšlet novinky. Stav na tom, co už je obecně známo.

Z ecových modů tvorby znakového systému a z jejich aplikace na problematiku interakce s rozhraními jasně vyplývá, že srozumitelnější jsou pro uživatele systémy, ve kterých se pracuje s již známými znaky, než ty, kde je snaha vytvářet nové. Podobně jako je kultura zásobárnou významů, do které saháme při mezilidské komunikaci. Může operační systém a jeho grafická nadstavba sloužit jako zásobárna znaků pro tvorbu rozhraní. Pokud existuje například nějaký ustálený způsob, jak uživateli oznamovat chyby, nemá cenu vymýšlet nový. Zde také leží problém přenositelných aplikací, které nemohou čerpat z jednoho OS a stávají se tak mnohdy na první pohled nesrozumitelnými.

Přísně dodržuj systém, který jsi stanovil. Pohybuj se jenom v jeho rámci a nikde z něj bez upozornění neslevuj.

Je pro uživatele velice matoucí, když jeho sémiozu navedeme na očekávání nějakého výsledku, ale ten se nedostaví z důvodu výjimky z pravidla. Zde je možné se opřít o gricovy maximy. Například maximu relevance, nebo maximu kvality. Ty jsou v komunikaci sice běžně porušovány, ale to je něco, co nikdy nemůže, mezi tak nesourodými partnery jako je člověk a počítač, fungovat. Designerův zástupce je schopen zpracovat pouze a přesně to, co je v něm zakódováno, proto nepochopí, když bude uživatel přehánět, nebo na oko lhát. Stejně tak, jelikož člověk podvědomě tuto jeho striktní povahu chápe, v žádném případě z jeho strany neočekává podobné porušování pravidel komunikace. Toto pravidlo je značně obecné a lze namítnout, že každý si ho může vysvětlovat jinak, ale to je samozřejmé, jelikož každý designer vytváří svůj systém rozhraní jinak a unikátně a potýká se se specifickými problémy. Pokud někde použije nějaké pravidlo, musí ho svědomitě používat i nadále.

3.2.2 Pravidla odvozená z metody

Následují pravidla vytvořená z metod sémiotického inženýrství. Ty jsou určeny k tomu, aby odhalovaly konkrétní selhání v rozhraní a jejich původ, proto jsou pro tvorbu pravidel, jejichž úkolem je selháním předcházet.

Snaž se vizualizovat proces, kterým doposud uživatel prošel a dávej mu vždy najevo, jaké možnosti jsou pro něj v daném okamžiku relevantní v závislosti na tom, co právě dělá.

Tato rada se týká zejména problémů s navigací, které jsou na výstupu CEM označeny štítky: „Kde je to?“ a „Co teď?“. Tyto poměrně časté typy selhání se týkají situací, kdy se uživatel v průběhu interakce dočasně zastaví, jelikož neví, jak má dále pokračovat. Ne proto, že by neměl dostatečné informace ke splnění svého cíle, ale spíše proto, že mu rozhraní neposkytuje dostatečná vodítka pro pokračování v práci. V prvním případě očekává uživatel v rámci své probíhající sémiózy, že by mělo něco nastat, ale nevidí to. V druhém tápe a neví, jak by měl pokračovat. To, že bude uživateli neustále vizualizováno, čím prošel, mu umožní mít přehled o jeho již proběhlé interakci a hlavně o tom, jak je interpretována systémem (tzn. důležitý je i feedback o tom, jestli uživateli systém rozumí). To může být realizováno např. formou záznamu proběhlých operací, tak jak ho má například v sobě zabudovaný Adobe Photoshop. Je ale pravda, že zde se jedná hlavně o pomůcku umožňující vrátit se zpět v daném procesu a vizualizace interakční cesty je pouze jakýmsi vedlejším produktem.

Je také důležité dbát na to, aby si byl uživatel vždy vědom možností, které leží v dané situaci před ním. To může být provedeno například zešednutím nerelevantních položek menu, nebo zvýrazněním těch, u kterých je pravděpodobné, že by měli být použity. Je zde důraz na designerovu sémiózu, aby se snažil předvídat na základě své představy o uživateli, jak bude chtít postupovat. Jeho zástupce pak může lépe reagovat na nastalou situaci a vést efektivnější konverzaci s uživatelem.

Poskytuj důkladnou a kontextově citlivou¹¹⁸ nápovědu přímo v rozhraní. Nejlépe při prodlení myši.

Toto je celkem konkrétní pravidlo, ale neméně důležité. Je vhodné, aby rozhraní komunikovalo s uživatelem ze svého vlastního rámce (kontextu který vytváří) a nebylo pro

118Context sensitive

radu vždy nutné přepínat do nápovědy a hledat v ní odpovědi. Uživatel často ani neví, na co se má ptát a jak to vyjádřit slovy. Je proto žádoucí umožnit mu, aby se zeptal gestem (najatím myši na neznámý objekt). Designerův zástupce mluví s uživatelem pomocí dynamických a statických znaků přímo v rozhraní a je na uživateli, aby se tento jazyk naučil. Tomuto procesu učení prospívá, když se může designerova zástupce přímo zeptat, co ta a ta položka znamená. Další možností je například ikona otazníku, která zobrazí okénko s nápovědou, při jejím přetažení na daný objekt. Tento systém nápovědy je celkem rozšířený, ale v 90% procentech nepřináší nic nového, je proto třeba, aby designer postupovat při tvorbě nápovědy opravdu zodpovědně. Jeho zástupce musí být připraven odpovídat na všechny možné otázky v závislosti na situaci. Po najetí na tlačítko OK by se nemělo objevit „odešle dialog ke zpracování“ ale spíše „uloží soubor se jménem diplomka.doc do adresáře [C:/dokumenty](#).“

Umožni uživateli dostat se z každé fáze postupu přímo na nápovědu týkající se této fáze, v nápovědě dej jasně najevo, co přechází a co následuje.

Toto pravidlo je podobné předchozímu, ale netýká se tolik významu jednotlivých statických a dynamických znaků na ploše, jako spíše celkovému úkolu, který je třeba splnit. Platí ale to, co bylo řečeno výše. Designer by měl být připraven skrze svého zástupce za jakékoli situace konstruktivně mluvit o tom, jaký je jeho pohled na problém, který uživatel právě řeší a jaká jsou jím navrhovaná řešení a způsoby, jak jich dosáhnout. Pro plynulost konverzace je třeba, aby uživatel nemusel v nápovědě vyhledávat, jelikož stejně jako výše platí, že mnohokrát ani neví, na co se má ptát. Je proto vhodnější, aby se nápověda vždy nalistovala na místo, které je relevantní pro danou činnost. Sama o sobě by také neměla být konstruována mechanicky, pouze jako popis jednotlivých položek menu a jejich funkcí, ale měla by sledovat nějaké předem dané scénáře řešení situací, které designer uživateli připravil.

Vizualizuj efekty činnosti ať už úspěšné, nebo neúspěšné v místě, kde daná činnost probíhá.

Častým problémem při užívání nějakého rozhraní je to, že akce uživatele vyvolá nějakou změnu, ale zůstane nepovšimnuta, to může způsobit problémy. Pokud například změním výchozí kódování html stránky v editoru a to se přímo neprojeví na výstupu, můžeme posléze zjistit, že daná stránka je pro prohlížeč nečitelná. Je důležité, aby byly vizualizovány, jak neúspěšné výsledky, tak úspěšné, aby měl uživatel neustálou

kontrolu nad tím, jak efektivní je jeho probíhající sémióza. Selhání v tomto směru jsou metodou CEM často odhalována pomocí štítků: „Co se to stalo?“ a „Proč to nedělá co má?“. Uživatel nevidí následky své činnosti a proto není schopen pochopit správně, jak systém funguje.

Pokud pracuješ s více mody. Dej jasně najevo, které funkce jsou jim společné a které jsou specifické.

Toto pravidlo se týká zejména pojmu kontext. Některé konverzace jsou platné pouze v daném kontextu a ne v jiném. Je proto třeba uživateli dávat jasně najevo, jaké specifické funkce má daný mód a jaké jsou všem společné. Sdělení vždy nějak odkazuje ke kontextu, který je sumou všech možných prvků, které mohou ovlivnit uživatelskou sémiózu. Kontext by měl pomocí svých aktuálně zobrazených prvků směřovat sémiózu správným směrem a neměl by být zavádějící. Například možnost kopírovat-vložit zobrazená při přepnutí programu (např. pro správu médií) do modu prohlížení může zavádět uživatelskou sémiózu nesprávným směrem a umožňuje mu tvořit si mylné závěry o funkci rozhraní.

Poskytuj feedback, dělej to důsledně a konzistentně.

Toto pravidlo souvisí s výše zmíněnou potřebou neustále uživateli sdělovat výsledky jeho činnosti a ujišťovat ho o správnosti, nebo nesprávnosti jeho abdukce. Je zde ale také kladen důraz na to, aby byl feedback důsledný. To znamená, aby jednou nastolené pravidlo v zobrazování reakce na uživatelský vstup bylo vždy plně respektováno. Z vymezení prostoru designu za pomoci ecových modů znakové tvorby vyplývá, že bychom se měli při tvorbě rozhraní vždy přiklánět k systému, který má pevná gramatická pravidla. Takovýto znakový systém by pak měl tvořit systém konverzací, opět závislých na pevných pravidlech. Není možné, a často se to stává, naučit uživatele nějaký způsob, jakým systém odpovídá na jeho vstupy a posléze ho porušovat, nebo plně nedodržovat. To, co je možné v běžné konverzaci, kde jsou pravidla běžně porušována (viz výše), není možné v konverzaci mezi designerovým zástupcem a designerem, jelikož se zde nedá počítat s tím, že: „to uživateli dojde.“ To souvisí i s výše zmíněným pravidlem týkajícím se dodržování vytvořeného systému.

Snaž se statickými prvky naznačovat, jaká je jejich funkce a jaký je jejich vztah k ostatním. Jestli něco vykonávají, nebo čekají na vstup, nebo pouze zobrazují

výstup.

Toto pravidlo vychází zejména z metody SIM, kde je často odhalovaným selháním nesouhlas mezi statickými a dynamickými znaky. Dynamické znaky potvrzují, nebo vyvracejí uživateli jeho předpoklady o funkci statických znaků. Je proto nutné, aby ve svém vzhledu zdůrazňovali, jak se zapojují do dynamických procesů. Také je třeba, aby byly plošně rozmístěny způsobem odpovídajícím logice jejich používání. Tedy pokud dva prvky spolu v daném okamžiku funkčně souvisí, měla by tato souvislost být naznačena v rámci rozhraní. To souvisí i s tvorbou jazyka rozhraní. Musí být navržen tak, aby takovéto prostorové a funkční uspořádání podporoval. Mělo by v něm být jasně definováno, jak bude vypadat prvek který je činný v daném okamžiku a jak bude vypadat prvek, který je v daném okamžiku vůči němu trpný, tedy na kterém je činnost vykonávána a také jak budou vypadat zbylé prvky, které nejsou ani činné ani trpné, ale které slouží pro účely výstupu a vstupu.

3.2.3 Pravidla odvozená z praxe

Při tvorbě následujících pravidel jsem vycházel ze výzkumů publikovaných SERG v posledních deseti letech¹¹⁹.

Počítej s metodou pokus – omyl a zpracovávej chyby tak, aby vždy byla jasná příčina a řešení.

Jak už bylo řečeno, není možné spoléhat na to, že vše půjde na straně uživatele podle designera plánu. Je nutné počítat s tím, že uživatel bude experimentovat se znakovým systémem, který je mu k dispozici. Toto jeho experimentování je třeba udržet v mezích a dávat jasně najevo, kdy ústí do chybových situací. Pokud k takovému kritickému bodu dojde, měl by designérův zástupce nejen oznámit, že došlo k chybě, ale i proč a měl by zároveň nabídnout řešení. Vyskakovací okno s nápisem „Na stránce se vyskytla chyba“ a tlačítkem „OK“ nijak ke konverzaci mezi uživatelem a rozhraním nepřispívá.

Dostatečně rozlišuj znaky ležící vedle sebe. Nepojmenovávej dvě různé věci stejně nebo podobně, pokud nechceš zdůraznit jejich vztah.

V konverzaci mezi uživatele a rozhraním je třeba, aby se rozhraní vyjadřovalo jasně a čistě bez možností omylu, nebo nepochopení. Stejně tak musí být zajištěno to, aby

¹¹⁹Zejména: De Souza a Cypher 2008; De Souza 2010; De Souza, Prates, Barbosa 2000; Prates, De Souza 2000

uživatel co nejméně chyboval ve svých promluvách. V rozhraní, jehož základ tvoří zejména nějaké plošné vizuální znaky má člověk tendenci spojovat si navzájem ty, které vypadají podobně. Pokud tedy dva funkčně nesouvisející symboly vypadají podobně, mate to uživatele v jeho uvažování o možnostech a funkcích rozhraní zejména v počátcích užívání programu, než získá zkušenosti, tápe a snaží se pochytit smysl designera sdělení.

Nedostatečné rozlišení statických znaků bývá velice častý problém. Ztěžuje také možnost zapamatovat si význam jednotlivých prvků a i v pozdějších fázích užívání vede k častým výskytům štítků typu „Sakra.“, kdy se člověk tzv. Překlikne. To je způsobeno tím, že jazyk který je designem vytvářen brání jak uživateli, tak designerovu zástupci, aby se jasně a srozumitelně vyjadřovali. Designerův zástupce, je tak z povahy jazyka, kterým je tvořen, nucen porušovat gricovu maximu způsobu a uživatel svým způsobem také.

Na každý výstup programu měj připravenou možnost vstupu. Zároveň zamez vstupům, na které není připraven výstup.

Nedodržování tohoto pravidla bývá častým zdrojem problému při práci s rozhraním. Jakékoli oznámení, nebo výsledek programu, obecně jakýkoli výstup, by měli umožňovat uživateli, aby na ně mohl reagovat, případně na jejich základě dál pokračovat. Pokud mi například Adobe Photoshop oznámí, že v mém výběru je ve vrstvě ve, které se nacházím, prázdné místo a není zde tedy ani co vybrat. Měl by mi zároveň nabídnout seznam vrstev, ve kterých se pod daným výběrem nějaký obsah nachází. Každé zdělení rozhraní, které neumožňuje reakci odporuje zásadám komunikace v rozhraní a jediná cesta ven je v tomto případě vrátit se o několik kroků zpět a zkusit to znovu. Nemusí ale jít pouze o chybová hlášení. Pokud se například programu zeptám, kolik jsem do dokumentu napsal znaků měla by součástí odpovědi být i možnost vložit tento výsledek do schránky, aby s ním bylo možné dále manipulovat.

Opačně platí, že by si každý designer-programátor měl dávat dobrý pozor, aby uživateli neumožňoval vstup v místech, kde není možné nabídnout výstup. Jinými slovy, designerův zástupce by měl od nás přijímat pouze takové povely, kterým je schopen rozumět. Extrémním příkladem je příkazový řádek v linuxu, který vám dovolí jakýkoli alfanumerický vstup, ale systém odpovídá pouze na určité sekvence (slova). V takovémto případě je uživatel začátečník absolutně ztracený.

Snaž se samotnou strukturou programu dávat najevo řešení problému. Například strukturou menu.

Z definice rozhraní jako intelektuálního artefaktu vyplývá, že sám v obě kóduje pochopení daného problému a jeho řešení. Designerův zástupce by neměl způsob, jakým je schopen řešit daný problém, komunikovat pouze skrze nápovědu. Může toho docílit i samotným rozložením znaků. Například strukturou jednotlivých menu, nebo složením panelů s nástroji. U jednoduchého programu sloužícího k úpravě velikosti obrázků není jediný důvod, proč by mělo být menu rozděleno do složek: soubor, editace, Je sice možné namítnout, že se jedná o standardní způsob, jakým jsou rozložena menu v mnoha systémech a jedním z výše zmíněných pravidel je, co nejvíce využívat možností daného systému. Toto pravidlo se ovšem týká spíše způsobu tvorby znaků než způsobu, jak je skrze tyto znaky komunikováno. Čím je navrhovaný způsob řešení problému v rámci struktury rozhraní více zakódován, tím je pro uživatele jednodušší s ním pracovat a to zejména v počátcích, kdy si nemusí pamatovat, kde se která funkce nachází. Je třeba, ale samozřejmě tuto strukturu navrhovat tak, aby nebyla překážkou pro zkušenější uživatele případně pro ně poskytovala možnosti, zkrácení cesty.

Nesnaž se zaujmout příliš široké publikum. Pokud je to nutné připrav pro ně několik módů a důsledně je odděluj.

Obecně bývá problémem pokud se designer snaží oslovi příliš velkou skupinu uživatelů najednou. Každý má od rozhraní trochu jiné očekávání a každý je na jiné úrovni. Není možné očekávat, že jedno rozhraní stejně pochopí profesionál a nováček v oboru. Součástí tzv. Metakomunikační šablony, která byla citována výše, je apel na porozumění svým uživatelům a přenesení tohoto porozumění do rozhraní. Do jednoho rozhraní nelze vmístit více sdělení o tom, jak designer chápe v rámci své semiózy uživatele a problémy, které potřebují vyřešit. Jedním ze způsobů, jak toho dosáhnout je připravení několika různých módů. Takto například postupují tvůrci vývojářského nástroje pro web Macromedia Dreamweaver. Existuje zde několik různých nastavení rozhraní podle toho, zda li je člověk programátor, grafik nebo kodér. V tomto případě samozřejmě není možné, aby se části rozhraní neprolínaly do různých módů, je tedy třeba dobře abstrahovat z designerovy představy o různých uživateliích to, co mají společné a co je pro různé skupiny specifické.

3.3 Povaha pravidel

Výše uvedená pravidla se formulací příliš neliší od pravidel poskytovaných například teorií UCD. Pro ilustraci jich zde několik uvedu.¹²⁰

- Poskytuj zkratky v menu.
- Poslouchej standarty, pokud není nějaký opravdu důležitý důvod proč bys neměl.
- Automaticky ukládej soubory a nastavení.
- Bohatá vizuální odezva je klíčem k úspěšné přímé manipulaci.
- Používej odkazy pro navigaci a tlačítka pro akce.
- Odděluj důležité položky seznamů grafickými ikonami
- Error nemusí být nutně tvou chybou, ale je tvou zodpovědností.
- Vizuálně komunikuj funkci a chování.

Je zřejmé, že mezi pravidly UCD by se dala najít i taková, která mají mnoho společného s těmi výše uvedenými. Stejně tak lze ale nalézt množství podrobnějších pravidel, která se týkají konkrétních částí rozhraní, jako jsou menu, posuvníky, seznamy, interakce s myší a podobně. Takovéto konkrétní rady nejsou ta výše uvedená schopna poskytnout, jelikož jejich charakter je obecnější. Měla by sloužit spíše jako jakási sada imperativů, kterými by se programátoři měli řídit. Je ale pravda, že i zde se najdou pravidla obecnější a konkrétnější. Zejména, co se týká těch konkrétnějších, by bylo možné některá přidat.

Myslím si ale, že, aby mohla vzniknout rozsáhlejší sada pravidel, která by fungovala jako iterace těch výše uvedených a rozhodovala by, jak volit v různých konkrétních situacích, by bylo třeba zvolit jiný postup a nikdy by nebylo možné dosáhnout konečné sady. Dokáží si ale představit, že by mohl vzniknout nějaký konkrétní manuál, jak postupovat při tvorbě rozhraní. Metoda SIM je schopna zkoumat strategie rozhraní při řešení různých problémů a posuzovat jejich úspěšnost. Bylo by tak možné každé z výše uvedených pravidel rozpracovat do otázek, které by mohli být použity jako základ následné SIM analýzy různých rozhraní a ta by na tyto otázky mohla přinést odpovědi.

¹²⁰ Vybráno z Cooper 2007

Nikdy bychom se ale nedostali ke konkrétním radám typu „Tlačítko uložit by mělo být v poměru X:Y ke zbytku stránky.“ atd. Jelikož, jak už bylo zmíněno při popisu SIM, ta není schopna odpovídat na konečné otázky, ale pouze na otázky otevřené.

Konečné odpovědi o konkrétních prvcích rozhraní jsou něčím, co nelze z teorie Sémiotického inženýrství získat. Jejimi teoretickými základy jsou pojmy jako sémióza a abdukce, které popisují nekonečný a předem neurčitelný proces uvažování. Nedokáže proto poskytovat konkrétní rady pro konstrukci nových rozhraní, konkrétně dokáže mluvit pouze o předmětech svého výzkumu. Zde je schopna poukázat na chyby. Každé rozhraní vychází ze specifické designerovy sémiózy o jeho uživateli, jejich potřebách a cílech. Jeden konkrétní prvek může v rámci jednoho rozhraní v kombinaci s ostatními znaky fungovat a v rámci jiného naopak selhávat.

Výše uvedená pravidla tedy nejsou nějakými konkrétními radami pro „pohodlné designery“, ale jsou spíše návody, jakým směrem upírat svoji pozornost. Je na každém zvlášť, jak konkrétně bude postupovat, aby tvořil v souladu s nimi.

4 Závěr

Cílem této práce bylo uceleně nastínit teorii Sémiotického inženýrství a zasadit ji do kontextu ostatních teorií HCI. Na tomto teoretickém základě pak vypracovat soubor pravidel, která by byla schopna překlenout propast mezi sémiotickou teorií HCI a praxí. Záměrně jsem daná pravidla vybíral ze všech částí teorie i z jejích metod, abych tak pokryl co nejvíce celé její spektrum. Je pravděpodobné, že některé formulace by bylo možné zpřesnit na základě testování v praxi. Bude také ještě třeba daná pravidla nějakým způsobem seřadit do podoby publikovatelné tak, aby mohla sloužit v praxi. Záměrně jsem se vyhýbal ve formulaci jednotlivých pravidel uvádění konkrétních případů, jelikož ty mohou zavádět k tomu, aby byly bezmyšlenkovitě kopírovány a vytratil by se tak jejich základní charakter jako vodítek k zamyšlení. Je ovšem nutno na závěr podotknout, že právě tento jejich neurčitý charakter je tím, co by mohlo bránit jejich reálnému praktickému rozšíření.

Seznam použité literatury

- ANDERSEN, Peter B. Computer Semiotics. *Scandinavian Journal of Information Systems*. 1991, 3, s. 3-30.
- ANDERSEN, Peter B. *A Theory Of Computer Semiotics*. Cambridge : Cambridge University Press, 1997. 445 s.
- ANDERSEN, Peter B. What semiotics can and cannot do for HCI. *Knowledge-Based Systems*. 2001, 14, s. 419-424.
- COOPER, Alan. *About Face : The Essentials of interaction Design*. Indianapolis : Wiley Publishing, 2007. 574 s.
- DE SOUZA, Clarisse. *The Semiotic Engineering Of Human Computer Interaction*. Massachusetts : MIT Press, 2005. 283 s.
- DE SOUZA, Clarisse Sieckenius; LEITÃO, Carla Faria. *Semiotic Engineering Methods for Scientific Research in HCI*. [s.l.] : Morgan & Claypool Publishers, 2009. 136 s.
- DE SOUZA, Clarisse Sieckenius; BARBOSA, Simone D. J.; PRATES, Raquel O. A semiotic engineering approach to user interface languages. *Knowledge-Based Systems*. 2001, 14, s. 461-465.
- DE SOUZA, Clarisse Sieckenius; BARBOSA, Simone D. J.; DA SILVA, S. R. P. Semiotic engineering principles for evaluating end-user programming environments. *Interacting with Computers*. 2001, 13, s. 467-495.
- DE SOUZA, Clarisse Sieckenius, et al. The Semiotic Inspection Method . *Anais do IHC*. 2006, November, s. 19-22.
- DE SOUZA, Clarisse Sieckennius. Can inspection methods generate valid new knowledge in HCI?. *International Journal of Human-Computer Studies*. 2010, 68, s. 22 - 40.
- DE SOUZA, Clarisse Sieckennius; PRATES, Raquel O.; BARBOSA, Simone D.J. A case study for Evaluating Interface Design trough Communicability. *Designing Interactive System* 308 - 316s. 2000, s. 308 – 316.
- DE SOUZA, Clarisse Sieckennius, CYPHER A., *AVI '08: Proceedings of the working conference on Advanced visual interfaces*. 2008, s. 165-172.
- ECO, Umberto. *Teorie sémiotik*. Brno : Janáčkova akademie múzických umění, 407. 2004 s.

- GARET, Jesse J. *Jesse James Garrett's Hidden Agenda* [online]. 2009. Dostupné z WWW: <www.jjg.net>.
- GASSON, Sussan. Human-centered vs. user-centered approaches to information system design. *Journal of Information Technology Theory and Application*. 2003, 2, s. 29-46.
- GRABOWSKI, Susanne; NAKE, Frieder. Human-computer interaction viewed as pseudo-communication. *Knowledge-Based Systems*. 2001, 14, s. 441-447.
- GULLIKSEN, JAN, et al. Key principles for user-centred systems design. *BEHAVIOUR & INFORMATION TECHNOLOGY*. 2003, 6, s. 397-409.
- JAKOBSON, Roman. Linguistics and Poetics. In SEBEOK, T. A. *Style In Language*. Cambridge : MIT Press, 1960. s. 350 - 377.
- NADIN, Mihai. Interface design: A semiotic paradigm. *Semiotica*. 1988, 60-3/4, s. 269-302.
- NORMAN, Donald . *User Centered System Design*. Hillsdale : L. Erlbaum Associates, 1986. Cognitive Engineering, s. 526.
- NORMAN, Donald. *Don Norman's jnd.org* [online]. 1993 [cit. 2010-09-12]. Affordances and Design. Dostupné z WWW: <http://www.jnd.org/dn.mss/affordances_and.html>.
- NORMAN, Donald. Emotion and design: Attractive things work better. *Interactions Magazine*. 2002, 4, s. 36-42.
- PALEK, Bohumil. *Sémiotika : Ch. S. Peirce, C. K. Ogden & I. A. Richards, Ch. W. Morris, H. B. Curry: skripta pro posluchače Filozofické fakulty Univerzity Karlovy*. Praha : Karolinum, 1997. 335 s.
- PRATES, Raquel O.; DE SOUZA, C. S. A method for evaluating the communicability of user interfaces. *Interactions*. 2000, 7, s. 31 - 38.
- RHEINFRANK, J.; EVENSON, S. Designing Languages. In WINOGRAD, T. *Bringing design to software*. MA : Addison-Wesley, 1996. s. 63 - 85.
- SEARLE, John Rogers. *Mysl, mozek a věda*. Praha : Mladá fronta, 1994. 129 s.
- Semiotic approaches to user interface design. *Knowledge-Based Systems*. 2001, 14, s. 415-418.
- VTÍPIL, Adam. *Sémiotické inženýrství: moderní teorie HCI a metody hodnocení uživatelských rozhraní*. Brno, 2009. 60 s. Bakalářská práce. Masarykova universita v Brně.