

Návod k sémantické anotaci NÁDRAŽÍ

Filip Jurčíček, Jiří Zahradil a Libor Jelínek

Aktualizace: 15.2.2006

Obsah

Úvod.....	2
Sémantická anotace.....	2
Dimenze – DOMAIN	4
Dimenze – SPEECH–ACT	5
Dimenze – SEMANTICS.....	10
Úvod do sémantické anotace.....	10
Seznam všech používaných sémantických konceptů.....	10
Poznámky k anotaci.....	17
Dělení promluv na dialogové akty.....	17
Opakovaná informace na začátku věty je implicit_confirmation	17
Kontextová nezávislost sémantické anotace.....	17
Closing a thanking.....	18
Anotace příjezdového času v dotazu na odjezd vlaku.....	18
Anotace rozprostřeného času.....	18
Používání Dialogue Act Editor.....	19
Stavy zpracování dialogů.....	19
Příklad anotačních stavů.....	19
Odevzdání práce – jednoho anotačního balíku.....	19
Detailní popis stavů.....	19
Změny dokumentu.....	21

Úvod

Tento dokument popisuje principy sémantické anotace dialogového korpusu NÁDRAŽÍ, který obsahuje textové přepisy telefonických rozhovorů dotazů na informace o vlakové dopravě ČD.

Text je rozdělen do čtyř částí. První část popisuje z čeho se skládá sémantická anotace textových přepisů promluv řeči. Další části pak popisují pravidla a příklady, jak rozdělovat a anotovat jednotlivé typy frází a způsobů vyjádření. Poslední část ukazuje, jak používat program DAE (Dialogue Act Editor).

Sémantická anotace

Úkolem sémantické anotace je poskytnout podstatné významové informace o každém dialogu z korpusu NÁDRAŽÍ ve zjednodušené podobě. Proto byl vyvinut symbolický popis (způsob anotace), který vyjadřuje sémantickou funkci, tj. význam promluv obou účastníků dialogu (uživatelé i operátora) pomocí speciálního umělého jazyka. Pomocí tohoto jazyka lze vlastně „přeložit“ původní text do formy vhodné pro zpracování počítačem.

Tento umělý jazyk si popíšeme blíže později. Nyní se podíváme, jak vypadá původní text.

V dialogu se vždy střídají promluvy uživatele a operátora. Celá promluva uživatele nebo operátora se nazývá *dialogový turn*. Sémantická anotace se provádí po těchto částech.

Na následujícím příkladu můžete vidět dialog, který obsah 18 turn-ů.

Příklad: *Dialog CD03~Part1~04-000427-075758-1.xml*

- 1: **operator:** informace prosím
2: **user:** dobrý den prosím vás já bych potřeboval spojení ze [station]sušice[/station] do [station]brna[/station] dneska kolem poledne
3: **operator:** v jedenáct čtyřicet čtyři vyjedete ze [station]sušice[/station] dvanáct dvanáct jste v [station]horažďovicích[/station]
4: **user:** dvanáct dvanáct
5: **operator:** z [station]horažďovic[/station] dvanáct dvacet
6: **user:** dvanáct dvacet
7: **operator:** třináct třicet šest jste tady v [station]plzni[/station]
8: **user:** třináct třicet šest [station]plzeň[/station]
9: **operator:** odsud vám jede rychlík ve čtrnáct deset do [station]prahy[/station]
10: **operator:** v [station]praze[/station] na hlavním nádraží jste v šestnáct nula nula
11: **user:** ano
12: **operator:** z [station]prahy[/station] další rychlík šestnáct třicet šest a v [station]brně[/station] devatenáct čtyřicet dva
13: **user:** devatenáct čtyřicet dva [station]brno[/station]
14: **operator:** ano
15: **user:** ano děkuju moc
16: **operator:** není zač
17: **user:** naschle
18: **operator:** naschle

Protože *turn* bývá významově někdy příliš bohatý, je třeba jej občas podle potřeby rozdělit na dílčí *dialogové akty* (dialogue acts, ve zkratce DA), které obsahují jen jednu hlavní myšlenku.

Pravidlo: Promluvy dělíme tak, aby byla rozdělena před spojkou. Např.: a, nebo, apod. by mělo být na začátku nového DA.

„až v jedenáct deset“

„a to by tak nák stačilo tich jedenáct deset z hlavního jo“

„a dá se tam vzít kočárek“

Textové přepisy spontánních promluv jsou často gramaticky nesprávné a neobsahují interpunkční znaménka. Přesto je však nutné z kontextu rozlišovat stejně vypadající otázku od oznámení!

Sémantická anotace se skládá z třech částí, zvaných dimenze. Jednotlivé dimenze poskytují odlišné typy informací, které jsou vzájemně nezávislé (ortogonální). Jsou to dimenze: DOMAIN (konverzační oblast), SPEECH-ACT (obecná významová funkce), SEMANTICS (konkrétní významová funkce).

Na následujícím příkladu je vidět příklad rozdělení turnů na dialogové akty a již hotová sémantická anotace.

Příklad: Dialog CD03~Part1~04-000427-075758-1.xml

1:operator:	informace prosím		
	frame	opening	GREETING
2:user:	dobrý den		
	frame	opening	GREETING
	prosím vás já bych potřeboval spojení ze [station]sušice[/station] do [station]brna[/station] dneska kolem poledne		
	task	request_info	DEPARTURE (FROM (STATION
) , TO (STATION) , TIME)		
3:operator:	v jedenáct čtyřicet čtyři vyjedete ze [station]sušice[/station] dvanáct dvanáct jste v [station]horažďovicích[/station]		
	task	present_info	TIME , FROM (STATION) ,
	TIME , TO (STATION)		
4:user:	dvanáct dvanáct		
	communication	implicit_confirmation	TIME
5:operator:	z [station]horažďovic[/station] dvanáct dvacet		
	task	present_info	FROM (STATION) , TIME
6:user:	dvanáct dvacet		
	communication	implicit_confirmation	TIME
7:operator:	třináct třicet šest jste tady v [station]plzni[/station]		
	task	present_info	ARRIVAL (TIME , TO (STATIO
	N))		
8:user:	třináct třicet šest [station]plzeň[/station]		
	communication	implicit_confirmation	TIME , STATION
9:operator:	odsud vám jede rychlík ve čtrnáct deset do [station]prahy[/station]		
	task	present_info	DEPARTURE (TRAIN_TYPE ,
	TIME , TO (STATION))		
10:operator:	v [station]praze[/station] na hlavním nádraží jste v šestnáct nula nula		
	task	present_info	ARRIVAL (TO (STATION) ,
	TIME)		
11:user:	ano		
	communication	acknowledgement	ACCEPT
12:operator:	z [station]prahy[/station] další rychlík šestnáct třicet šest		
	task	present_info	FROM (STATION) ,
	TRAIN_TYPE , TIME		
	a v [station]brně[/station] devatenáct čtyřicet dva		
	task	present_info	TO (STATION) , TIME
13:user:	devatenáct čtyřicet dva [station]brno[/station]		
	communication	implicit_confirmation	TIME , STATION
14:operator:	ano		
	communication	acknowledgement	ACCEPT
15:user:	ano		
	communication	acknowledgement	ACCEPT
	děkuju moc		
	frame	thanking	---

Návod k sémantické anotaci NÁDRAŽÍ

16:operator:	není zač		
	frame	thanking	---
17:user:	naschle		
	frame	closing	---
18:operator:	naschle		
	frame	closing	---

Na tomto příkladu můžeme u každého segmentu příslušejícímu určitému dialogovému aktu najít trojici představující kompletní sémantickou anotaci. Tato trojice (například: „communication, acknowledgement, ACCEPT“ představuje tři dimenze v pořadí DOMAIN, SPEECH-ACT a SEMANTICS.

Poznámka: V dalším textu se u příkladů vyskytují následující symboly:

- O;**, **U;** - pro zjednodušené označení řečníka v souladu s barevným rozlišením textu anotace
(O=operátor, značen červeně; U=uživatel, značen modře)
- (?), (.), (!) - pro upřesnění funkce segmentované části textu, zdali je to z kontextu otázka, odpověď nebo pokyn. Interpunkční znaménka jsou pomyslná, v textu se neobjevují, proto jsou v závorkách.

Dimenze – DOMAIN

Dimenze DOMAIN přiřazuje každou promluvu do jedné ze tří konverzačních oblastí. Máme tyto konverzační oblasti: *task*, *communication*, *frame*.

Task – tento tag (symbolické označení) se použije tehdy, pokud odpovídající promluva přispívá k úspěšnému posunu k cíli dialogu. Pro označení promluvy tímto tagem se rozhodneme v případě, když promluva poskytuje novou informaci, která nebyla ještě předtím zmíněna. Například se takto označí promluva, ve které se objevuje dotaz na čas odjezdu nějakého vlaku. S tímto tagem se nejčastěji pojí následující tagy pro SPEECH-ACT dimenzi: *request_info*, *present_info*, *offer*, *acknowledgement*.

Communication – tento tag popisuje promluvu, která souvisí s udržováním bezchybné komunikace. Například lze použít k označení věty, která vyjadřuje ověření již sdělené informace pro kontrolu, jestli bylo dobře porozuměno. Toto tzv. implicitní potvrzení je velmi časté. Vyskytuje v podobě, že uživatel nebo operátor opakuje poslední údaj bez očekávání odezvy od svého protějšku. Přepokládá, že platí pravidlo: „mlčení je souhlas“. Typickým příkladem je prosté opakování sděleného času odjezdu vlaku. S tímto tagem se nejčastěji pojí následující tagy pro SPEECH-ACT dimenzi: *explicit_confirmation*, *implicit_confirmation*, *acknowledgement*.

Frame – tento tag popisuje promluvu, která má pouze formální nebo obsahově nevýznamný věcný charakter a v podstatě přímo nepřispívá k posunu informační výměny k cíli dialogu. Například pozdrav, děkování, omluva nebo instruování uživatele. S tímto tagem se nejčastěji pojí následující tagy pro SPEECH-ACT dimenzi: *instruction*, *apology*, *opening*, *closing*, *thanking*, *speech_repair*, *status_report*.

Dimenze – SPEECH–ACT

Dimenze SPEECH-ACT přiřazuje každé promluvě hodnotu, která vyjadřuje její funkci z hlediska samotné komunikace. Tedy bez ohledu na to, co konkrétně daná promluva obsahuje. Tato dimenze rozlišuje třeba i takové promluvy, které mají stejnou SEMANTICS dimenzi. Například hodnoty *request-info* a *present-info* rozlišují otázku od odpovědi se stejným věcným obsahem v SEMANTICS dimenzi: *DEPARTURE(TIME, FROM(STATION))*.

Pro tuto dimenzi jsou definovány následující tagy: *request_info*, *present_info*, *offer*, *acknowledgement*, *status_report*, *explicit_confirmation*, *implicit_confirmation*, *instruction*, *apology*, *opening*, *closing*, *thanking*, *speech_repair*.

Apology – promluva ve které se uživatel nebo operátor omlouvá že nemůže zodpovědět položený dotaz, protože to nemůže zjistit nebo jej nepochopil (což se vztahuje k *frame*), nebo že něco neslyšel (*communication*).

Příklad:

O: „bohuzel, to vám nepovím“ (?)
frame apology ---

O: „moment jak to myslíte jaký vagóny“ (?)
frame apology ---

Acknowledgement – tento tag odpovídá nejčastěji promluvě, kdy se přijímá nebo zamítá platnost nějaké předem dotazované informace či nabízené možnosti (při *DOMAIN=task*), nebo v případě vyjádření, že předem oznámená informace vyhovuje požadavkům (při *DOMAIN=task*), případně promluvě, která potvrzuje předchozí dotaz na ověření správného pochopení sdělené informace (*DOMAIN=communication*). Předpokládá se přijetí nebo zamítnutí všech vyjmenovaných parametrů současně. Tento tag se používá i pro výběr varianty z předchozí nabídky označené **Verify**, kde vlastně jde buď o zopakování vybrané možnosti, nebo o upřesnění či zúžení nabízeného intervalu. V tomto případě většinou v sémantice není *ACCEPT* ani *REJECT*, ale pouze konkrétní hodnota.

Příklad:

U: „a to by tak něk stačilo tich jedenáct deset z hlavního jo“ (.)
task acknowledgement ACCEPT (TIME, FROM (STATION))

Uživatel informaci přijímá jako vyhovující.

U: „spíš ten rychlík“ (.)
task acknowledgement TRAIN_TYPE

Uživatel vybral z nabídnutých variant.

O: „ano“ (.)
communication acknowledgement ACCEPT

V tomto případě se jedná o potvrzení, že bylo dobře porozuměno.

Closing – označuje promluvu, která ukončuje dialog. Takováto promluva má dimenzi SEMANTICS prázdnou. *Closing* má vždy hodnotu *DOMAIN* rovní *frame*.

Příklad:

U: „nashle“ (.)
frame closing ---

Explicit confirmation – reprezentuje promluvu, která se přímo dotazuje na bezprostředně sdělenou informaci formou ověřovací otázky. Používá se pro ověření správného zachycení informace a je očekáváno následné potvrzení.

Příklad:

O: „odpoledne jste říkala že jo“ (?)

communication explicit_confirmation TIME

Implicit confirmation – reprezentuje promluvu, která se nepřímou dotazuje na bezprostředně sdělenou informaci formou zopakování. Používá se pro ověření správného zachycení, příp. pochopení informace, ale v případě, že je vše v pořádku, si nevynucuje následné potvrzení. Tento tag reprezentuje tedy promluvu, kdy opakováním již sdělené informace se chceme utvrdit, že jsme vlastně slyšeli dobře. Například uživatel opakuje poslední oznámený čas odjezdu vlaku. V tohoto tagu neočekáváme odezvu od svého protějšku pokud vše je v pořádku. Pokud je chyba v zopakované informaci, potom očekáváme, že nám protějšek dá vědět. Často nejsme schopni rozlišit, jedná-li se o dotaz nebo oznámení, ale v obou případech to má takto stejný smysl.

Příklad:

U: „v jedenáct deset z hlavního“ (?.)

communication implicit_confirmation TIME, FROM(STATION)

Instruction – tento tag odpovídá instruktážní promluvě, která prezentuje, jak se má protějšek zachovat, jak má mluvit, nebo co má udělat pro získání požadované informace. Tento tag lze použít i v případě, kdy se uživateli vysvětluje jak má sdělenou informaci využít, např.: jakým způsobem má přestoupit, resp. přejít na jiné nádraží, jak si koupit jízdenku, zkusit stihnout, atd.

Příklad:

O: „no když budete na mě paní křičet tak najdu“ (!)

frame instruction ---

O: „tam musíte přejít na [station]masarykovo nádraží[/station]“ (!)

task instruction TRANSFER(STATION)

Offer – tento tag odpovídá promluvě, kdy operátor nabízí uživateli nějakou další možnost, o kterou vlastně nežádal a v případech kdy operátor nabízí **změnu zadání dotazu** tak aby uživateli pomohl (více spojů, rychlejší spoje, levnější spoje atd.) Velice speciální tag, který se pozná většinou pouze z kontextu. Je to například když operátor nabízí, zda chce uživatel také zpáteční vlak aniž se na to uživatel ptal. Typické je použití po negativní odpovědi, kdy operátor nabídne změnu zadání (ne v neděli dopoledne nic nejede ale můžete jet odpoledne)

Příklad:

O: „ne v neděli dopoledne nic nejede“

task acknowledgement REJECT (TIME)

O: „ale můžete jet v neděli“

task offer DEPARTURE (TIME)

O: „mám najít i zpáteční vlaky“

task offer BACK

O: „lepší to bude přes domažlice“

task offer THROUGH (STATION)

Opening – označuje promluvu, která zahajuje dialog. *Opening* má vždy hodnotu DOMAIN rovnou *frame*. *Opening* může obsahovat koncepty: GREETING, AREA, PERSON. Úvodní hláška informace se anotuje také jako opening/GREETING. Nezapomínejte anotovat ve správném pořadí – rozlišujte pokud je jméno, pozdrav a pokud člověk nejdřív pozdraví a pak se představí.

Příklad:

O: „informace prosím“ (.)		
frame	opening	GREETING
U: „dobrý den“ (.)		
frame	opening	GREETING

Present info – tento tag reprezentuje oznamovací promluvu, která odpovídá poskytnutí informace. To je například velmi často odpověď na *request_info*.

Příklad:

O: „v jedenáct čtyřicet čtyři vyjedete ze [station]sušice[/station] dvanáct dvanáct jste v [station]horažďovicích[/station]“ (.)		
task	present_info	TIME, FROM(STATION), TIME, TO(STATION)

Request info – tento tag reprezentuje dotazovací promluvu, která odpovídá požadavku na poskytnutí nějaké informace. Vždy obsahuje předmět dotazu (objekt) a specifikaci jeho atributů (hodnoty).

Příklad:

U: „prosim vás já bych potřeboval spojení ze [station]sušice[/station] do [station]brna[/station] dneska kolem poledne“ (.)		
task	request_info	DEPARTURE (FROM(STATION), TO(STATION), TIME)

Pozor první výraz je **vždy** na co se ptám a dále následují výrazy, které byly zmíněny v dotazu! Pozor, například:

Příklady:

U: „Kdy odjíždí“ (?) (ptá se na odjezd!)		
task	request_info	DEPARTURE
U: „V kolik hodin“ (?) (ptá se na čas)		
task	request_info	TIME
O: „Odkud“ (?) (nepíše se tam STATION, není v promluvě)		
task	request_info	FROM
O: „Kam chcete jet“ (?)		
task	request_info	TO

Status report – tento tag odpovídá oznamovací promluvě prezentující informaci, která souvisí jen nepřímo nebo vůbec s hlavním tématem dialogu. Obvykle se takto označuje sdělení charakterizující rozsah poskytovaných informací nebo připravenost řečníka k dialogu, tj. to co není podstatné z hlediska vyřešení hlavního záměru dialogu. Tento tag se tedy proto vztahuje výhradně k *frame* v dimenzi DOMAIN. Odpovídající promluva má pak dimenzi SEMANTICS prázdnou.

Příklad:

Návod k sémantické anotaci NÁDRAŽÍ

O: „no tak tam už moc na výběr nemáte“ (.)
frame status_report ---

U: „moment já si to napíšu“ (.)
frame status_report ---

Speech_repair – má dva významy, v prvním významu označujeme jako speech_repair ty části promluvy, kde došlo k chybě, přerušování a označuje se tím tedy ta část, která byla poté znovu správně řečena. V druhém významu tím označujeme libovolnou část promluvy která je příliš nesrozumitelná, zašuměná nesmyslná ale je dostatečně krátký a dialog za ní plynule navazuje a pokračuje dál. *Speech_repair* má vždy hodnotu DOMAIN rovnu *frame*, bez ohledu na to, jestli obsahuje nějaké informace, které by patřily do *task*.

Příklad:

U: „v sobotu teda vlastně...“
frame speech_repair TIME

U: „...v neděli“ (!)
task present_info TIME

Příklad:

U: ehm já sobota vlastně odjezd do chrr v osm hehe šedesát pět“
task speech_repair

Thanking – označuje promluvu, která reprezentuje poděkování. Takováto promluva má dimenzi SEMANTICS prázdnou. Reprezentuje jak samotné poděkování, tak i zdvořilostní odpověď na něj. *Thanking* má vždy hodnotu DOMAIN rovnu *frame*.

Příklad:

U: „děkuju moc“ (!)
frame thanking ---

O: „není zač“ (!)
frame thanking ---

Verify – tento tag reprezentuje dotazovací promluvu, jejímž smyslem je ověření nějaké konkrétní věcné informace týkající se obsahu dialogu. Ověřuje se jestli v kontextu dialogu platí, či neplatí uvedený údaj. Může to být buď výzva k upřesnění nějakého údaje s návrhem možné varianty, nebo otázka jestli uvedená varianta odpovídá dříve diskutované variantě. Obvykle se na tuto promluvu dá odpovědět kladně, záporně, nejistě („nevím“), příp. potvrdit zopakováním nebo upřesněním návrhu. Pokud je v ověřovacím dotazu uvedeno více možných variant z nichž se má vybrat ta správná, stačí všechny varianty reprezentovat jedním konceptem. Pouze v případě, kdy se opakuje celá úvodní fráze „chcete... nebo chcete raději...“ se tyto varianty anotují odděleně.

Příklad:

O: „a chcete jet pane přes [station]svojšíň/[station] nebo přes [station]nicov/[station]“ (?)
task verify THROUGH (STATION)

O: „může to být eurocity“ (?)
task verify TRAIN_TYPE

U: „to je ta půlnočka“ (?)
task verify TIME

Poznámka:

Dotaz typu: „Kdy chcete jet, dopoledne nebo odpoledne?“ je kombinací **request_info** a **verify**, a musí se tudíž anotovat po částech.

Dotaz typu: „Kdy ráno chcete jet?“ je **request_info**, a ne **verify**.

Verify_neg – tento tag reprezentuje dotazovací promluvu, jejímž smyslem je ověření nějaké konkrétní věcné informace týkající se obsahu dialogu, ale v negativní formě. Oproti výše definovanému **Verify** se používá k označení ověřovacích frází, které vlastně reprezentují významový opak použitého konceptu. To znamená, že sémantický obsah se považuje za negovaný a potvrzuje se vlastně záporom. Díky tomuto záporu nám např. stačí jeden pozitivní koncept NEXT k vyjádření různých forem dotazu na další spoje.

Příklad:

U: „to je poslední že jo / další už asi nejede že jo“ (?)
task verify_neg NEXT

O: „stačí vám to takhle“ (?)
task verify_neg NEXT

Poznámka:

Dotaz typu: „nechcete přímý rychlík?“ je vlastně dotaz v pozitivní formě (má význam „chcete...?“), a naopak třeba „to je poslední?“, kde není žádné „ne“ znamená vlastně: „další už nejsou?“. Takže pozor!

Dimenze – SEMANTICS

V dimenzi SEMANTICS se detailně popisuje významový obsah promluvy pomocí tzv. konceptů. Koncepty mohou být na sebe vázány pomocí závorek, přičemž nejvyšší úroveň (kořen) tvoří hlavní smysl celého segmentu promluvy a nejnižší úroveň (listy, které již za sebou nemají žádné závorkové vazby) tvoří pak dílčí atributy nebo hodnoty tohoto segmentu. Platí zde anotační zásada, že každý použitý koncept by měl mít nějakou typickou slovní reprezentaci nezávisle na kontextu sousedních promluv. Tj. každý koncept by měl vyjadřovat to, co lze pochopit výhradně z textu bez ohledu na průběh dialogu.

Příklad:

U: „prosim vás já bych potřeboval ze [station]sušice[/station] do [station]brna[/station] dnes kolem poledne“
DEPARTURE (FROM (STATION) , TO (STATION) , TIME)

Ze sémantické anotace promluvy z předchozího příkladu můžeme odvodit, že mluvilo o *odjezdu* a specifikovala se *dojezdová stanice*, *cílová stanice* a nakonec *čas*. Hodnotu dimenze SEMANTICS můžeme považovat za překlad do velmi speciálního jazyka, který nám později umožní strojové zpracování. Je dobré si povšimnout, že v hodnotě dimenze SEMANTICS není informace jestli šlo o dotaz nebo o poskytnutí informace. Tato informace je v dimenzi SPEECH-ACT. Na následujícím příkladu vidíme promluvu, která má stejnou hodnotu dimenze SEMANTICS, ale hodnota dimenze SPEECH-ACT je *present_info*.

Příklad:

O: „jede to z [station]sušice[/station] do [station]brna[/station] v dvanáct třicet tři“
DEPARTURE (FROM (STATION) , TO (STATION) , TIME)

Úvod do sémantické anotace

Nejčastěji používáme čtyři hlavní neboli kořenové koncepty: DEPARTURE, DEPARTURE_CONF, ARRIVAL a ARRIVAL_CONF. DEPARTURE je koncept, který reprezentuje otázku nebo odpověď na odjezd nějakého částečně specifikovaného vlaku (odpověď je obvykle konkrétní čas kdy vlak odjíždí z konkrétní stanice). DEPARTURE_CONF je koncept pro promluvu, která ověřuje jestli konkrétní vlak odjíždí v konkrétní čas. Podobně jsou definovány koncepty ARRIVAL, ARRIVAL_CONF. Ty ovšem reprezentují promluvy týkající se příjezdu vlaku.

Každý tento kořenový sémantický koncept může být následován tzv. neterminálními neboli nekoncevými koncepty: FROM, TO nebo THROUGH, které musí být následovány tzv. terminálním neboli koncevým symbolem STATION. Dále se pod kořenovým symbolem nebo i volně mohou vyskytovat ostatní terminální symboly, nejčastěji: TRAIN_TYPE a TIME, případně další, které jsou definovány v tomto manuálu.

Je třeba si uvědomit, že neterminální symboly odpovídají v podstatě označení nějaké vlastnosti nebo akci (např.: DEPARTURE = „odjezd“) a terminální naopak odpovídají vyjádření nějaké konkrétní hodnoty vztahující se k před tím zmíněné vlastnosti (např.: TIME = „ve dvacet hodin“). Toto dělení ale není jednoznačné a některé koncepty mohou být někdy použity jako neterminální a jindy jako terminální, v závislosti na skutečném významu promluvy. Terminální koncepty, které vyjadřují nějakou konkrétní hodnotu (zatím pouze STATION) by měli odpovídat symbolickému označení tzv. *named-entities* v textu (např.: úsek „... [station]sušice[/station] ...“ odpovídá sémantické anotaci: ... STATION ...).

Seznam všech používaných sémantických konceptů

Následující abecední seznam definuje všechny dostupné typy konceptů a jejich použití v příkladech.

ARRIVAL – tento koncept reprezentuje promluvu, ve které se objevuje dotaz nebo odpověď na příjezd vlaku.

Příklad:

U: „abych byla v [station]praze[/station] tak mezi druhou a půl třetí nebo dřív“ (?)
task request_info ARRIVAL (TO (STATION) , TIME)

O: „ten je v [station]praze[/station] na hlavním nádraží jedenáct třicet pět“ (.)
task present_info ARRIVAL (TO (STATION) , TIME)

ACCEPT – tento koncept reprezentuje kladnou odpověď, souhlas, nebo přijetí skutečnosti dotázané nebo oznámené předcházející promluvou typu SPEECH-ACT=verify nebo implicit_confirmation.

Příklad:

O: „ano bez výjimky“ (.)
task acknowledgement ACCEPT

Příklad:

U: „deset padesát dva“ (.)
communication implicit_confirmation TIME

O: „ano“ (.)
communication acknowledgement ACCEPT

AMOUNT – tento koncept označuje konkrétní peněžní částku, která se musí zaplatit například za lístek na vlak. Používá se ve spojení s konceptem PRICE.

Příklad:

U: „a kolik to stojí“ (?)
task request_info PRICE

O: „cena šedesát dva korun“ (.)
task present_info PRICE (AMOUNT)

ARRIVAL_CONF – je koncept pro promluvu, která ověřuje jestli konkrétní vlak přijíždí v konkrétní čas.

Příklad:

U: „přijíždí pořád stejně ten rychlík v devět třicet od [station]Prahy[/station]“ (?)
task request_info ARRIVAL_CONF (TRAIN_TYPE, TIME, FROM (STATION))

AREA – tento koncept popisuje část promluvy upřesňující nějaké místo odkud se volá. Nejčastěji se tedy vyskytuje v zahájení při SPEECH-ACT=opening.

Příklad:

U: „dobrý den [person]hilianová[/person] [area]plzeň[/area]“ (.)
frame opening GREETING, PERSON, AREA

BACK – tento koncept slouží k označení požadavku na zpáteční vlak. Například když uživatel chce spoj z A do B, ale nakonec chce také spoj z B do A. Je to tedy takové nepřímé vyjádření odjezdu s obrácenými parametry FROM a TO (např. dotaz „a zpátky?“).

Příklad:

U: „prosim vás já bych potřeboval vědět vlaky v sobotu co jede z [station]varů[/station] z dolního nádraží do [station]nový_role[/station] ráno někdy“ (?)

task request_info DEPARTURE (TIME, FROM (STATION), TO (STATION), TIME)

U: „a potom v neděli zpátky“ (?)

task request_info DEPARTURE (TIME, BACK)

DELAY – tento koncept reprezentuje dotaz na zpoždění vlaku, nebo prezentaci informace o zpoždění vlaku.

Příklad:

O: „ale teďka mu hlásí zpoždění“ (.)

task present_info DELAY

U: „vo kolik to je“ (?)

task request_info DELAY

DEPARTURE – tento koncept reprezentuje dotaz nebo odpověď na informaci o odjezdu vlaku. Používá se nejčastěji ve spojení s koncepty TRAIN_TYPE, TIME, FROM, TO, apod.

Příklad:

U: „jede nějaký přímý vlak do [station]mnichova[/station] z [station]plzně[/station]“ (?)

task request_info DEPARTURE (TRAIN_TYPE, TO (STATION), FROM (STATION))

O: „tak přímý by vám jel buď ráno v devět třicet čtyři eurocity“ (.)

task present_info DEPARTURE (TRAIN_TYPE, TIME, TRAIN_TYPE)

DEPARTURE_CONF – reprezentuje promluvu, která ověřuje jestli konkrétní vlak odjíždí v konkrétní čas.

Příklad:

U: „prosim vás jede v sobotu ten přímý vlak do [station]horažďovic[/station] v šest dvacet osm“ (?)

task request_info DEPARTURE_CONF (TIME, TRAIN_TYPE, TO (STATION), TIME)

DISCONNECT – tento koncept reprezentuje promluvu, která popisuje rozpojení vlaku na více částí. Samotné části vlaku se popisují konceptem TRAIN_TYPE.

Příklad:

O: „zadní část ta zadní část jede do [station]brna[/station] jo“ (.)

task present_info DEPARTURE (TRAIN_TYPE, TO (STATION))

U: „jinak ten předeek potom odpojujou v [station]budějovicích[/station]“ (.)

task present_info DISCONNECT (TRAIN_TYPE, STATION)

U: „jo zadní část jede do [station]brna[/station] přímý“ (.)

task implicit_confirmation DEPARTURE (TRAIN_TYPE, TO (STATION),

TRAIN_TYPE)

DISTANCE – tento koncept popisuje dotaz nebo odpověď na vzdálenost mezi dvěma vlakovými stanicemi. Používá se spolu s konceptem LENGTH.

Příklad:

U: „já bych potřebovala zjistit kolik kilometrů je to vocat' do [station]kolína[/station]“ (?)
task request_info DISTANCE (FROM (STATION) , TO (STATION))

O: „takže odsud do [station]kolína[/station] sto sedumdesát šest kilometrů“ (.)
task present_info DISTANCE (FROM (STATION) , TO (STATION) ,
LENGTH)

DURATION – tento koncept popisuje dotaz nebo odpověď na dobu trvání cesty mezi dvěma vlakovými stanicemi. Používá se spolu s konceptem TIME.

Příklad:

O: „ale to jako *je tam jede strašně dlouho“ (.)
task present_info DURATION (TIME)

O: „to tam jede hnedle sedm hodin to je hrůza“ (.)
task present_info DURATION (TIME)

FROM – tento koncept označuje zdrojovou stanici. Například stanici ze které se bude odjíždět. Používá se s koncepty DEPARTURE, ARRIVAL, apod.

Příklad:

U: „jede nějaký přímý vlak do [station]mnichova[/station] z [station]plzně[/station]“ (?)
task request_info DEPARTURE (TRAIN_TYPE, TO (STATION) ,
FROM (STATION))

GREETING – tento koncept popisuje promluvu, která má význam pozdravu nebo odpovědi na pozdrav. Vztahuje se výhradně k SPEECH-ACT=*opening*.

Příklad:

O: „informace prosím“ (.)
frame opening GREETING

U: „dobrý den [person]hanzlíková[/person]“ (.)
frame opening GREETING, PERSON

LENGTH – tento koncept reprezentuje hodnotu vzdálenosti mezi dvěma stanicemi. Používá se spolu s konceptem DISTANCE.

Příklad:

U: „já bych potřebovala zjistit kolik kilometrů je to vocat' do [station]kolína[/station]“ (?)
task request_info DISTANCE (FROM (STATION) , TO (STATION))

O: „takže odsud do [station]kolína[/station] sto sedumdesát šest kilometrů“ (.)
task present_info DISTANCE (FROM (STATION) , TO (STATION) ,
LENGTH)

MAYBE – tento koncept reprezentuje nejistou odpověď (ano i ne) na promluvu typu SPEECH-ACT=*verify*

Návod k sémantické anotaci NÁDRAŽÍ

nebo *implicit_confirmation*.

NEXT – reprezentuje promluvu, která požaduje informaci o vlaku který jede později než již dříve sdělený vlak, nebo prostě požadavek na další možnosti. Používá se jen v otázce `request_info` nikdy ne v `present_info`.

Příklad:

U: „a prosím vás něco dalšího“ (?)
task request_info NEXT

NUMBER – reprezentuje nějakou číselnou hodnotu, která není použita k vyjádření času. Používá se obvykle ke specifikaci pořadí vyjmenovaných možností, nebo k označení číselné hodnoty nástupiště.

Příklad:

U: „a ten druhý je taky rychlík“ (?)
task verify REF (NUMBER) , TRAIN_TYPE
O: „ten přijede na druhé nástupiště“ (.)
task present_info ARRIVAL (PLATFORM (NUMBER))

OTHER_INFO – reprezentuje nějakou další významnou vlastnost nebo hodnotu, která nelze zařadit jinak, např. věci na které se uživatel ptá a operátor je sděluje, ovšem neexistuje na ně žádný koncept. Pomocí `OTHER_INFO` lze snadno anotovat komplexní dotaz a odpověď které by jinak anotovat nešly a dialog by se musel označit out-of-topic. Hlavně se jako `OTHER_INFO` označuje hodně jevů, které se dřív označovaly za `status_report`, jako např. počet nalezených spojů. Anotují se takto také například výluky a další mimořádné spoje.

Příklad:

O: „jednou pouze dva spoje“ (.)
task present_info OTHER_INFO
U: „kolik vlastně jede těch vlaků“ (?)
task request_info OTHER_INFO
O: „zadní část jede do brna“ (.)
task present_info OTHER_INFO (TO (STATION))

PERSON – tento koncept popisuje část promluvy, která má význam představení osoby která mluví. Vyskytuje se výhradně vedle pozdravu při `SPEECH-ACT=opening`.

Příklad:

U: „dobrý den [person]hanzliková[/person]“ (.)
frame opening GREETING, PERSON

PRICE – tento koncept popisuje dotaz nebo odpověď na informaci kolem ceny, slevy nebo příplatku. Koncept se často používá ve spojení s konceptem `AMOUNT`. Pozor jako `AMOUNT` anotovat i vágní formu jako například „normální“ cena, „běžná“, „levná“ cena apod.

Příklad:

U: „a kolipa to stojí ještě“ (?)
task request_info PRICE
U: „tam to máte za osumdesát šest korun“ (.)

Návod k sémantické anotaci NÁDRAŽÍ

task present_info PRICE (AMOUNT)

PREVIOUS – reprezentuje promluvu, která požaduje informaci o vlaku který jede dříve než již dříve sdělený vlak. Používá se jen v otázce request_info nikdy ne v present_info.

Příklad:

U: „a něco dřív“ (?)

task request_info PREVIOUS

PLATFORM – tento koncept reprezentuje pojem nástupiště nebo místo odjezdu na konkrétním nádraží. Používá se společně s konceptem DEPARTURE a ARRIVAL.

Příklad:

U: „z kterýhopak nástupiště“ (?)

task request_info DEPARTURE (PLATFORM)

O: „ten nám odjíždí třetí východ“ (.)

task present_info DEPARTURE (PLATFORM (NUMBER))

U: „třetí východ“ (.)

communication implicit_confirmation PLATFORM (NUMBER)

REF – reprezentuje odkaz na nějaký dříve zmíněný objekt v dialogu, který je určen hodnotou jeho atributu.

Příklad:

U: „a kdy jede ten rychlík co jste říkala jako první“ (?)

task request_info DEPARTURE (REF (TRAIN_TYPE, NUMBER))

U: „a ten v těch šest je tam v kolik“ (?)

task request_info ARRIVAL (REF (TIME))

REJECT – reprezentuje promluvu, ve které se objeví zamítnutí nějaké informace. Obvykle se jedná o odpověď na SPEECH-ACT=verify. Koncept REJECT je velmi často spojen s koncepty TIME, TRAIN_TYPE, STATION, FROM, TO, apod.

Příklad:

O: „ne ne ten už jel osm nula pět“ (.)

task acknowledgement REJECT (TIME)

REPEAT – tento koncept reprezentuje promluvu, která vybízí druhou stranu k opakování informace protože ji nebylo slyšet nebo nebyla pochopena.

Příklad:

O: „a bude to ještě teďko někdy jo“ (?)

task verify TIME

U: „prosím“ (?)

communication apology REPEAT

STATION – tento koncept slouží jako identifikace vlakové stanice. Reprezentuje nejen vlastní názvy měst a obcí, ale celé označení stanic nebo zastávek, a to i v hovorovém tvaru (např. „hlavák“, „plzeň zastávka“, apod.).

Příklad:

U: „já bych potřebovala zjistit kolik kilometrů je to vocat' do [station]kolína/[station]“ (?)
task request_info DISTANCE (FROM (STATION) , TO (STATION))

SYSTEM_FEATURE – tento koncept slouží k označení promluvy kdy si uživatel chce ověřit zda systém (v našem operátor) může poskytovat nějaké informace nebo služby. Například fráze „jsou to informace?“.

Příklad:

U: „prosím vás dáváte taky na mezinárodní spoje informace“ (?)
task verify SYSTEM_FEATURE

O: „no co potřebujete“ (.)
task acknowledgement ACCEPT

TIME – tento koncept slouží k označení data nebo času v promluvách. Používá se buď samostatně nebo ve spojení s koncepty DEPARTURE, ARRIVAL, DURATION, apod.

Příklad:

U: „dvacet dva nula šest“ (.)
communication implicit_confirmation TIME

O: „který den že to bylo prosím“ (?)
task request_info TIME

U: „v neděli ted'ka“ (.)
task present_info TIME

O: „odpoledne pane ale v tudlenctu dobu přichází v úvahu vlak intercity nevadí“ (?)
task verify TIME, TRAIN_TYPE

TO – tento koncept označuje především cílovou stanicí reprezentovanou předložkami „do“ nebo „na“. Jeho parametrem je koncept STATION. Tento koncept lze také použít ve smyslu „ve směru“ nebo „staví v“.

Příklad:

O: „ano v [station]praze/[station] na hlavním nádraží budete v šestnáct čtyřicet šest“ (.)
task present_info ARRIVAL (TO (STATION) , TIME)

U: „já nevím asi na hlavní“ (.)
task present_info TO (STATION)

U: „a ten vlak na [station]prahu/[station] staví v [station]berouně/[station]“ (?)
task verify REF (TO (STATION)) , TO (STATION)

U: „a ten vlak do [station]klatov/[station] jede směrem na [station]budějice/[station]“ (?)
task verify REF (TO (STATION)) , TO (STATION)

TRAIN_TYPE – tento koncept označuje nějakou speciální vlastnost vlakového spoje. Jestli to je osobní, spěšný vlak nebo rychlík, apod. Jestli to je vlak s příplatkem, umožňující převoz kol a kočárků apod. Vlak také může být přímý. Ovšem pro ostatní informace, či vlaky s přestupem nutno řádně použít koncept TRANSFER.

Příklad:

U: „prosím vás bude přes svátky jezdit nějaká parnička“ (?)

task request_info DEPARTURE (TIME, TRAIN_TYPE)

O: „no v patnáct šestnáct přímo“ (.)

task present_info TIME, TRAIN_TYPE

U: „osumnáct padesát tři normální rychlík“ (.)

communication implicit_confirmation TIME, TRAIN_TYPE

O: „tak pokud chcete paní přímý tak vám jede v devět nula nula“ (.)

task present_info DEPARTURE (TRAIN_TYPE, TIME)

O: „včetně příplatku jo“ (.)

task verify TRAIN_TYPE

TRANSFER – tento koncept označuje přestup z jednoho nádraží nebo vlaku na druhé nádraží nebo vlak. Bývá napojen na koncepty FROM, TO, ale nemusí být napojen na nic „je tam přestup“ - TRANSFER.

Příklad:

O: „takže paní tam přechazíte na [station]prahu[/station] masarykovo“ (.)

task present_info TRANSFER (TO (STATION))

Příklad:

O: „no ale to je s přestupem v [station]praze[/station]“ (.)

task present_info TRANSFER (STATION)

THROUGH – tento koncept označuje stanici přes kterou má vlak projíždět. Používá se buď samostatně, nebo jako parametr nadřazených konceptů DEPARTURE, ARRIVAL.

Příklad:

O: „a chcete jet pane přes [station]svojšín[/station]“ (?)

task verify THROUGH (STATION)

U: „ano přes [station]svojšín[/station]“ (.)

task acknowledgement ACCEPT (THROUGH (STATION))

WAIT – tento koncept reprezentuje dotaz, nebo odpověď na fakt jestli daný vlakový spoj čeká na nějaké stanici po nějakou dobu. Parametry konceptu *WAIT* mohou být *STATION*, *TIME*.

Příklad:

U: „čeká to tam“ (?)

task request_info WAIT

O: „tam je to hned na přestup no čtvrthodinka“ (.)

task present_info WAIT (TIME)

WHAT TIME – reprezentuje vlastnost „čas“, která není blíže určena jestli jde o odjezd, nebo příjezd. Označuje se tím obvykle dotaz na hodnotu obecného času, tedy fráze obsahující („kdy, v kolik“) a ne fráze, která obsahuje hodnotu času („ráno, v šest“). Používá se pouze v request_info v případě nerozlišení DEPARTURE/ARRIVAL.

Příklad:

O: „a kdy přesně“ (?)

task request_info WHAT_TIME

O: „a v kolik jako ráno“ (?)

task request_info WHAT_TIME (TIME)

Doplňující poznámky k anotaci

Dělení promluv na dialogové akty

V zásadě méně dělení je lépe, protože se potom uchovává stromová struktura promluvy (strukturální informace). Vyplatí se používat pokud možno nadřazené koncepty DEPARTURE, ARRIVAL, apod.

Příklad:

„takže pane v [station]čičenicích[/station] byste byl dvacet třicet pět už“

Se nedělí na dva dialogové akty:

„takže pane v [station]čičenicích[/station]“

task present_info TO (STATION)

„byste byl dvacet třicet pět už“

task present_info TIME

Ale označí se dohromady jako:

task present_info ARRIVAL (TO (STATION) , TIME)

Příklad:

„jo dvacet třicet pět výborně“

Se nedělí na dva dialogové akty:

„jo dvacet třicet pět“

communication implicit_confirmation TIME

„výborně“

task acknowledgement ACCEPT

Ale označí se dohromady jako:

task acknowledgement ACCEPT (TIME)

Opakovaná informace na začátku věty je implicit_confirmation

Příklad:

U: „a v sobotu ráno nák“ (?)

O: „v sobotu ráno jede v šest šestnáct“ (.)

Promluvu operátora dělíme na dva dialogové akty:

„v sobotu ráno“

task implicit_confirmation TIME

„jede v šest šestnáct“

task present_info TIME

Implicit_confirmation neposouvá dialog dále ve zpracování, pouze ověří jestli si dobře rozumí účastníci hovoru. Proto SPEECH-ACT=*implicit_confirmation* patří do DOMAIN=*communication*.

Kontextová nezávislost sémantické anotace

Sémantická anotace by měla být kontextově nezávislá, to znamená, že by měla vypovídat o významu každé promluvy pouze na základě slovního vyjádření bez ohledu na kontext okolních promluv. Kontext a pragmatické znalosti v rámci promluvy ovšem je nutno uvažovat a je velmi žádoucí toto do anotace zahrnout. Předpokládá se, že se vše týká vlakového spojení a v tomto rámci je i sémantika definována.

Příklad:

„ten máte v [station]čičenicích[/station] v sedm čtyřicet devět“

task present_info ARRIVAL (TO (STATION) , TIME)

„a nebo osm nula nula“

task present_info TIME

„a ten je tam devět čtyřicet“

task present_info ARRIVAL (TIME)

Closing a thanking

Je třeba dodržovat dělení dialogových aktů na closing a thanking.

Anotace příjezdového času v dotazu na odjezd vlaku

Následující promluva:

„a ze [station]žatce[/station] vám odjíždí přípoj no dvanáct nula tři a jste v [station]chomutově[/station] dvanáct třicet sedm“

lze anotovat v dimenzi SEMANTICS následovně:

DEPARTURE (FROM (STATION) , TIME , TO (STATION) , ARRIVAL (TIME))

ale správné anotování je rozdělit promluvu na dva samostatné dialogové akty DEPARTURE a ARRIVAL, tj.:

DEPARTURE (FROM (STATION) , TIME)

ARRIVAL (TO (STATION) , TIME)

Anotace rozprostřeného času

Když se čas nebo datum vyskytuje na více místech promluvy, potom se anotuje vícekrát. Můžeme použít více konceptů TIME v sémantické anotaci. Pokud je ale vyjádření souvislé, použije se jen jednou. Koncept TIME tedy může reprezentovat jak konkrétní čas, interval, tak i seznam možností.

Příklad:

„já mám prosbu jakpak jedou dneska osobní vlaky nák dopoledne do [station]starýho_plzence[/station]“

DEPARTURE (TIME , TRAIN_TYPE , TIME , TO (STATION))

Příklad:

„dopoledne můžu vám nabídnout pane v osm devatenáct nebo deset třicet“

DEPARTURE (TIME, TIME)

Je zajímavé si všimnout, že slova vážící se ke konceptu DEPARTURE se objevují třeba uprostřed věty.

Používání Dialogue Act Editor

Dialogue Act Editor (DAE) je program na provádění sémantické anotace Nádraží.

Stavy zpracování dialogů

Zpracování každého dialogu prochází následujícími stavy: *unannotated*, *out-of-topic*, *partially-annotated*, *completely-annotated*, *approved*. Počáteční stav je vždy *unannotated*. Koncové stavy jsou *out-of-topic* a *approved*.

Příklad anotačních stavů

1. dialog není anotovaný: *unannotated*
2. začínáte anotovat, označíte si dialog jako *partially-annotated* a do poznámky si zapíšete poznámku – proč dialog není hotový, případně co zajímavého na co jste se chtěli zeptat v dialogu je. Zajímavosti psát pouze ZŘÍDKA – u opravdu zajímavých jevů. Psát hlavně dotazy. Pokaždé se snažit napsat číslo promluvy ke kterému se poznámka vztahuje. Pokud mám k dialogu nějaký dotaz, nechám si dialog označený jako částečně o anotovaný a neoznačuji ho jako anotovaný!!!
3. narazíte-li v dialogu na jev, který URČITĚ nelze anotovat, tj. netýká se úlohy vlakových spojů (poštovní zásilky, telefonní čísla) případně je velmi složitý a nelze reprezentovat koncepty (průkazky, slevové karty apod) dokončí se anotace těch částí dialogu které anotovat lze a dialog se označí *out-of-topic* a do poznámky se napíše hlavní důvod proč dialog je out-of-topic.
4. Pokud mám dialog celý o anotovaný a nemám k němu žádné připomínky ani dotazy, označím dialog jako *completely-annotated* a v poznámce nechám v ojedinělém případech co zajímavého v dialogu bylo. Ve většině dialogů v tuto chvíli poznámku vymažu – do poznámky se píše hlavně dotazy a nejasnosti u *partially-annotated* dialogů.
5. Jako *approved* se označují dialogy, které kontrolujete po někom jiném. Každý takový „completely-annotated“ dialog projdu a pokud ho shledám správný, označím ho jako *approved*. Pokud ho neshledám správný, opravím chyby a nechám jej jako *completely-annotated*. U dialogů *out-of-topic* provedu to samé – zkontroluji a případně opravím chyby.

Detailní popis stavů

Unannotated – toto je prvotní stav. V tom to stavu jsou všechny nezpracované dialogy.

Out-of-topic – tento stav se použije pro dialog, který není možno sémanticky anotovat protože samotná sémantika neobsahuje potřebné koncepty pro její zpracování. Například se jedná o hovor kdy si volající spletl telefonní číslo, nebo se ptá na dopravu. Musíme mít na paměti, že dialogy by se měli týkat dotazů na vlakové spoje.

V současnosti se anotují dialogy out-of-topic, i když se dotaz týká slev na cenu jízdenky a s tím spojené informace vztahující se k různým průkazkám a slevovým kartám. Anotují se pouze jednoduché dotazy na cenu.

Partially-annotated - takto označený dialog má některé promluvy již sémanticky anotovány a pracuje se na něm, případně je celý ale je nutné se na nějaké jevy zeptat.

Completely-annotated – takto označený dialog má již všechny promluvy anotovány a je hotový.

Approved - takto označený dialog byl již zkontrolován a odsouhlasen nadřazeným anotátorem.

Odevzdání práce – jednoho anotačního balíku

Práce se odevzdává ve stavu, že jsou všechny dialogy anotovány jako *completely-annotated* nebo *out-of-topic*. Při kontrole navíc některé dialogy budou *approved*. Po odevzdání, se dialogy zazálohují a do tabulky na CVS se zapíše datum odevzdání, jméno atd.

Překlepy

Zkontrolujte si prosím občas překlepy – už se snad nestává, že by někdo koncepty psal ručně (ve velkém), doufám že všichni používáte zkratky Alt+písmeno. Ojediněle jsem viděl drobné chyby ve smyslu překlepu při používání zkratk: dvakrát místo ACCEPT – ARRIVAL a jednou místo REJECT – REPEAT. U klávesy T to patrně je složitější, tam je jako překlep místo TIME koncept TRAIN_TYPE – což už se velice snadno přehlédne takže nevím, zda se to také někdy nestane. Prosím dávejte na tyto překlepy také pozor, zvlášť pokud se snažíte „pracovat rychle“.

Změny dokumentu

- 6.9.2005 - Přidány koncepty DELAY, DISCONNECT, DURATION, WAIT.
- Odstraněny koncepty TRAIN, IN_DIRECTION.
- 20.9.2005 - Přidány koncepty REF, NUMBER, WHAT_TIME v kapitole SEMANTICS
- Upřesněno použití a přidány příklady v kapitole SPEECH_ACT
- Předefinován VERIFY a VERIFY_NEG z konceptu na speech_act
- 26.9.2005
- anotace stavů dialogu
 - dopsán speech_repair
 - request_info (TIME/FROM/TO)
 - other_info
 - offer
- 27.9.2005 sjednocení grafické úpravy a oprava některých chyb
- 24.11.2005 přidány časté chyby anotátorů