The Implementation of a ReALIS-based Method of Static Intensional Interpretation

Márton Károly[1] and Gábor Alberti[2]
[1]"Science, Please!" Project Office, University of Pécs, 4 Vasvári Pál Street, Pécs, Hungary
[2]Department of Linguistics, University of Pécs, Pécs, Hungary

Keywords: Extensional and Intensional Predicates, Static and Dynamic Interpretation, Modeling Human Interactions.

Abstract: This paper is grounded in the joint work within our project to implement a static (and, later, dynamic) interpretation of sentences, using the theory of ℜeALIS. It outlines definitions and implementations of truth-evaluating extensional and intensional predicates, with an aim to present the Prolog-based core (i.e. the future server-side) of the program. This program has been partially demonstrated in recent publications; it is now mature enough to have a (presumably Java/JSP-based) graphical interface deployed to it. In this paper we still use a bottom-up approach because the theoretical complexity of ℜeALIS does not concur with a too early finalization of the architecture. Most notably, the world model data in ℜeALIS, equipped with the semantic postulates, share a great many features with the program code itself. This causes problems when planning the user interface. As for future work, the most important goals are, on the one hand, to implement verbal semantics (while sticking to the bottom-up approach) and, on the other hand, to finalize the architecture, the use-cases, the components and the deployment (switching to a top-down approach).

1 INTRODUCTION

ℜeALIS is a post-Montagovian theory which concerns the formal interpretation of sentences constituting coherent discourses (Asher and Lascarides, 2003) and which uses an extension of Kampian DRT (Kamp et al., 2011). As stated in Alberti and Károly, 2010, ℜeALIS provides a theoretical background to represent the contents of the human interpreters' minds, including their BDI (belief, desire and intention, see Vadász et al., 2013), background knowledge and pragmatic relations – beyond the actual contents and structure of the discourse. ℜeALIS also overcomes the problem of representationalism (see Alberti and Károly, 2011) by embedding the interpreters themselves into the world model, along with their mental states.

It has become clear over the past few months that a pure NL-based input should be replaced by an extendable (and replaceable) language model, based on a context-free grammar. Using this, it is possible to build a ℜeALIS-based model generator by which the λ-labels of worldlets (definition: Alberti, 2011, pp. 139–177, see also Alberti and Károly, 2012) are assigned to linguistic elements.

This paper describes the problem of switching between extensionality and intensionality (in ℜeALIS, $\mathbf{W_0}$ and W[i,t], see Alberti and Károly, 2010), and it also offers a partial solution to it: predicates can be grouped by how they are related to the outer world $\mathbf{W_0}$ and to the internal W[i,t] worldlets of the interpreters. Predicate types can be regarded as semantic postulates which are, in fact, defined in the **oracle.** Unlike any represented human interpreter, the oracle has no BDI; but it still contains semantic postulates and logical rules in non-BDI worldlets, in order to define intensional predicates). Code fragments of actual and possible implementations of certain predicate types are shown in Sections 2 and 3 where we use a greatly simplified approach to NP-anchoring which is, indeed, at least in theory, refined in Section 4.

The planned architecture of the ℜeALIS software is summarized in the last section, while possible applications of ℜeALIS in NLP-based expert systems were shown in Alberti and Károly, 2010. Most importantly, ℜeALIS-like theories and software based on them could be used in the jurisdiction: by investigators, judges, prosecutors and barristers.

Károly M. and Alberti G..
The Implementation of a ReALIS-based Method of Static Intensional Interpretation.
DOI: 10.5220/0004627503930398
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2013), pages 393-398
ISBN: 978-989-8565-81-5

2 LINGUISTIC AND MODEL (LEXICAL AND SEMANTIC) GENERATION

In its current stage of development, the linguistic generation is CFG-based so that we can overcome the difficulties of linguistic acceptance. The Prolog facts and data structures look much like the ones sketched in Alberti and Károly, 2011. Linguistic input is provided by applying the appropriate rules of the CFG. Possible λ- (worldlet) labels are assigned to certain modal lexemes and morphemes, or, in the case demonstrated here, phrases like *D thinks (S)*:

Example 1: Linguistic generation (see CFG below) and truth evaluation of the **extensional** predicate *to be married*. The presence of the Prolog facts below (see code fragments) is checked in the Prolog factual database.

S' → S | D think(s) S' | according to D S' | …
S → D am/are/is married | D am/are/is pretty | …
D → I | you | Mary | Joe | Peter | the/a boy | …
(the rules are triggered by the interface)

EGO (entity no.1.) talks to YOU (entity no.3.),
NOW (20130515 as date), HERE (Budapest) are all set by the user.
Sentence: ***Peter is married.***

```
referent(102,ref,r,'Peter').
lambda(102,ref,1,[]).
%root worldlet (label []) of EGO
%EGO knows a certain Peter
alpha(102,ref,2,ent,out,gest).
%referent 102 is out-anchored to
%entity 2 by gesture
external(2,ext,ent,['Peter
Smith',i,19790201,20601010]).
%external/4: ID,two type args,
%parameter list depending on type
%(name,i=interpreter,birth,death)
lambda(302,ref,3,[]).
%ref. 302 in the root worldlet of YOU
alpha(302,ref,2,ent,out,gest).
%YOU also knows 'Peter' whose referent
%is anchored to the same entity

referent(115,ref,p,'married').
lambda(115,ref,1,[]).
%'married' is an extensional predicate
%it has an external equivalent to
%which it is alpha-(out-)anchored:
alpha(115,ref,5,crel,out,gest).
external(5,ext,crel,'married').
%extensional predicates are anchored
%to core relations (crel)
external(5,ext,inf,[[5,crel],[20101121,
20140515],[2,ent]]). %arg=[2,ent]
%parameter list of infons: core
%relation,[time interval], 1 or more
%entities as arguments]
%inf=infon (SSIS only, see Seligman–Moss, 1997):
%YES, Peter is married. Without the
%last external fact (cf. if the last
%check failed), the answer would be NO
```

Examples 1 and 2 also demonstrate how anchoring is checked before truth evaluation (see Section 4). As noted above, the values of the κ cursor function in ℜeALIS (Alberti and Károly, 2011) i.e. speaker (I/EGO), addressee (YOU/Hearer), time (NOW) and location (HERE) – must be preset before the actual analysis. This is necessary not only for the adequate handling of pronouns and/or interpersonal knowledge, but also to uniquely anchor the referents denoted by the topic of the sentence (here: *Peter*) – since the name *Peter* is not unique on its own. (The name *Peter Smith* is considered unique.) This means that, in real life, the hearer could ask the question: *Which Peter?* Taking all this into account, semantic analysis has two phases: the **anchoring** phase, in which pre-suppositions are checked (if any of these checks fail, the sentence is neither true nor false, e.g. *The reigning French king is bald*), and the **actual truth evaluation.** Anchoring always starts from the actual λ-level of the discourse (here, it is the root worldlet). Referents of other worldlets can also be used (obeying the "up or left" rule of Kampian DRT), but the truth values of the predicates must be checked from the viewpoint of the outer world, stepping into λ-worldlets if needed. In order to achieve this, it is best to create a special interpreter – an oracle. If a certain λ-worldlet does not exist, its creation is simply done by asserting new `lambda` predicates).

Example 2: *Think* is an intensional verb, it turns off the extensional checking of *Peter is married.* In this case, only the presence of an internal eventual referent is checked (and not that of an infon).

Sentence: ***Joe thinks Peter is married.***

```
%Joe is entity no. 4.
referent(104,ref,r,'Joe').
lambda(104,ref,1,[]).
%root worldlet (label []) of EGO
%EGO knows a certain Joe
alpha(104,ref,2,ent,out,gest).
external(4,ext,ent,['Joseph
Taylor'],i,19770311,20591112]).
lambda(304,ref,4,[]).
%root worldlet of YOU
alpha(304,ref,4,ent,out,gest).
%YOU also knows 'Joe'
referent(102,ref,r,'Peter').
lambda(102,ref,1,[]).
```

```
%root worldlet (label []) of EGO
%EGO knows a certain Peter
alpha(102,ref,2,ent,out,gest).
external(2,ext,ent,['Peter
Smith'],i,19790201,20601010]).
lambda(302,ref,3,[]).
%root worldlet of YOU
alpha(302,ref,2,ent,out,gest).
%YOU also knows 'Peter'
%Joe, too, has to know the same Peter
referent(402,ref,r,'Peter').
lambda(402,ref,4,[]).
%root worldlet (label []) of Joe
%EGO knows a certain Peter
alpha(402,ref,2,ent,out,gest).
external(2,ext,ent,['Peter
Smith'],i,19790201,20601010]).
referent(998,ref,e,[[410,ref,p,
'think'],20130515,[104,ref,r,'Joe'],
[999,ref,e,null]]). (…)
referent(999,ref,e,[[415,ref,p,
'married'],20130515,[404,ref,r,'Peter'
]]). %eventual referent
lambda(999,ref,4,[[bel,med,20130515,+]
]). %lambda-equivalent of to think
%YES, Joe thinks Peter is married.
```

It is very important to note that in Example 1, the eventuality of *Peter is married* is **not** necessarily present in the mind of the speaker or the hearer, or at least not with a positive polarity (**referent**(…,ref,e,…) is not listed because it is not checked at all). The description of lies, bluffs, fibbing or killing the joke equally involve a change in the polarities of the worldlets to negative, zero or even 'non-zero' (e.g. for yes/no questions, see Kilián, 2013 and Vadász et al., 2013). The hearer also has "free choice" to believe or to not believe the sentence: (s)he can move, or more precisely, **accommodate** the eventual referent to **any** of his/her internal worldlets of belief.

The speaker, too, can have the eventual referent in his mind and he can refer to it: *Peter is married and this is what makes Mary sad.* But the question of having or not having the eventual referent in one's mind does not necessarily depend on the truth value of *Peter is married.* In the Middle Ages, the fact that many people bore in their minds the eventuality of *Susan is a witch* often caused dire consequences…

The introduction of *Joe thinks* results in beclouding the actual truth value of *Peter is married.* Private intensional predicates like *think, believe, desire, want, lap up* as well as most modal verbs do not have external equivalents so **no** extensional check is performed (apart from checking the uniqueness of *Peter* and *Joe* in the given context).

Some possible types of predicates are listed in Table 1: they are assigned to modal particles, intensional verbs, adjectives etc. The semantic postulates by which these types are defined in the oracle and which roughly correspond to the "code segment" of a truth-evaluating program (functioning in a different way for each type, after checking the predicates – in a similar way as in Examples 1 and 2) no longer belong the linguistic generator but to the model generator. The CFG (the linguistic generator) is used as input, and this invokes the model generator to determine truth value (or to not determine it in case of anchoring problems – e.g. lack of uniqueness, anchoring mismatch, or falsity of the presupposition).

In the near future, the fact database will be extended to a real model generator in which the former database will actually operate as a "data segment": **internal users** (administrators, linguists) will be able to load the world model, including the interpreters' simulated brains with data – by using the (web) interface.

Summarizing the above, the program is constituted of two parts connected to each other:

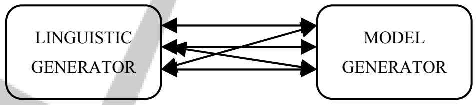

Figure 1: The linguistic generator and the model generator – corresponding to the linguistic form and the meaning of words.

Polysemy and synonymy are depicted by the crossing arrows in Figure 1: if we exploit the backtracking features of Prolog, it can result in multiple solutions and/or the same solution for two or more different inputs.

3 POPULATING THE LINGUISTIC DATABASE. PREDICATE TYPES

Linguistic elements are inserted into the CFG as possibilities. Nouns, adjectives and verbs are inserted one by one or in groups into the grammar by internal users, applying `assert` on the Prolog server. (The CFG should be extended to an indexed grammar which is more suitable for morphologically rich languages like Hungarian or Turkish.)

The real problem is to determine the types of the certain predicates (adjectives and verbs) and to **invoke the model generator** to extend the world model. Extensional predicates have external equivalents in the (Montagovian, see Dowty et al., 1980) outer world, which causes their semantic postulate to establish at least one α-anchoring into the outer world (i.e. a referent is anchored to **an entity** and not to another referent) and also to include a definition of a core relation in it (see definitions in Alberti and Károly, 2010, 2011 and the full description in Alberti, 2011). Intensional predicates, however, have no external equivalents. Instead, their semantic postulates include creating and/or checking certain λ- (modal) labels or more complex intensional structures – such as checking the mental states of more than one interpreter. One such example is *be pretty* – see Table 1. The heads of the checking predicates are mostly omitted because they are taken out of context.

Table 1: Predicate types and their definitions / truth value evaluations.

Predicate type, definitions/checks	Examples
<u>Extensional:</u> $e_{\mathbf{married}}$: $p_{married}$ t r_1 core relation is defined in the outer world **`external`**`(RID,ext,`**`crel`**`,NAME)`, with infons: **`external`**`(ID,ext,`**`inf`**`,` `[[RID,crel], [TIME_BEGIN,` `TIME_END]|ARGUMENTS])` etc.	*married* *Ukrainian (as citizen)* *to swim* anchorings: p → c. rel. e → infon

<u>Private intensional:</u> $e_{\mathbf{believe}}$: $p_{believe}$ t r_1 e_2
no core relation, one or more λ-relations in the mind of a certain interpreter
`referent``(EBL,ref,e,[[BID,ref`
`,p,'believe'],TIME,R1,E2])`
`lambda``(EBL,ref,`
`INTERPRETER_ID,[[bel,`
`STRENGTH,TIME,POLARITY]]),`
only the presence of a certain eventuality `EID` (identifier of e') is checked
`referent``(E2,ref,e,`
`[[PREDID,ref,p,PRNAME],TIME`
`| ARGUMENT_REFERENTS])` etc.
argument referents are inferred from the results of anchoring checks
	to believe e'
	to think e'
	to desire e'
	to search
	(for r)
	e: eventual referent (described by subordinate clause or similar)

<u>Quantified intensional:</u> $e_{\mathbf{pretty}}$: p_{pretty} t r_1 No external equivalents. Example: *Mary is pretty.* After a check for the uniqueness of *Mary* (e.g. 'Mary Johnson'; let it be `IENT`), anchoring to `IENT` is checked for every interpreter. Then we take the set `IIDLIST` of those who anchor a referent **to the same** `IENT` entity (know her). If, for example, more than 2/3 of `IIDLIST` think (s)he is *pretty* (and have *pretty* as a predicate name, this means they know what *pretty* means), the predicate is considered to be true. **`quantcheck`**`(PRNAME,IENT):-`	*beautiful* *pretty* *ugly* (`PRNAME`) neither e nor p has an external equivalent

Table 1: Predicate types and their definitions / truth value evaluations (cont).

`setof``(ENTID,`**`quantintev`**`(IENT,`
`PRNAME,ENTID),PILIST),`
(those who think `IENT` is `PRNAME`)
`findall``(IID,(`**`alpha`**`(REFID,ref`
`,ENTID,ent,out,_),`**`lambda`**`(REF`
`ID,ref,IID,[])),IIDLIST)`(…)
(those who anchor some referent to `IENT` – not necessarily with the same name)
Then, the length of `PILIST` is divided by the length of `IIDLIST`.
The details of **`quantcheck`** are omitted.	

<u>Mixed intensional:</u> $e_{\mathbf{bald}}$: p_{bald} t r_1 *Peter is bald:* Same as quantified intensional but **with** an external representation of the predicate as **prototype**. The eventuality may or may not have an external equivalent (non-prototypical case). Not only 'Peter' but the predicate 'bald', too, must be unique and common. The eventuality of 'Peter is bald' may or may not be `alpha`-anchored to an infon. In the second case, both p and r_1 have to be anchored correctly to grant the interpreter the "right to vote".	*bald* p → c. rel. is e anchored to an infon? yes → extensional no → quantified intensional (with voting)

<u>Intensional naming:</u>
see anchoring of *Peter* and *Joe* in Examples 1-2. Similar to pronouns.
`alpha``(ID,ref,TO_WHAT_ID,ent/`
`ref,arg/pred/ant/adj/out,`
`cat/ord/agr/hum/gest etc.)`
is checked, see also Section 4.
	nicknames
	r_{Peter} → 'Peter Smith' etc.

Arguments written in lower-case letters like `ref`, `ent` etc. are type and subtype names for referents, entities, linguistic (`cat`egory, `ord`er etc.) and extra-linguistic (`gest`ure) anchoring categories.

3.1 The "Already given" Extensional Predicates

The seemingly simplest predicate type consists of a predicate referent which is located in the root worldlet and is **`alpha`**`(ID,ref,CRELID,crel,`**<u>`out`</u>**`,_)`-anchored to a **strictly homogeneous** core relation in ℜeALIS (this is called **<u>out</u>**-anchoring), but where, unlike in many ontologies, the arity of core relations and predicates is not limited.

All entities of the outer world are defined with **`external`** predicates (cf. **`referent`** in the internal world[let]s): entities, core relations, infons, and, later, time intervals.

Core relations consist of **infons** of Seligman and Moss (1997) but only simple infons are permitted. They also contain discrete time intervals. **Each** element of a core relation is described by one or more infons, depending on time. The last, list argument of an `external` predicate describing an infon always has two further elements in addition to the arity of the core relation to which the infon belongs. These predicates are "already given" because their existence depends neither on the language, nor on the interpreters (e.g. to be *married* is strictly defined by the law).

3.2 Extensional Predicates in an Intensional Domain

Let the interpreter be i. "Intensional domain" means that referents are in any of the fictive worldlets of i, so to say, not in the outer (real) world. Infons turn into **eventual** referents when they are mentally depicted in any of the fictive worldlets. In theory, unlike infons, the structure of eventual referents (e) can be arbitrarily complex because the arguments of intensional and modal verbs/particles (see Table 1) are often eventualities themselves (e'). But because this also implies a λ-level switch from w to a new w' worldlet, e' only has to be explicated in w': a subordinate clause r_1 *believes* e', more formally $e_{believe}$: $p_{believe}(t,r_1,e')$ means that, apart from the presence of $e_{believe}$ in w, only the presence of the eventual referent e' (with the appropriate contents) should be checked in i's w' worldlet of belief (`bel` is present in its λ-label with positive polarity). Being an argument of e, the internal structure of e' is omitted in w (see `null` in referent 999, Example 2). For non-positive polarities see Vadász et al., 2013, e.g. the semantic postulate of the verb *lie* only contains a simple polarity check.

3.3 Predicates with a Purely Intensional Definition. Quantified Intensionality

Similar to modal and intensional verbs and adjectives, *pretty* has no external equivalent because the actual meaning of 'prettiness' greatly depends on who is said to be *pretty* by whom.

Indeed, the truth evaluation of *pretty* may be done by simply saying "pretty are those who are said to be pretty by most people". This can be used as a semantic postulate, making a predicate type on its own. Even this type is already implemented with relatively small simplifications.

Let us truth-evaluate *Mary is pretty.* The minds of all interpreters in the universe of ℜeALIS are searched for the predicate name *pretty*. Then, the entity which the referent named 'Mary' refers to (in the speaker's mind or in the common knowledge of the speaker and the hearer, see Section 4) is taken and all referents referring to it are collected, along with the interpreters themselves. After that, "thinking that Mary is pretty" – a **belief,** see Table 1 – is checked in all interpreters with such referents.

Then, the number of interpreters who think Mary is pretty is divided by the number of interpreters who **actually know the same** Mary as the one (unique!) present in the common knowledge of the speaker and the hearer.

3.4 Hidden or Mixed Intensionality

The difference between *bald* and *pretty* is that the former actually has a **prototype** (when somebody has no hair) – this counts as an external equivalent. As such, a core relation (Z) should be defined to it which may even be empty if we assume that "prototypical baldness" is very rare. Z is just a common ancestor of **all** predicate referents of 'bald' in **any** interpreter. Even those should know the prototypical meaning of 'bald' who use it for people who actually have some hair. Only these interpreters should have the "right to vote" in the way described in 3.3. Truth evaluation of *Peter is bald* is a combination of 3.1 and 3.3: if someone is bald according to Z, it is true. If not, interpreters who know "prototypical baldness" and (a certain) Peter at the same time, "vote" for or against the truth.

There are many more predicate types which are yet to be implemented: the most complex verbs have **five** phases (preparatory phase, starting point, cumulative phase, cumulative point, result phase) with five different semantic postulates. For example, both the preparatory and the result phase of the VP *to fly home* lasts at least some hours. In the preparatory phase, everything is intensional: the eventual referent is in the worldlet $\langle int_{max}, i, \tau, + \rangle$. Moreover, i is most likely to be preparing for his/her journey: buying the tickets, packing, and the like.

One assumes that, in the result phase, i *is probably at home*. But this probability decreases over time: the result phase is much like some kind of "limitation period" – if we borrow this expression from the legal terminology. But this belongs to the **dynamic interpretation of ℜeALIS** which is yet to be implemented and researched. The preliminary results are expected to be published very soon.

4 ANCHORING NOUN PHRASES

Let s be the speaker and h be the addressee. Before any truth evaluation, the topical parts – mostly NPs – of the sentence (such as *Peter*, *the boy* etc.) must be **anchored to existing entities.** Here, only the definite case is described. (For the indefinite – specific or non-specific case, relevant sets have to be extracted from the context, which is not yet possible at the current stage of development.)

In most cases, uniqueness is needed to properly interpret the sentence. (Nick)names, like those in Examples 1 and 2 are the best examples to demonstrate this. Therefore, let us again use the sentence *Peter is married* to illustrate the case.

Let us take four sets of *Peter*s: P_1 to P_4. P_1 contains the entities known by s, the elements of P_4 are the ones known by h, P_2 and P_3 are assumed sets: s believes that h knows the elements of P_2 and h believes that s knows the elements of P_3.

To be pragmatically correct, $|P_1 \cap P_2|$ and $|P_3 \cap P_4|$ should be 1 and the two entities must be the same. If this is not true, uniqueness is not guaranteed from either the speaker's or the hearer's side.

Of course, in this case, uniqueness can be inferred from a wider context: *Peter has died.* Both s and h may know many *Peter*s but it is only one "common Peter" who actually died: both s and h might have known which Peter that was. Although Prolog is capable of performing even this task, it has not been implemented yet (we are assuming strict uniqueness), and also, it would slow down the program considerably.

5 PLANNED ARCHITECTURE

Prolog has two interfaces to Java: PrologBeans and Jasper. Since the prototype is mature enough for the Prolog core and the (future) interface to be separated and since this will render it very important to implement a multi-user interface (for internal and external users), we are considering building a web application from ℜeALIS, skipping the phase of a stand-alone graphical application. Moreover, because Jasper is only suitable to create stand-alone applications, **PrologBeans** will be used as an intermediate layer between the Prolog server and Java. Communication between PrologBeans and JSP is also quite well documented, so it seems possible to build two web-based interfaces for ℜeALIS: one for internal users (linguists and administrators) and one for external ones. Only internal users would have the right to add new linguistic elements and new semantic postulates.

Even later, the Prolog core might be extended with an SQL background to handle large databases. Although we have experimented with this, the actual implementation will greatly depend on the memory limits of SICStus Prolog and the actual memory consumption of the program.

ACKNOWLEDGEMENTS

We are grateful to SROP-4.2.2.C-11/1/KONV-2012-0005 (Well-being in the Information Society) for their financial contribution to ℜeALIS.

REFERENCES

Alberti, G., 2011. *ℜeALIS. Interpretálók a világban, világok az interpretálóban [Interpreters in the world, worlds in the interpreter]*, Academic Press. Budapest.

Alberti, G., Károly, M., 2010. The ℜeALIS Model of Human Interpreters and Its Application in Computational Linguistics. In *Proceedings of the 5th ICSOFT, Athens, Greece Vol. 2,* SCITEPRESS. 468–474.

Alberti, G., Károly, M., 2011. The Implemented Human Interpreter as a Database. In *Proceedings of the 5th IC3K-KEOD, Paris, France*, SCITEPRESS. 379–385.

Alberti, G., Károly, M., 2012. Multiple Level of Referents in Information State. In *Computational Linguistics and Intelligent Text Processing – 13th International Conference, CICLing 2012, New Delhi, India*, Springer. Berlin.

Alberti, G., Kleiber J., 2010. The Grammar of ℜeALIS and the Implementation of its Dynamic Interpretation. In *Informatica Vol. 34. No. 2.* 103–110.

Alberti, G., Kleiber, J., 2011. Where are Possible Worlds? (Arguments for ℜeALIS) In *4th Syntax, Phonolohy and Language Analysis Conference*. Budapest.

Asher, N., Lascarides, A., 2003. *Logics of Conversation*, Cambridge University Press. Cambridge.

Dowty, D. R., Wall, R. E., Peters, S., 1980. *Introduction to Montague Semantics*, D. Reidel Publishing Company. Dordrecht.

Kamp, H., van Genabith, J., Reyle, U., 2011. Discourse Representation Theory. In *Handbook of Philosophical Logic, Vol. 15*, Springer. Berlin. 125–394.

Kilián, I., 2013. A Metamodel-Driven Architecture for Generating, Populating and Manipulating "Possible Worlds" to Answer Questions. In *Proceedings of the 8th ICSOFT, Reykjavík, Iceland*, SCITEPRESS. To appear.

Seligman, J., Moss, L. S., 1997. Situation Theory. In *Handbook of Logic and Language*, Elsevier and MIT Press. Amsterdam and Cambridge. 239–309.

Vadász, N., Alberti, G., Kleiber, J. 2013. The Matrix of Beliefs, Desires and Intention – Sentence by Sentence. In *International Journal of Computational Linguistics and Applications Vol. 4 No. 1.* 95–110.