

# HOW CAN ONE EVALUATE A CONVERSATIONAL SOFTWARE AGENT FRAMEWORK?

*Dr Kulvinder Panesar*

*School of Art, Design and Computer Science, York St John University, York, UK*

## **Abstract**

This paper presents a critical evaluation framework for a linguistically orientated conversational software agent (CSA) (Panesar, 2017). The CSA prototype investigates the integration, intersection and interface of the language, knowledge, and speech act constructions (SAC) based on a grammatical object (Nolan, 2014), and the sub-model of belief, desires and intention (BDI) (Rao and Georgeff, 1995) and dialogue management (DM) for natural language processing (NLP). A long-standing issue within NLP CSA systems is refining the accuracy of interpretation to provide realistic dialogue to support the human-to-computer communication.

This prototype constitutes three phase models: (1) a linguistic model based on a functional linguistic theory – Role and Reference Grammar (RRG) (Van Valin Jr, 2005); (2) Agent Cognitive Model with two inner models: (a) knowledge representation model employing conceptual graphs serialised to Resource Description Framework (RDF); (b) a planning model underpinned by BDI concepts (Wooldridge, 2013) and intentionality (Searle, 1983) and rational interaction (Cohen and Levesque, 1990); and (3) a dialogue model employing common ground (Stalnaker, 2002).

The evaluation approach for this Java-based prototype and its phase models is a multi-approach driven by grammatical testing (English language utterances), software engineering and agent practice. A set of evaluation criteria are grouped per phase model, and the testing framework aims to test the interface, intersection and integration of all phase models and their inner models. This multi-approach encompasses checking performance both at internal processing, stages per model and post-implementation assessments of the goals of RRG, and RRG based specifics tests.

The empirical evaluations demonstrate that the CSA is a proof-of-concept, demonstrating RRG's fitness for purpose for describing, and explaining phenomena, language processing and knowledge, and computational adequacy. Contrastingly, evaluations identify the complexity of lower level computational mappings of NL – agent to ontology with semantic gaps, and further addressed by a lexical bridging consideration (Panesar, 2017).

## **1 Introduction to the conversational agent**

A conversational software agent (CSA) is a program that engages in conversation using natural language (NL) dialogue with a human user. NL is the most easily understood knowledge representation for people, but certainly not the best for computers because NL is inherently ambiguous, complex and dynamic. The challenge here is for the system to encapsulate sufficient knowledge from the user's question to present a grammatically correct response. Here a discussion of knowledge and its role is very important. When a person hears or sees a sentence, an individual will make full use of their knowledge and intelligence to understand it; additionally, besides the grammar, knowledge about the words, the sentence context, and an understanding of the subject matter is necessary. Here the computational model, must model this language understanding and the interactions that combine grammar, semantics and reasoning.

Two main requirements for a CSA specified by Mao et al. (2012) and Lester et al. (2004) are the ability of accurate natural language understanding (NLU) and the technical integration with an application. A CSA must respond appropriately to the user's utterance via three phases: (1) interpret the utterance, (2) determine the actions (logic) that should be taken in response to the utterance (3) perform actions replying with text.

In the last decades, there has been a great evolution in the field of CSA enfolded three emerging trends of more sophisticated natural language processing (NLP) via improved parsing techniques, humanising of agents through language and their pervasiveness (Perez-Marin et al., 2011). CSAs are becoming prominent in the marketplace where the consumer's use and familiarisation of the smartphone technology and consumer support agents, human-computer interaction, and ubiquitous computing is maturing, and is of great value.

A long-standing issue within NLP CSA systems is refining the accuracy of the interpretation of meaning to provide a realistic dialogue to support the human-to-computer communication. The focus here is that NL is conceived of as a functional system (Francois, 2014).

A functional mature grammar - Role and Reference Grammar (RRG) is deployed as a linguistic theory that can adequately explain, describe and embed the communication-cognitive thinking in conversation, in a computational form. With this in mind, we propose a language model, knowledge model, speech act constructions (SAC) and belief, desires, and intentions (BDI) framework for a linguistically motivated text based CSA namely LING-CSA. It will further investigate how language can be comprehended and produced, to gain a deep understanding, and how it interfaces with knowledge. In order to use the RRG linking system and to create an effective parser, RRG is re-organised to facilitate the innovative use of SAC as grammatical objects (Nolan, 2014) and rich lexicon.

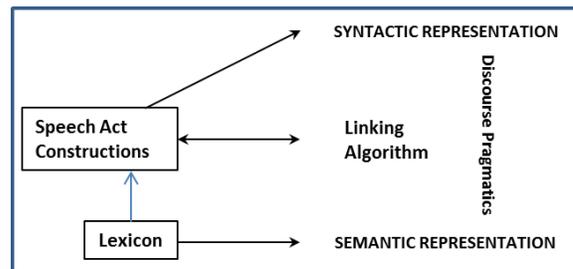


Figure 1 Re-organisation of RRG for the CSA

This paper presents a critical evaluation framework for LING-CSA (Panesar, 2017). It investigates the evaluation of the integration, intersection and interface of the language, knowledge, and speech act constructions (SAC) based on a grammatical object (Nolan, 2014), and the sub-model of belief, desires and intention (BDI) (Rao and Georgeff, 1995) and dialogue management (DM) for natural language processing (NLP).

This objectives of this paper: (1) an account of the conceptual architecture of LING-CSA; (2) implementation details; (3) an evaluation and testing of the CSA framework via a range of evaluation criteria and series of tests, (4) a summary of the findings; (5) conclusions and recommendations.

## 2 Conceptual architecture of the conversational software agent

LING-CSA constitutes three phase models as illustrated in Figure 2 below.

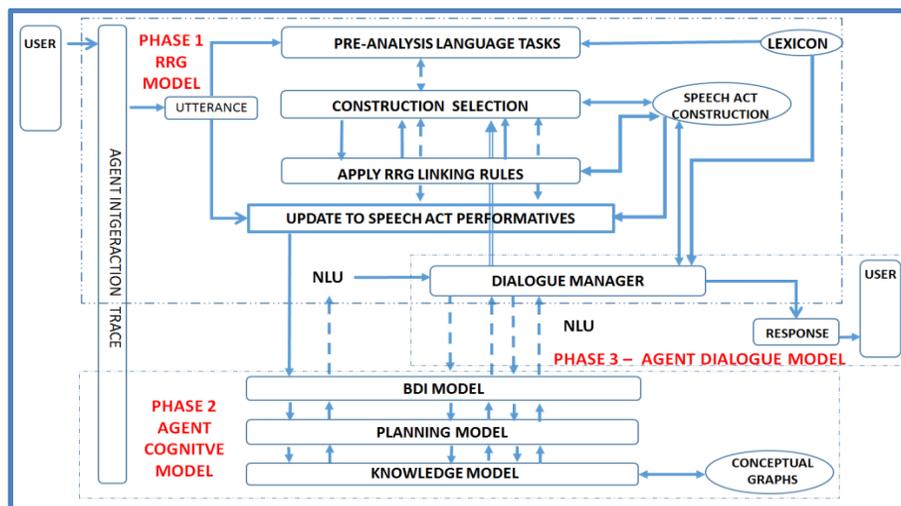


Figure 2 Conceptual architecture of the LING-CSA (Panesar, 2017)

Figure 2 identifies: (1) RRG language model, (2) Agent Cognitive model (ACM) with two inner models of: (a) knowledge model; (b) planning model; (3) Phase 3 – Agent Dialogue Model (ADM) where the DM is a common component of Phases 1 and Phase 3 – due to the discourse referents of an utterance, and the need to create a grammatically correct response.

### 2.1 RRG Language Model (Phase 1)

The LING-CSA will constitute a closed domain - dialogue on food and cooking, for its richness and universality. A range of NLP tasks manipulated via the language model will include tokenisation, sentence splitting, part-of-speech-tagging, morphological analysis, syntactic and semantic parsing. It is the DM that will assist with the SA dialogue working with the syntactic parser to work effectively with the data sources (word list, lexicon, empty speech acts constructions (SACs)). RRG manipulations exist for both simple and complex sentences, but the LING-CSA is limited to the manipulation of the linking system of simple sentences (active or passive) using transitive, intransitive and ditransitive verbs/auxiliary verbs with variable word order flexibility with SVO, and modelled via speech acts (Searle, 1969) applied to cognitive and rational interaction analysis. RRG’s bi-directional linking algorithm and discourse-pragmatics interface will be mapped into completed SACs, invoking the lexicon to supports all three LING-CSA phases, and internal BDI manipulations (Panesar, 2017).

Note: Logical structures have been partially expanded for readability in this paper

Table 1 - Snapshot of the lexicon - lexical entries and lexemes (data source 1)

LEXICAL ENTRY	POS-TYPE	VERB TENSE/ ASPECT	DEF	P TYPE	NO	GR	CASE	ANIM	HUM	LOGICAL STRUCTURE (LS)
eat	VERB	PRS/ FUT	DEF+/-	3	SG	M/F	DNA	ANIM	HUM	<tns:prs <do'(x, [eat'(x,y)] ) & BECOME consumed'(y) >> <tns:fut <do'(x, [eat'(x,y)] ) & BECOME consumed'(y) >>
eating	VN	PROG	DEF+/-	3	SG	M/F	DNA	ANIM	HUM	<tns:prs <asp:prog <do'(x, [eat'(x, y)] ) & BECOME consumed'(y) >>>
is	VBE	DNA	DEF+	DNA	DNA	DNA	DNA	DNA	DNA	be'(x,[pred'])
hungry	ADJ	DNA	DNA	DNA	DNA	M/F	DNA	ANIM	HUM	DNA
restaurant	N	DNA	DEF+/-	DNA	SG/PL	DNA	DNA	DNA	DNA	DNA

Table -2 Overview of the RRG steps (Panesar, 2017)

Step	PHASE	(TRACKING) DISPLAY
<b>PHASE 1 - RRG LANGUAGE MODEL</b>		
1	PHASE 1A - PRE-ANALYSIS	Parsing, Sentence and word tokenisation, check against the lexicon.
2	PHASE 1B – SAC SELECTION	Construction selection computations
3	PHASE 1C – SYNTAX-TO-SEMANTICS	<ul style="list-style-type: none"> <li>• Verb selection and voice</li> <li>• Retrieve the LS and verb from the lexicon</li> <li>• Assignment of arguments to the LS arguments</li> <li>• Semantic representation</li> <li>• Update the LS</li> <li>• Map the operator projection and focus projections</li> <li>• Actor and undergoer assignment</li> <li>• Select the morphosyntactic properties of the argument</li> </ul>
4	PHASE 1E -	• Verb selection and voice
		Display the utterance and various internal representations.
		Display construction selected
		Display the verb selection and voice topics
		Display the LS and verb from the lexicon
		Display the argument value
		Display the semantic representation
		Map the operators to the Tense and illocutionary force
		Display the actor and undergoer as part of the LS.
		Display the additional information
		Display the verb selection and voice

	SEMANTICS-TO-SYNTAX	<ul style="list-style-type: none"> <li>Retrieve the LS and verb from the lexicon</li> <li>Assignment of arguments to the LS arguments</li> </ul>	topics Display the LS and verb from the lexicon Display the argument values
5	PHASE 1F – DIALOGUE MANAGER	<ul style="list-style-type: none"> <li>Ascertain whether there is any missing information</li> <li>Refer to the stored discourse referents – if any, and process with the earlier Phase 1 steps</li> </ul>	Display the specific elements that missing Display appropriately any saved discourse referents
6	PHASE 1G – CREATE SAP	<ul style="list-style-type: none"> <li>Attach the performative information to the Speech Act Construction</li> </ul>	Display the performative type information

The analysis of utterance/responses will be constrained to assertive and interrogative (WH-word) SAs. SAC example shown in Table 2 include: - ‘assertive: ATE’, taken from a sample collection of 34.

The selected SAC, is updated with all the grammatical information. These include: voice opposition; macrorole; pragmatic; semantic; lexical; the PSA (privileged syntactic argument)- similar to the ‘subject’ of S-V-O; matching signature; syntactic morphology (rules); focus; semantic information such as tense and aspect. Here the updated SAC will use the generalised lexicon and computationally work with the surface syntax to the underlying semantic forms. Any discourse referents that are generated are also checked and updated accordingly. The updated SAC will store each text and the associated complete logical structure (LS) based on the working semantic predicating element. The linking system will facilitate the syntactic parse to enable fixed word order for English (SVO), and to unpack the agreement features between elements of the sentence into a semantic representation (the logical structure) and a representation of the Layered Structure of the Clause (LSC). The linking system will facilitate procedures for semantic-to-syntax and syntax-to-semantics, parsing and in the process of formulating a grammatical correct response.

### 3.2 Design framework - Agent Cognitive Model (Phase 2)

Cohen and Levesque (1988) take the viewpoint of language as action, and view utterances as events (updated to speech act performatives (SAP) messages) that change the state of the world, and hence speakers and hearer’s mental state change as a result of these utterances. Further it is necessary for a computer program to recognise the illocutionary act (IA) of a SA, for both the speaker (USER) utterance and hearer (AGENT) and a response. The single agent environment will constitute a language model, mental model to work the BDI states, working model of memory (to reason with states based on current knowledge), knowledge model (world knowledge – made up of shared and individual beliefs) and dialogue model (to provide a response back to the user). The next question is how will this integrate and function; this is achieved in the Phase 2 – Agent Cognitive Model (ACM) with an input of the Phase 1 RRG Language Model. The ACM contains two inner models namely the Agent Knowledge Model (AKM) and Agent Dialogue Model (ADM). The ACM Phase 2 model has a series of pre-agent steps and main agent steps illustrated in Figure 5.

The behaviour of agent is identified by the following basic loop where agent iterates the following two steps at regular intervals: (1) Read the current message, and update the mental state including the beliefs and commitments; (2) Execute the commitments for the current time, possibly resulting in a further belief change. The preliminary steps (Panesar, 2017) involve Step Pre-Agent (1) Create Belief Base - the agent’s knowledge; Step Pre-Agent (2) Map to Message Format - the appropriate Speech Act Performative (SAP) (from the Phase 1-RRG Language Model) is re-mapped into a data structure to reduce redundancy and enable efficiency in the planning and reasoning process.

The main steps (Panesar, 2017) are identified in (1):

- (1)

- a) Step (1) Agent Perceive - to perceive the environment, and ascertain the state of the environment, this event is added to the event queue (EQ);
- b) Step (2) Agent Query - the current beliefs – here the user’s beliefs are queried against the belief base - this event is added to the EQ;
- c) Step (3) Agent Select Plan - invokes the list of possible plans, and asserts the best way forward; this event is added to the EQ;
- d) Step (4) Agent -Update Belief - if appropriate the agent’s knowledge is updated;
- e) Step (5) Agent Select goal - here a goal is selected from the intention stack – stored as part of the plan library . This could be simple goal such as “True assertion”; or “missing information” for instance, the location, or the object, or it temporal information;
- f) Step (6) Agent Execute Action–involves invoking the Agent Dialogue Model, and the RRG Language Model for an effective Speech Act performative message back to the Agent;
- g) Step (7) Agent Event Handler - handles each different type of event - from the perception (of messages) , querying (belief base), updating (belief base), selecting (a plan, and further goal), to executing an action. Prior to checking the BDI states of the user and agent, the SAC is updated to include the performative information of the message in an agent environment.

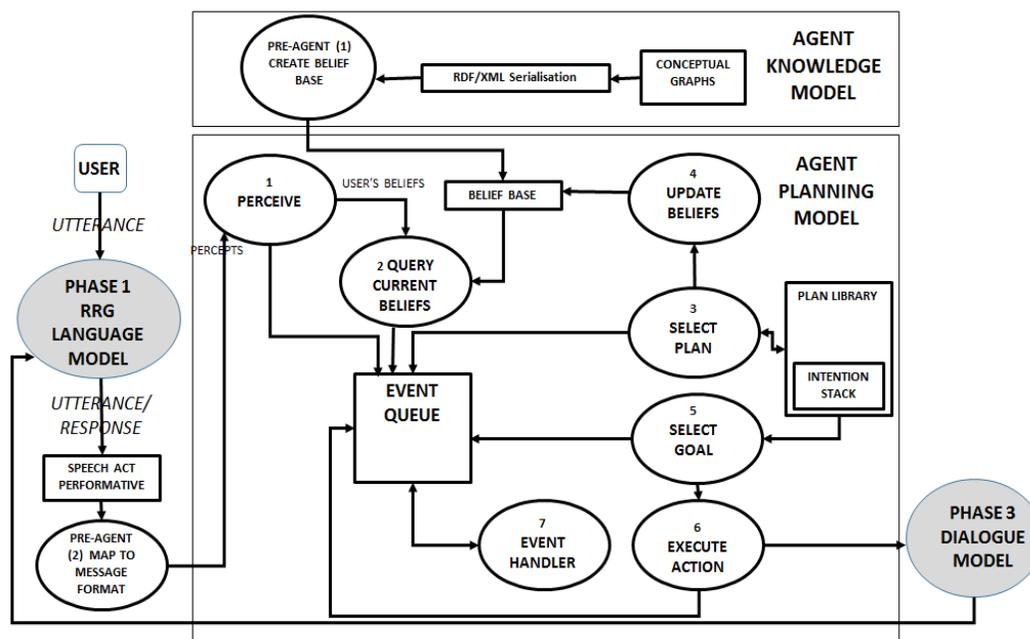


Figure 3 - Design framework - CSA Agent Cognitive Model (Panesar, 2017)

Bratman (1987) identifies the workings of how people make decisions and take action in the form of the mental attitudes of beliefs, desires and intentions - goals (BDI) and in relation to rational interaction. In relation to the conversation, the methods used will work based on a sentence pair. The user instigates the utterance into the CSA framework, and it is the agent which decides on the appropriate dialogue and grammatical response. The sentence pair and sub dialogue internal representations will be stored – based on theories and models such as the discourse representation theory (DRT) (Kamp et al., 2011). To facilitate conversation, the DM is invoked, and discourse referents in the previous utterance of the sub-dialogue are resolved. This will serve two purposes: 1) to establish the NLU of the utterance; 2) to forward to the dialogue model to ascertain a response. This response generator here will further make use of the language model, and speech act construction model, to formulate a grammatically correct response.

### 3 Implementation details

Figures 1 to 3 are mapped to an operational framework, and implemented as a Java prototype, developed in Eclipse IDE platform, predominantly as POJO (plain old java objects) with some API support, based on a food and cooking domain (ontology) due to its rich and representative (different lexical entries) nature. This prototype constitutes three phase models: (1) a linguistic model based on a functional linguistic theory – Role and Reference Grammar (RRG) (Van Valin Jr, 2005); (2) Agent Cognitive Model with two inner models: (a) knowledge representation model employing conceptual graphs serialised to Resource Description Framework (RDF); (b) a planning model underpinned by BDI concepts (Wooldridge, 2013) and intentionality (Searle, 1983) and rational interaction (Cohen and Levesque, 1990); and (3) a dialogue model employing common ground (Stalnaker, 2002). In the next section, we present an evaluation framework for LING-CSA.

### 4 Discussion of the selection evaluation methodology and testing framework

The purpose of evaluation is to assess the system to see if does what it is supposed to do. Radziwill and Benton (2017) present a case of reviewing articles on evaluating CSA using terms such as ‘*quality assurance*’, ‘*evaluation*’, and ‘*assessment*’ over the years 2016 and 2017 and concluded that the quality of CSAs was aligned with the ISO 9241 concept of usability: “The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments” (Abran et al., 2003). These outcomes: (1) efficiency considered performance; (2) effectiveness considered functionality and humanity; (3) satisfaction considered affect, ethics and behaviour and accessibility. One interesting attribute was in question: whether or not a CSA should pass the Turing Test (Turing, 1950). According to most researchers, and following the earliest conversational interfaces, responding and interacting like a human should be the top priority, and subsequently this principle guided development for nearly four decades since ELIZA (Weizenbaum, 1966) came online. However, it is critical that the goal of CSA should not aim to act human - is shared by some analysts like Wilson et al. (2017), and further every implementation will have different goals and will prioritize different quality attributes at different phases of the system’s lifecycle, using various quality assessment approaches from test scripts, question-and-answer, linguistic quality, and metric-based to name a few (Radziwill and Benton, 2017).

Subsequently, a customised multi-approach assessment will be adopted to address the unique goals of the LING-CSA based on the phase models, checking performance both at internal processing, interfacing of the models and their stages and post-implementation assessments, invoking the tracking log as a qualitative analysis. Testing strategy is driven by: (a) grammatical testing (English language utterances) and NLP tasks; (b) software engineering (UML modelling, architecture centric, data structures, algorithms and patterns); (c) knowledge representation (KR) logics (first order logics and graph theory) and important performance indicators for KRR as: i) representational adequacy ii) inferential adequacy – is the ability to represent the appropriate inference procedures; (d) agent practice (message passing, planning and cognitive responses and conforming to its appropriate inputs, environment, actions, performance measures (IEAP) (Russell and Norvig, 2016); (e) A range of RRG tests applied at various representations to address the goals of RRG - explanatory, descriptive, cognitive and computational adequacies (Panesar, 2017). A set of evaluation criteria are identified in (3):

(3)

- a) Criteria 1- Could the system present a mapping of the syntactic representation to a semantic representation, for the utterance taking the form of a simple sentence?
- b) Criteria 2 - Could the system present an adequate explanation of the NLU of the utterance?
- c) Criteria 3 - Could the system demonstrate the SAC use in the manipulation of the utterance?
- d) Criteria 4 - Can the dialogue manager interface the language model?
- e) Criteria 5 - Could the system demonstrate the agent BDI and knowledge representation?
- f) Criteria 6 - Could the system represent the user’s BDI states?
- g) Criteria 7 – Could the system query the knowledge base for a fact (from the speech act performative)
- h) Criteria 8 - Could the system devise an appropriate plan based on the BDI states?
- i) Criteria 9 - Could the system generate a grammatically correct response in RRG based on the agent’s knowledge?

## 5 Detailed evaluation of the CSA framework

This section evaluates the LING-CSA framework and three phase models with criteria considered by grouping per model, and assessed with some supporting tests, and their summative findings and screen captures.

### 5.1 RRG model evaluations

RRG is a linguistic theory invoked to facilitate the implementation of a linguistic engine for the NLP and understanding for the CSA. RRG concepts will be used for the linguistic manipulation of the utterance and the response. Criteria (1-4) are reiterated below:

- Criteria 1- Could the system present a mapping of the syntactic representation to a semantic representation, for the utterance taking the form of a simple sentence?
- Criteria 2 - Could the system present an adequate explanation of the NLU of the utterance?
- Criteria 3 - Could the system demonstrate the SAC use in the manipulation of the utterance?
- Criteria 4 - Can the dialogue manager interface the language model?

#### 5.1.1 Criteria 1 – Syntactic and semantic representation

The syntactic representation (layered structure of the clause - LSC) and the semantic representation (logical structure LS) are part of the basic concepts of the RRG functional linguistic theory. Further in the CSA, the LS and LSC work hand in hand together via the RRG linking system, and the semantic-to-syntax algorithm, and the syntax-to-semantics for the production and the comprehension of the utterance. As part of the implementation the linguistic algorithms and principles have been consulted, and developed into CSA logics. Table 3 identifies a range of utterances used for checking a snapshot of syntactic and semantic representations.

The utterance example ‘they are hungry’ is highlighted in Figures 4 and 5. Here the dialogue management work is resolved. The tense, aspect, privileged syntactic argument (PSA), actor-undergoer hierarchy (AUH) and case marking are also assigned, and LSC and LS representations generated.

Table 3 - Set of utterances for testing RRG linking system

No	Utterance	Type of verb	Logical Structure info and number of nominals in the LS Syntactic representation (SR); Expected results	Actual results
1	Gareth ate	Intransitive Verb ‘ate’	1 argument and 1 nominal LS (do (ate) and main clause (SR)	√
2	Gareth eats the pizza	Verbal noun	2 arguments and 2 nominals LS (do (eat) and main clause (SR)	√
3	Gareth is hungry	Copula verb with an adjective	1 argument and 1 predicating argument LS (be) and Main clause (SR)	√
4	Gareth ate the pizza quickly	Transitive	2 arguments LS (do (eat). SR has a main clause and ‘quickly’ is a periphery	√
5	Yesterday Gareth ate in the pizza	Transitive	2 arguments LS (do (eat) SR has a left detached position (LDP) yesterday’ and main clause has ‘quickly’ as the periphery constituent.	√
6	The pizza is in the oven	Copula verb and be-LOC	2 arguments LS (be (is) and main clause has ‘the pizza’ and ‘the oven’ are both noun phrases (NP).	√
7	Where is the pizza	WH- word Copula verb	1 argument, 2 nominals. Here ‘where’ is replaced in the first argument position as a nominal of the LS. SR – ‘where’ token takes on noun phrase (NP)	√
8	Gareth ate the pizza in the restaurant	Transitive	3 arguments and 2 LS LS (do (eat) and be-in and main clause (SR) Main clause and ‘in the restaurant’ is a periphery preposition (PP).	Partially

```

Console
<terminated> MainCAversion28 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (5 Apr 2017, 02:18:19)
**** PSA & AUH & CASE SELECTION - 1 ARGUMENT AND PREDICATING ELEMENT ****
*****
PSA > they
ACTOR > they
NOM case > they
Updating the selected LS with its arguments
*****
Working LS > be'(x,[pred'])
Tense value > <PST>
Working LS > <IF>ASS<TNS><PST> be'(x,[pred'])
x > they
pred > hungry
WorkingLS length is 34
x=they
pred=hungry
current index is 28 Updated LS > <IF>ASS<TNS><PST> be'(they,[hungry'])
Back to Main - Phase 1 LS is <IF>ASS<TNS><PST> be'(they,[hungry'])

```

Figure 4 - Utterance - 'they are hungry' - LS and case marking

```

Console
<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 20
*****
Syntactic representation of this utterance >>>>>
SENTENCE ( <LDP> ( CLAUSE ( <CORE> <NP> they ( <NUC> ( <PRED> <AUX> are ) )
ADJ hungry ) )
*****
Speech Act Performative
*****
:::Performative =SAP ASSERTIVE are :::Sender =<USER>:::Receiver<AGENT:::ontology
= :::Signature =[PRO VBE ADJ]:::Constraint =DEFAULT:::Input =they are
hungry:::Workspace =[they, PRO], [are, VBE], [hungry, ADJ]:::Syntax =SENTENCE (
<LDP> ( CLAUSE ( <CORE> <NP> they ( <NUC> ( <PRED> <AUX> are ) ) ADJ hungry
) ) ::: PSA =they:::SemanticsRRG =NONE:::Linking =CONTAINS A NOUN PHRASE BEFORE
AND AFTER THE VERB:::Morphology =DEFAULT:::Pragmatics =TRUE/FALSE:::IllForce
=ASSERTIVE:::FocusStructure=NARROW FOCUS ON THE ELEMENT:::OutputLS
=<IF>ASS<TNS><PRT> be'(they,[hungry'])

```

Figure 5 - Utterance - 'they are hungry' - LSC and LS representations

Table 3 highlights a range of snapshot utterances, from the full range of the tense and aspect spectrum. Here the auxiliary verb is followed by a verb or an adjective, and every utterance has a matching speech act construction and their respective signature that can be processed by the CSA.

Example 9 Analysis - 'Gareth ate the pizza in the restaurant', it worked partially.

The syntactic representation was generated correctly as:

```

SENTENCE ( CLAUSE ( <CORE> <NP> gareth ( <NUC> ( <PRED> <V> ate ) ) ( <NP> ( the
pizza ) ) ( PERIPHERY ( in ( <NP> ( the restaurant ) ) ) ) )

```

However the semantic representation is:

```

<IF>ASS<TNS>PT do'(gareth,[eat'(gareth,restaurant)]) & INGR
consumed'(restaurant)<IF>ASS<TNS>PT be-in'(gareth,restaurant)

```

Here the propositional assignment rules have been applied, but some refinement of logic rules is still necessary, to ensure the macrorole assignment is accurate, using a range of agreement features. From the tense and aspect list a few entries have been challenging for the LS. For example, Gareth was invited to dinner, presents two LSs. Alternatively, ‘she has to eat dinner’ has two LSs with an outputLS as: <IF>ASS<TNS><PST> do’(she,[eat’(she,dinner)]) & BECOME consumed’(dinner)<IF>ASS<TNS><PST> has’(she,dinner).

From a LSC perspective, some of the syntactic representations include the auxiliary verb and not the main verb. For example, ‘I can cook dinner’ creates a correct semantic representation, but a syntactic representation with one token missing.

To summarise for this criteria, the CSA can produce both a semantic and syntactic representation of an utterance, in most cases accurately. The specific limitations are discussed in the final section.

### 5.1.2 Criteria 2 – Natural language understanding

For the pre-analysis tasks – tokens were checked for viability, and furthered stemmed. The WordNet API was tested for some stems – which provided inappropriate answers for this model – hence a stemming algorithm was coded. For example in the CSA, the token ‘serving’ would be reduced to the base verb of ‘serve’ as opposed to ‘serv’ an answer provided by the WordNet API. The base verb of ‘serve’ is ‘serve’, and the correct SAC is selected, with a range of valid signatures. Figure 6 identifies the base verb.

```

<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 20
-----
Selecting the appropriate Speech Act Construction
-----
The signature of the utterance = [[gareth, PN], [is, VBE], [eating, VN], [the, DET],
[pizza, N]] is [PN, VBE, VN, DET, N]
source string [[gareth, PN], [is, VBE], [eating, VN], [the, DET], [pizza, N]]
Column 1 is PN
Column 1 is VBE
Column 1 is VN
Column 1 is DET
Column 1 is N
Selected verb for checking is > eating
No of characters of tempVerb is 6
Verb to analyse is eating
Extract some of the base verb > eat
Extract the ending > t
Extract the double ending > tt
Selected base verb form for the Speech Act Construction > eat
Size of the base verb form > 3
Verbal Noun/Adjective working here as a verb is eat
Back to main - Verb is eat

```

Figure 6 - Stemming feature to extract the base verb

### 5.1.3 Criteria 3 - Speech Act Constructions (SAC)

NLP sentences make use of lexical, syntactic, semantic and pragmatic information in order to fulfil the role of conveying meaning. The SAC is a grammatical object/container to contain some default information for the speech act type, and has the capacity for a series of computational updates.

For RRG manipulations, it is the application of speech act theory to utterances and conversations, which facilitate the interaction with the agent cognitive model and dialogue model. Once an utterance is uttered it is the illocutionary act – what is to be done – which takes precedence. Here the illocutionary act could be an assertive or interrogative statement (WH-words), with a desire for the hearer, to form some correlative attitude and hence act in a certain way. The adoption of SAC has worked well, both from the point of view of the selection of the SAC, and stages of populating of it, from the application of the RRG linking system. This is demonstrated in Figure 7.

SACs effectively extends to create a speech act performative (SAP) to form a container for a message that will form an input in to the Agent Cognitive Model, as shown in Figure 8. This SAP with its seventeen attributes is available in the agent environment to establish the user’s belief states and further support the querying of the agent’s knowledge base and planning a response back to the user.

To summarise, the SAC as a grammatical object together with the rich RRG Lexicon has enabled an effective re-organisation and integration of the RRG Linking system.

```

<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 20
-----
Selecting the appropriate Speech Act Construction
-----
The signature of the utterance = [[gareth, PN], [ate, VERB], [the, DET], [pizza, N]]
is [PN, VERB, DET, N]
source string [[gareth, PN], [ate, VERB], [the, DET], [pizza, N]]
Column 1 is PN
Column 1 is VERB
Column 1 is DET
Column 1 is N
Back to main - Verb is ate

Signature to check is [PN, VERB, DET, N]
Construction type is ASSERTIVE:ATE
Construction specific signatures > [NP VERB NP], [PN VERB], [ADV PN VERB DET N], [PN
VERB N ADJ], [PRP DET N PN VERB DET N], [PN VBE VERB N], [PN PRP DET N PRP DET N],
[PRO VERB DET N], [PN VERB NP], [PN VERB DET N], [NP VERB QNT N], [DET N VERB DET
N], [DET N VERB QNT N], [NP VERB (DET) (ADJ) N (ADJ)], [PN VERB DET N ADJ], [PN VERB
(DET) ADV N ADJ], [PN VERB DET N PRP DET N], [PN, VERB, N, PRP, DET, N], [PN VERB N
PRP DET N]
Utterance signature to check > [PN VERB DET N]
-----
matches the construction
matches the signature
  
```

Figure 7 Selection of the Speech Act Construction

```

<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun
*****
Speech Act Performative
*****

:::Performative =SAP ASSERTIVE ate :::Sender =<USER>:::Receiver<AGENT:::ontology
= :::Signature =[PN VERB DET N]:::Constraint =DEFAULT:::Input =gareth ate the
pizza:::Workspace =[[gareth, PN], [ate, VERB], [the, DET], [pizza, N]]:::Syntax
=SENTENCE ( CLAUSE ( <CORE> <NP> gareth ( <NUC> ( <PRED> <V> ate ) ) ( <NP>
( the pizza ) ) ) ) ::: PSA =gareth:::SemanticsRRG =NONE:::Linking =CONTAINS A
NOUN PHRASE BEFORE AND AFTER THE VERB:::Morphology =DEFAULT:::Pragmatics
=TRUE/FALSE:::IllForce =ASSERTIVE:::FocusStructure=NARROW FOCUS ON THE
ELEMENT:::OutputLS =<IF>ASS<TNS>PT do'(gareth,[eat'(gareth,pizza)]) & INGR
consumed'(pizza)
  
```

Figure 8 - Speech Act Performative – message format > Agent Cognitive Model

#### 4.2.4 Criteria 4 - Dialogue manager

In the LING-CSA, after the standard utterance analysis steps, the dialogue manager facility is an interim process, which initially automatically checks for missing information or whether reference has been made to the other discourse elements, such as the pronoun category as demonstrated in Figure 9 Dialogue Preparation (1).

Dialogue Preparations (2) is involves making use of the current utterance and further extracting and saving the nominals for future utterances. The nominals are saved in an output file ‘DiscourseReferents.txt’ which is saved locally in the resource folder of the CSA program. The file contains a token with its associated POS tag. This information will be used by another utterance, via another execution of the CSA program.

```

Console
<terminated> MainCAVersion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\jav
*****DIALOGUE PREPARATIONS 1*****
*****
CHECKING FOR THE NEED OF ANAPHORIC RESOLUTIONS
*****

The number of referents that need resolving are >>>0
Further checking for pronouns >> false
*****
*****DIALOGUE PREPARATIONS 2 *****
*****
SAVING NOMINALS FOR LATER DIALOGUE
*****

Current tokens are: [[gareth, PN], [ate, VERB], [the, DET], [pizza, N]]
Saving a map for a required set of discourse referents
gareth=PN
pizza=N
-----
Saving the nominals map to the output resource file
-----

DiscourseReferentsMap.txt
1 gareth PN
2 pizza N

```

Figure 9 - Dialogue Preparations (1) and (2)

Taking the example, ‘he is in the restaurant’, Figure 9 illustrates the re-loading of these nominals in microlexicon. The new utterance has a ‘he’ which is a pronoun that needs resolving, and its own discourse lexicon is generated as well.

```

Console
<terminated> MainCAVersion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 2017, 17:24:49)
*****DIALOGUE PREPARATIONS 3 *****
*****
Reading nominals from the resource file, and creating a micro-lexicon discourse - For USE IF NECESSARY
-----

Reading from Discourse referents file
line is gareth PN
line is pizza N
pizza is N
gareth is PN
microLexDisc [158][0] = pizza
microLexDisc [83][0] = gareth
microLexDisc[0][0] = pizza microLexDisc[0][1] = N microLexDisc[0][2] = DNA microLexDisc[0][3] = DEF+ microLexDisc[0][4] = DNA microLexDisc[0][5] = SG microLexDisc[0][6] = DNA
microLexDisc[0][7] = DNA microLexDisc[0][8] = DNA microLexDisc[0][9] = DNA microLexDisc[0][10] = DNA
microLexDisc[1][0] = gareth microLexDisc[1][1] = PN microLexDisc[1][2] = DNA
microLexDisc[1][3] = DEF+/- microLexDisc[1][4] = DNA microLexDisc[1][5] = DNA
microLexDisc[1][6] = M microLexDisc[1][7] = NOM microLexDisc[1][8] = ANIM microLexDisc[1][9] = HUM microLexDisc[1][10] = DNA
Micro-lexicon discourse elements are: = [[pizza, N, DNA, DEF+, DNA, SG, DNA, DNA, DNA, DNA, DNA], [gareth, PN, DNA, DEF+/-, DNA, DNA, M, NOM, ANIM, HUM, DNA]]

Check further for pronouns and resolve with discourse referents>>> true

```

Figure 10 - Dialogue management (3)

Through a series of rules and using the agreement features such as gender, person, definiteness, animate and human, the pronoun can be resolved as demonstrated in Dialogue Preparations (3) and (4) in Figure 10 and Figure 11 respectively. The new utterance generated is ‘Gareth is in the restaurant’ and thus can go forth for further RRG processing to the selection of a SAC.

To summarise this dialogue management makes use of simple discourse representation theory (DRT), and forms a reusable feature as part of the agent planning process to formulate a grammatical correct response. For this reason the Dialogue Model computationally has been combined with the Agent Cognitive Model.

```

<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 2017, 17:24:49)
*****DIALOGUE PREPARATIONS 4*****
-----
Preparing for a new Utterance with nominals added
-----
Discourse elements are:
he=PRO
*****
Creating the referent lexicon
*****
he=PRO
refLexicon [95][0] = he
refLexicon[0][0] = he refLexicon[0][1] = PRO refLexicon[0][2] = DNA refLexicon[0][3] = DEF+
refLexicon[0][4] = 3 refLexicon[0][5] = SG refLexicon[0][6] = M refLexicon[0][7] = NOM refLexicon[0]
[8] = ANIM refLexicon[0][9] = HUM refLexicon[0][10] = DNA
Reference elements lexicon: = [[he, PRO, DNA, DEF+, 3, SG, M, NOM, ANIM, HUM, DNA]]

*****
Replace the pronoun with correct agreement features
*****
Existing lexicon : = [[he, PRO, DNA, DEF+, 3, SG, M, NOM, ANIM, HUM, DNA], [is, VBE, PRT, DEF+, 1, SG,
DNA, DNA, DNA, HUM, be'(x,[pred'])], [in, PRP, DNA, DEF+, DNA, DNA, DNA, DNA, DNA, DNA, be-in'(x,y)], [the,
DET, DNA, DNA, DNA, SG/PL, DNA, NOM/ACC, DNA, DNA, DNA], [restaurant, N, DNA, DEF+/-, DNA, SG, DNA, DNA,
DNA, DNA, DNA]]
Nom lexicon : = [[pizza, N, DNA, DEF+, DNA, SG, DNA, DNA, DNA, DNA, DNA], [gareth, PN, DNA, DEF+/-, DNA,
DNA, M, NOM, ANIM, HUM, DNA]]
Disc lexicon : = [[he, PRO, DNA, DEF+, 3, SG, M, NOM, ANIM, HUM, DNA]]

Nominal resolutions achiieved!

New utterance lexicon : = [[gareth, PN, DNA, DEF+, 3, SG, M, NOM, ANIM, HUM, DNA], [is, VBE, PRT, DEF+, 1,
SG, DNA, DNA, DNA, HUM, be'(x,[pred'])], [in, PRP, DNA, DEF+, DNA, DNA, DNA, DNA, DNA, DNA, be-in'(x,y)],
[the, DET, DNA, DNA, DNA, SG/PL, DNA, NOM/ACC, DNA, DNA, DNA], [restaurant, N, DNA, DEF+/-, DNA, SG, DNA,
DNA, DNA, DNA]]

```

DiscourseReferentsMap.txt

```

1 he PRO
2 restaurant N

```

Figure 11 - Dialogue management (4)

#### 4.1.5 Assess the CSA in terms of the goals of the linguistic theory

In this section we have demonstrated that criteria 3 (a-d) have been achieved as part of the language model. In order to ascertain the success of how RRG has performed as a linguistic engine for the CSA, it is important to assess the goals of a linguistic theory, and critique them, with the main theoretical characteristics of RRG in relation to grammatical relations.

Starting with (a) describing linguistic phenomena (central goal) - the LING-CSA can describe the utterance at various linguistic levels such as the syntax (layered structure of the clause). Next (b) explaining linguistic phenomena - the CSA provides semantic representations (logical structure) of simple sentences in English, which provide an explanation of the utterance, as well as the representation of meaning. Next (c) language processing and knowledge - has been achieved in the CSA, via invoking the RRG lexicon, as a form of knowledge of the language, and the selection of a SAC to facilitate the producing and the comprehension of the utterance. Further, the syntax-to-semantic and semantic-to-syntax algorithms provide a framework for processing language – for example, take a ‘wh-word’, it is the semantic-to-syntax algorithm which states that the ‘wh-word’ as a nominal will replace the first argument in the first argument position of the LS. Finally, for (d) computational adequacy – LING- CSA provides a computational framework of the RRG linking system and embedding an extension of speech act constructions.

To summarise this implemented RRG Language Model is fit for purpose as a linguistic engine for the LING-CSA framework (Panesar, 2017).

## 4.2 Agent Cognitive Model evaluations

LING-CSA's second phase of the Agent Cognitive Model has two inner models: the Agent Knowledge Model (AKM) and the Agent Planning Model (APM). The ACM composes the agent environment with a knowledge base, and takes on an input as a speech act performative (SAP) – a message with a propositional content reflecting the message send by the user. The elements of the speech act performative (SAP) will be used by various functions of the agent querying and planning process to generate a response. Figure 12 presents the class diagram for the ACM, linking with the RRG language model.

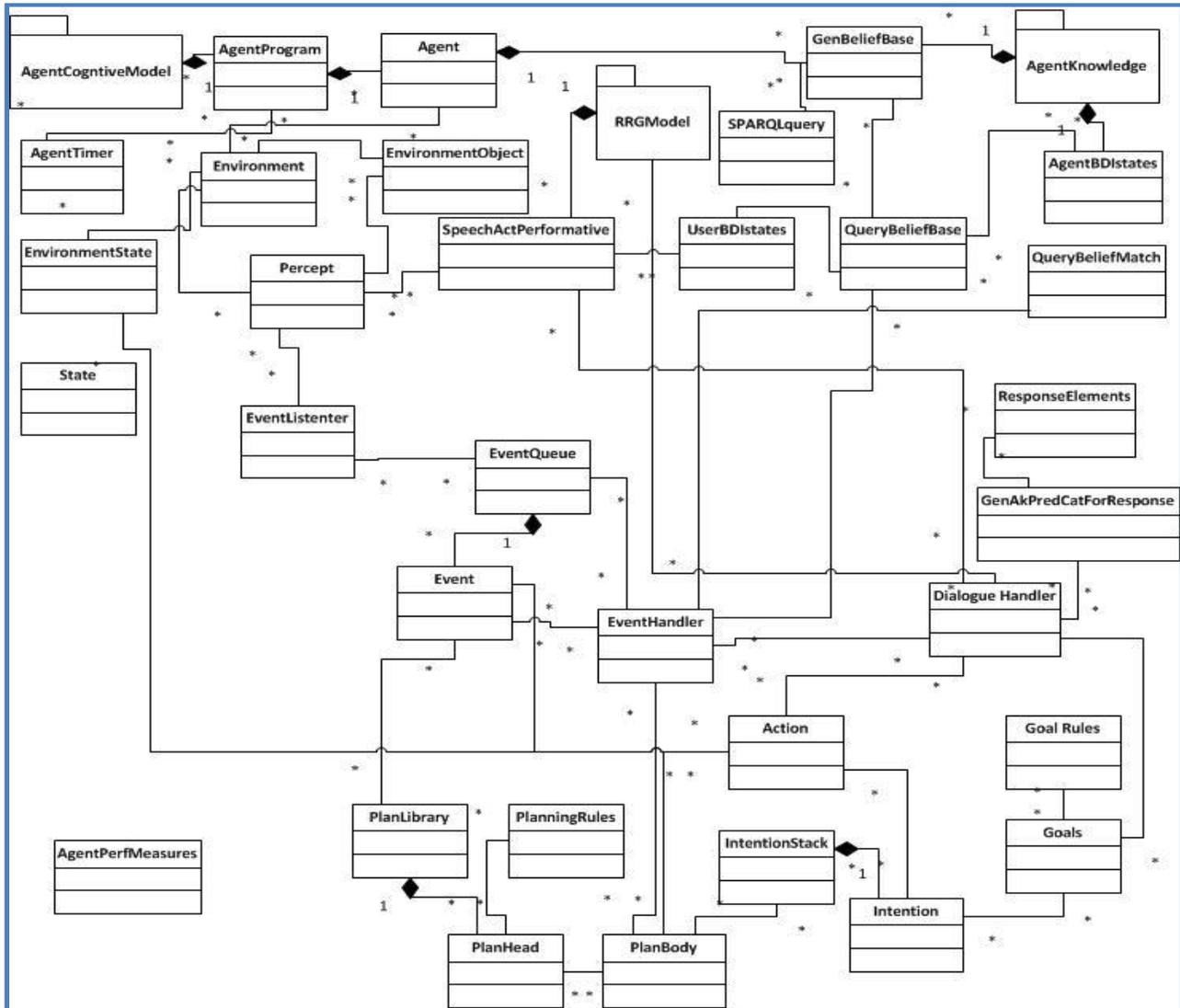


Figure 12 - Class Diagram of the Agent Cognitive Model

### 4.2.1 Knowledge Model evaluations

A standard known as the COGXML (Conceptual Graphs Extensible Markup Language) facilitates the exchange of graphs, and the connection of different knowledge systems to encode and decode CGs. CGs can be easily translated to RDFS (Yao and Eitzkorn, 2006), and its evolution OWL (Horrocks et al., 2005) applied to the Semantic Web space. This is achieved via COGUI which is able to export projects to RDF with RDF/XML, whereby only type hierarchies and facts are exported, but rules, constraints and type disjunctions (banned types) are ignored (GraphIK, 2012). Rules, banned types and constraints were successfully tested in COGUI.

In the first instance, the exported RDF/XML file is independently checked with a W3C RDF Validator, with success of a range of RDF triples. As the rules, constraints and banned types are ignored the OWL ontology rules will have to be deployed. In this section, criteria 5 and 6 are reiterated:

- Criteria 5 - Could the system demonstrate the agent BDI and knowledge representation?
- Criteria 6 - Could the system represent the user’s BDI states?

4.2.2.1 Criteria 5 – Agent BDI and knowledge representation

CGs created in COGUI are serialised into RDF/XML output form and parsed into RDF triples to form the agent’s knowledge base. Figure 13 and Table 3 demonstrate these two stages.

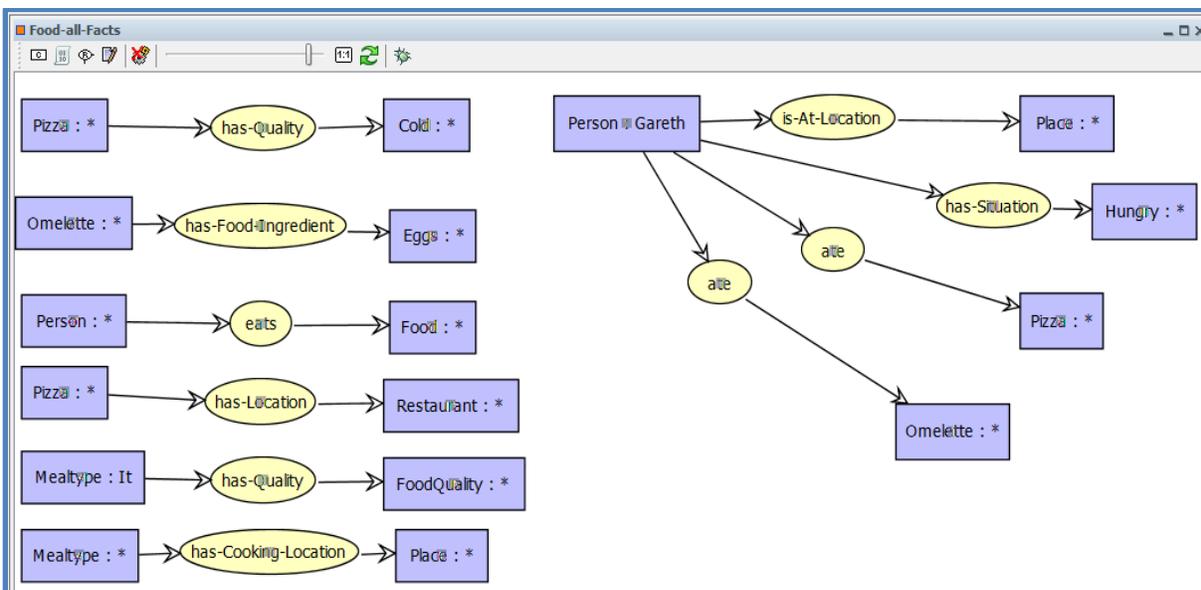


Figure 13 -COGUI - a set of facts for the food and cooking domain

Table 3 - An extract of the RDF triples -forming the agent's KB

No	Subject	Predicate	Object
1	<a href="http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b">http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2000/01/rdf-schema#Class">http://www.w3.org/2000/01/rdf-schema#Class</a>
2	<a href="http://www.lirmm.fr/cogui#ct_fdc6d7d0-1314-4fb7-8428-51e122953250">http://www.lirmm.fr/cogui#ct_fdc6d7d0-1314-4fb7-8428-51e122953250</a>	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">http://www.w3.org/1999/02/22-rdf-syntax-ns#type</a>	<a href="http://www.w3.org/2000/01/rdf-schema#Class">http://www.w3.org/2000/01/rdf-schema#Class</a>
3	<a href="http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b">http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b</a>	<a href="http://www.w3.org/2000/01/rdf-schema#subClassOf">http://www.w3.org/2000/01/rdf-schema#subClassOf</a>	<a href="http://www.lirmm.fr/cogui#ct_fdc6d7d0-1314-4fb7-8428-51e122953250">http://www.lirmm.fr/cogui#ct_fdc6d7d0-1314-4fb7-8428-51e122953250</a>
4	<a href="http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b">http://www.lirmm.fr/cogui#ct_ad452f18-e654-4ae6-b3a1-b7320616283b</a>	<a href="http://www.w3.org/2000/01/rdf-schema#label">http://www.w3.org/2000/01/rdf-schema#label</a>	"Breakfast"@en

In summary this criteria has been achieved, as RDF triples can be queried for specific facts and rules using a SPARQL query, based on the subject-predicate-object (S-P-O) concept.

4.2.2.2 Criteria 6 – User’s BDI states

The SAP is a message format to the Agent environment, and once listened and acknowledged this message is saved. As the SAP has been generated as an output from the RRG Language Model, the data structure and the algorithms to formulate the original SAC are available for reuse. The SAP has seventeen attributes which are each saved in their respective data structures in the agent environment. Here the propositional content ‘Gareth ate the pizza’ and the LS elements are used to establish the user’s BDI states, as identified in Table 4.

Table 4 -An example of the user's BDI states

Speech Act Performative attributes	Example
Propositional content	'Gareth ate the pizza'
Logical Structure	<IF>ASS<TNS>PT do'(gareth,[eat'(gareth,pizza)]) & INGR consumed'(pizza)
PSA, actor and undergeor	Gareth, Gareth and pizza
User's BDI states	Belief state – Gareth Desire – 'eat' Intention – 'consume pizza'

4.2.2.3 Assess the CSA's knowledge representation

CGs was selected for their effective representation of NL text (Sowa, 2009). The serialisation to RDF/XML format and further parsing to RDF triples was achieved due to their common graph theory. Further the RDF triples facilitate SPARQL querying (S-P-O) format and hence is a sound choice, and has been achieved. The computational solution of the user's BDI states has not been demonstrated, as on one hand the ground work has been established with the RRG Language Model, particularly the SAC and the LS, and the accessibility and reuse of their data structures and algorithms. On the other hand, a major technical issue presented an obstacle to the further use of the user's BDI states, and querying of NL text against the RDF triples. This issue has had an impact on the future development of LING-CSA (Panesar, 2017), and explained in the following section.

Morphology involves NL syntax composed typically of sophisticated words, phrases and clauses formed together to construct meaningful sentences, governed by grammatical rules. Similarly, multiple words combined can form constructs that can form a subject, predicate or object (direct or indirect). Comparatively, ontology and domain knowledge have a syntax limited to S-V-O sentences, for example, 'Mechanic repairs equipment' is called a triple, with passive and active statements, but having no ontological grammar rules to create new 'multi-word construct' which will take that form of a class and property. For example, in NL the separate words, lacrosse, sport, and equipment, can form the phrase 'lacrosse sports equipment', comprising constituent words and their individual meaning. In ontology, this will be conflated to sports as a single canonical form.

Taking a further NL sentence example 'An action is the subject of the action functional association' as illustrated in Figure 7, it is syntactically conflated to SVO triple 'action', is-the-subject-of, 'action-functional association'. These syntactic limitations give rise to syntax conflation which provides a serious of impacts such as (1) SVO triples are void of morphology (SVO do not vary in meaning – for instance tense and number); (2) a complex NL syntax will be reduced to a set of simple SVO triples; (3) the SVO construct will have one and only one named element; (4) The lack of function words makes to its difficult to connect different SVO triples to form compound sentences or form a discourse context.

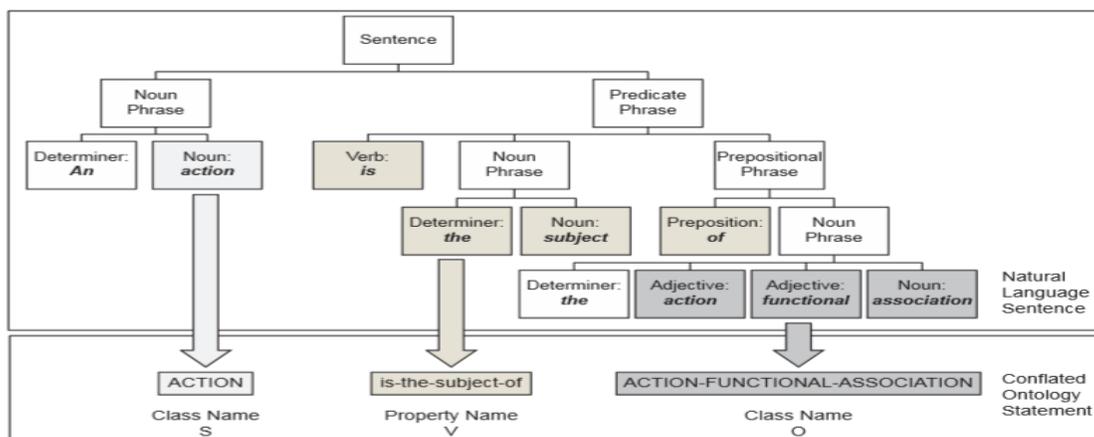


Figure 14 Complex NL sentence to a conflated ontology statement

Here the NL syntax is flattened into three ontology strings, whereby the original NL sentence lost its internal constituent phrase structure (phrase-splitting whereby the prepositions and predicative nominative) are completed into the relation name, and the object of the proposition is conflated to a class name.

The original NL sentence is conflated to a simple SVO triple with three ontology words. Linguistically, this demonstrates a restriction of noun-verb-noun structure with morphology, lexical and syntax limitations adding to the cascading conflation (Bimson and Hull, 2016) as identified in Figure 14. Languages are semantically more expressive than ontologies which mean that there is a major challenge, in translating NL semantic to ontology semantics, leading to a significant meaning loss, and characterised in terms of morphology, lexical and syntactic – semantic gaps.

The next cause of action is to reduce this semantic gap, by “building a lexical bridge (LB)” (Bimson et al., 2016) between the NL semantic and ontology semantics, with an aim to capture more of the meaning, by attempting to ‘lexicalize the ontology’. This will entail, parsing the ontology literals (class and property string names) to more lexicalised items, thus forming an ontology lexicon (OL). A further step can be taken to check the semantic equivalence of the ontology lexical meaning with a range of NL semantics. This OL can be used to identify and eliminate any redundant knowledge, such as RDF assertions, to find any hidden lexical meaning.

### 5.3 Considerations of a lexical bridge for the Agent Cognitive Model

This lexical bridge (LB) will form a component of the ACM of the LING-CSA specifically as an extension to the knowledge model. The initial KB is to be expressed in CGs and serialised into RDF/XML and then exported into RDF triples for further representation and reasoning. The LB will support this representation, to allow querying and facilitate any reasoning activities. Figure 15 below outlines the LB demonstrating the semantic crosswalk form the NL semantics to the RDF semantics based on parsing the RDF triples into NL words.

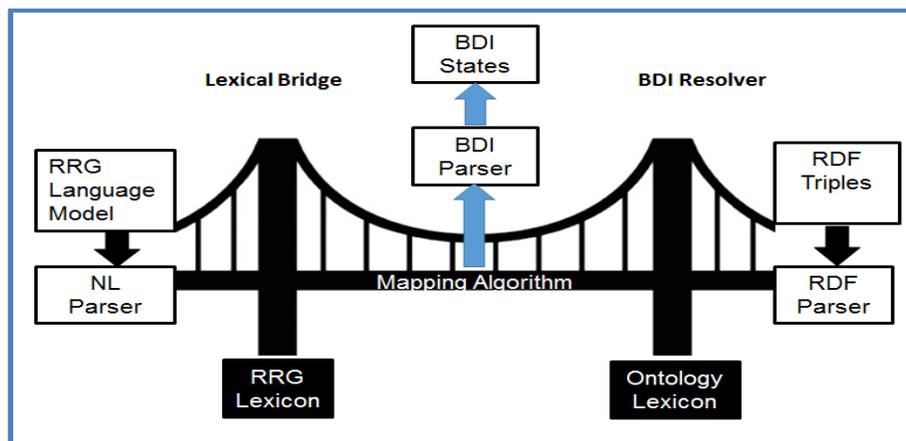


Figure 15 Lexical Bridge for the CSA's belief base + BDI Parser to resolve the agent's BDI states

This LB will require: (1) RDF parser to parse the RDF Triples into NL words, against the common RRG Lexicon (or to generate a RDF lexicon). (2) To reuse an existing NL parser (available in RRG Model) to algorithmically parse NL text into words, parts of speech from Phase 1 – RRG model using the RRG lexicon (3) an ontology-to-NL mapping-algorithm – comparison of the RDF triples with the NL words.

The current design framework of the Agent Cognitive Model in Figure 4, presented no opportunity to compute the agent's BDI states, as they are dependent on the KR of the agent's belief base, thus we need a sound model to query the belief base using NL text. Hence our dilemma is linked to the previous issue and resolution, in that once the LB and mapping algorithm computations are done, the resultant output together with a BDI parser, will support the deduction of the agent's BDI states as illustrated in Figure 4. A range of pseudo coding steps for the LB are presented in Table 5.

Table 5 Pseudo coding steps are identified for the Lexical Bridge

<b>Lexical Bridge (LB) Algorithm</b>
1. Translate the RDF Triples class or property literals into its component lexical items (using RDF Parser) - to determine the constituent words and POS.
2. Standard parsing of semantic equivalent NL text into lexical items (using the NL Parser invoking the RRG Lexicon)
3. Comparison of the RDF lexical components to NL text to determine their semantic equivalence.

Table 6 RDF object - semantic equivalent NL words in the Lexical Bridge – Step 1

<b>RDF triple object property</b>	<b>Natural Language paraphrase</b>
Has-Food-Ingredient	Food contains MealTypes
	Gareth has eaten
	Egg is an ingredient of omelette

Continue parsing with all RDF object property literals, to complete the lexicalisation process, invoking the RRG Lexicon as identified in Table 7.

Table 7 Parse with corresponding natural language texts – Step 2

<b>RDF concept</b>	<b>Word #</b>	<b>Word</b>	<b>POS tag</b>	<b>POS</b>	<b>Base verb</b>
Has-Food-Ingredient	1	has			has
	2	food	N	Noun	food
	3	ingredient	N	Noun	ingredients
Food contains mealtypes	1	food	N	Noun	food
	2	contains	V	Transitive verb	contains
	3	mealtype	N	Noun, plural	mealtype
Egg is an ingredient of omelette	1	egg	N	Noun	egg
	2	is	VBE		is
	3	an	ART	Indefinite article	an
	4	ingredient	NOUN	Noun	ingredient
	5	of	PRP	Proposition	of
	6	omelette	NOUN	Noun	omelette

Step 3 from Table 5 - requires an algorithm to compare the lexical semantics in the RDF form to the Natural Language Semantics – focusing on the RDF parse trees, and comparing the RDF words (RW) to text words (TW) in the text parse trees, as demonstrated in Table 7.. This algorithm will invoke Description Logic (DL) (objects having features), which is in effect a mapping algorithm of the words using a RDF equivalency algorithm, which is based on standard graph algorithms (Dijkstra, 1959). This LB could potentially provide significant benefits to the deployed CSA semantic solutions: (1) RDF redundancy identification and prevention; (2) improving semantic search in text; (3) Bridging the semantic gap between ontology semantic and NL semantics (4) semantic comparison analysis of ontologies. Simultaneously, the LB demonstrates aspects of simplicity as well as inherent complexity to support our semantic gap goal.

#### 4.3 Criteria (7), (8) and (9) – Agent Planning Model and Dialogue Model

The Agent Planning Model (APM) and Agent Dialogue Model (ADM) are interlinked in the Agent Cognitive Model (ACM), at the implementation level. To generate a response, the Dialogue Model is invoked, which further invoke the language model. The dialogue model has a dual role in the analysis of the utterance and in the generation of the response. In this section we address criteria 3(f-h):

- Criteria 7 – Could the system query the knowledge base for a fact (from the speech act performative)
- Criteria 8 - Could the system devise an appropriate plan based on the BDI states?
- Criteria 9 - Could the system generate a grammatically correct response in RRG based on the agent’s knowledge?

The remaining three criteria are examined together due to their dependencies. To start, the ACM interconnectivity is outlined. Figure 12 illustrates how the RRG Language Model and Knowledge Model interface with the Agent Cognitive Model. Further it demonstrates the dialogue handling process to support the generation of a response. Conceptually the Dialogue Model is embedded in the implementation of the ACM and further the output from the dialogue handling process, will be a response sent back to the RRG Language Model to refine to a grammatically correct response. Figure 16 presents a class diagram of Java based classes in Eclipse IDE platform demonstrating our ground work which evidences (a) Workings of the agent environment; (b) Perceiving input of a speech act performative (SAP) message; (c) Processing of the user’s BDI states; (d) Querying of the belief base; (e) Querying of user’s BDI states against the agent’s BDI states; (f) Event handling process; (g) Process for the selection of plans; (h) Process to carry out a plan based on the intention and goals; (i) Dialogue handling process to work towards a response invoking the construction of the predicate class category (type of verb – transitive, intransitive, modal) of the response.

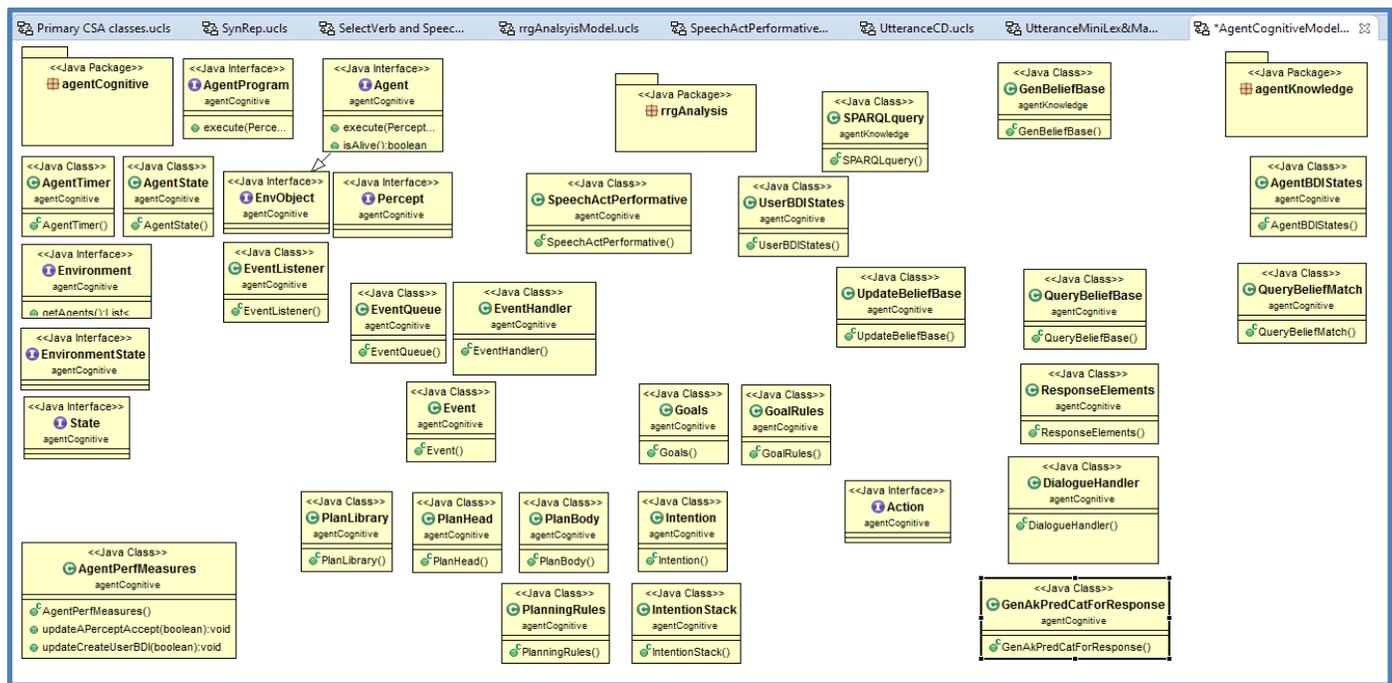


Figure 16 - Class Diagram - Agent Cognitive Model - Classes in Eclipse

Criteria (f) – querying of the agent’s KB against the content (for example, ‘who ate the pizza’) from the SAP message, posed a significant technical challenge for the continued work on the Agent Cognitive Model. To summarise, the propositional content is in a NL semantic format, and the RDF triples are in an ontological semantic format, and they do not map together, and present a semantic gap. This technical issue posed a massive challenge similarly identified by (Bimson and Hull, 2016). This technical issue has been explored and an abstract conceptual framework of a lexical bridge utilizing the RRG Lexicon with a mapping algorithm has been proposed to solve this problem, and further facilitate the querying of the KB.

Criteria (g) relating to the representation the agent’s BDI states was also an issue, and was explored by extending the lexical bridge with a BDI parser.

Criteria (h) will be achieved by adding a next layer of classes to reflect the technical solution of the lexical bridge, mapping algorithm and the BDI parser. The dialogue model has a dual role in the analysis of the utterance and in the generation of the response.

In summary, the groundwork for the agent environment and planning model has been demonstrated, with a working knowledge model, and a proposed conceptual framework of lexical bridge solution.

## 6 Evaluation conclusions and recommendations

To iterate, the goal of this linguistically centred CSA (LING-CSA) was to investigate the accuracy of the interpretation of meaning to provide a realistic dialogue to support the human-to-computer communication. LING-CSA was multi-disciplinary invoking natural language processing, computational linguistics, knowledge representation, functional linguistics, agent thinking and artificial intelligence, as demonstrated in the conceptual and operational framework of LING-CSA. Subsequently, by reviewing related works, a customised evaluation methodology and testing framework was selected, driven by linguistic theory goals, phase models, RRG grammatical structures, the specific speech act at play and agent environment and the functions of a conversational software agent. A set of evaluation criteria was devised to exhaustively assess where the requirements of linguistically centred CSA were being met.

### 6.1 RRG Model evaluations

The RRG phase model evaluations identify that RRG is fit-for-purpose as the linguistic engine for LING-CSA, and has a range of improvements and recommendations, (Panesar, 2017). For this paper, a subset is presented. However, one obstacle – *Example 1* identified is the workings of two logical structures (LSs) and three macroroles. This issue leads to problems with irregular verbs and exceptional transitivity (number of macroroles is greater than two), and further Van Valin Jr (2005) justifies and assumes a third macrorole for two reasons: (1) that the third argument is that of a ditransitive verb, (2) or to accommodate a dative case assignment. In some cases, there are some complex LSs, for the verbs give, present, show, teach, load and put. In mapping the semantic representation, with a non-macrorole assignment, the AUH and ranking is applied but, there is further dilemma as to how the non-predicating elements will behave. The proposition assignment rules for English deal with this situation, where there is a ‘.....to...’, ‘.....from...’, or ‘with’ is achieved by propositional stranding, and central to extending the current prototype of the RRG Language Model. *Example 2* - To refine the computational rules for anaphoric resolutions, for example, for other pronouns such as, 'my, your, his/her, ours, your, their'. The current solution is simple and effective, and can be extended. *Example 3* - as part of the semantic-to-syntax algorithm, the WH-word is correctly substituted for a noun phrase, but further work with the propositional stranding rules and argument (lexical features) is necessary, as there is links with *Example 1*.

As noted a snapshot of key recommendations are included: (1) consideration of the requirements for complex sentences (more than one clause) with the extended use of procedures for the linking system; (2) support for multi-lingual (additional lexicons) such as the Spanish, co-operating with RRG scholars; (3) consideration of other SA categories such as commissives and emotives, such as, for the automatic analysis of social media tweets; (4) to include superlative adjectives and adverbs RRG lexicon to enforce greater coverage.

### 6.3 Agent Cognitive Model recommendations

The BDI architecture is based on the ‘intentional stance’ which is the highest level of abstraction of human reasoning where the agent will do what it believes are its goals. There are two main areas for future research, based on this evaluation. Firstly, is to reduce the gap between knowledge and language, in the agent space. Here, language is in the context of a RRG linguistic representation against the RDF Triples representing the agent belief base posed the key obstacle of interoperability of low level mapping and representation. This has been termed the ‘lexical bridge’ (LB). Central to this LB is the re-use of the RRG lexicon and NL parser. The derivation of a mapping algorithm from OWL to a NLP representation is recommended, specifically invoking the RRG lexicon and RDF parser. This mapping algorithm will be interfaced with a further model known as the BDI Resolution composing a BDI parser motivated by description logics (DL) to derive the BDI states of the agent. This step will form an extra step to the pre-agent steps of ACM. The algorithm’s output representation, will further invoke the rest of the ACM steps of querying the knowledge base for a specific belief, creating planning schemes, selecting a plan, manipulation of the appropriate intention, to create a grammatical correct response using the RRG linguistic engine and the extended speech act constructions.

A second recommendation is based on the evaluations to the ACM is for an extended design framework to enable a multi-agent environment where there are both BDI logics and plan-based speech acts to formulate the dialogue response, and thus will require a more refined evaluation methodology and framework.

## References

- ABRAN, A., KHELIFI, A., SURYN, W. & SEFFAH, A. 2003. Consolidating the ISO usability models. . *In proceedings of 11th International Software Quality Management Conference, 2003*, pp. 23-25.
- BIMSON, K. D. & HULL, R. D. 2016. Unnatural Language Processing: Characterizing the Challenges in Translating Natural Language Semantics into Ontology Semantics. *In: WORKMAN, M. (ed.) Semantic Web: Implications for Technologies and Business Practices*. Cham: Springer International Publishing.
- BIMSON, K. D., HULL, R. D. & NIETEN, D. 2016. The Lexical Bridge: A Methodology for Bridging the Semantic Gaps between a Natural Language and an Ontology. *In: WORKMAN, M. (ed.) Semantic Web: Implications for Technologies and Business Practices*. Cham: Springer International Publishing.
- BRATMAN, M. 1987. *Intention, plans, and practical reason*, Cambridge, Harvard University Press.
- COHEN, P. R. & LEVESQUE, H. J. 1988. Rational interaction as the basis for communication. DTIC Document.
- COHEN, P. R. & LEVESQUE, H. J. 1990. Intention is choice with commitment. *Artificial Intelligence*, 42, 213-261.
- DIJKSTRA, E. W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1, 269-271.
- FRANCOIS, J. 2014. Functional views on the genesis and development of grammar *25th European Systemic Functional Linguistics Conference Paris*.
- GRAPHIK. 2012. *CoGui (Conceptual Graphs Graphical User Interface) User Guide* [Online]. Available: <http://www.lirmm.fr/cogui/newdoc/cogui-index.html> [Accessed 12 March 2014].
- HORROCKS, I., PATEL-SCHNEIDER, P. F., BECHHOFFER, S. & TSARKOV, D. 2005. OWL rules: A proposal and prototype implementation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3, 23-40.
- KAMP, H., VAN GENABITH, J. & REYLE, U. 2011. Discourse representation theory. *Handbook of philosophical logic*. Springer.
- LESTER, J., BRANTING, K. & MOTT, B. 2004. Conversational agents. *The Practical Handbook of Internet Computing*.
- MAO, X., SANSONNET, J.-P. & LI, L. Textual Conversation Agent for Enhancing Attraction in E-Learning. *Proceedings of International Conference on Computer Science and Information Technology, 2012*. 49.
- NOLAN, B. 2014. Constructions as grammatical objects : A case study of prepositional ditransitive construction in Modern Irish. *In: NOLAN, B. & DIEDRICHSEN, E. (eds.) Linking Constructions into Functional Linguistics: The role of constructions in grammar*. Amsterdam/Philadelphia: John Benjamins Publishing Company.
- PANESAR, K. 2017. *A linguistically centred text-based conversational software agent*. Unpublished PhD thesis, Leeds Beckett University.
- PEREZ-MARIN, D., PASCUAL-NIETO, I. & GLOBAL, I. 2011. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*, Information Science Reference.
- RADZIWILL, N. M. & BENTON, M. C. 2017. Evaluating Quality of Chatbots and Intelligent Conversational Agents. *arXiv preprint arXiv:1704.04579*.
- RAO, A. S. & GEORGEFF, M. P. BDI Agents - From Theory to Practice. *ICMAS, 1995*. 312-319.
- RUSSELL, S. & NORVIG, P. 2016. *Artificial Intelligence: A Modern Approach - Global Edition*, Pearson.
- SEARLE, J. R. 1969. *Speech acts: An essay in the philosophy of language*, Cambridge, CUP.
- SEARLE, J. R. 1983. *Intentionality: An essay in the philosophy of mind*, CUP.
- STALNAKER, R. 2002. Common ground. *Linguistics and philosophy*, 25, 701-721.
- TURING, A. M. 1950. Computing machinery and intelligence. *Mind*, 433-460.
- VAN VALIN JR, R. D. 2005. *Exploring the syntax-semantics interface*, CUP.
- WEIZENBAUM, J. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 36-45.
- WOOLDRIDGE, M. 2013. Intelligent Agents. *In: WEISS, G. (ed.) Multiagent Systems*. 2nd edn. ed. USA: Massachusetts Institute of Technology.
- YAO, H. & ETZKORN, L. 2006. Automated conversion between different knowledge representation formats. *Knowledge-Based Systems*, 19, 404-412.