RaY

Research at the University of York St John

For more information please contact RaY at ray@yorksj.ac.uk

An Evaluation of a Linguistically Motivated Conversational Software Agent Framework

Kulvinder Panesar

York St University, York, United Kingdom

Corresponding author: k.panesar@yorksj.ac.uk

Abstract

This paper presents a critical evaluation framework for a linguistically motivated conversational software agent (CSA). The CSA prototype investigates the integration, intersection and interface of the language, knowledge, and speech act constructions (SAC) based on a grammatical object, and the sub-model of belief, desires and intention (BDI) 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 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), (2) an Agent Cognitive Model with two inner models: (a) a knowledge representation model, (b) a planning model underpinned by BDI concepts, intentionality and rational interaction, and (3) a dialogue model. The evaluation strategy for this Java-based prototype is 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. 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 solution.

Keywords: conversational software agents, natural language processing, speech act construction, knowledge representation, belief-desire and intentions, functional linguistics.

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 (KR) 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. 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, Sansonnet, and Li (2012) and Lester, Branting, and Mott (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) and response generator.

In the last decades, there has been a great evolution in the field of CSA enfolding three emerging trends of more sophisticated natural language processing (NLP) via improved parsing techniques, humanising of agents through language and their pervasiveness (Perez-Marin and Pascual-Nieto 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 a grammatical object (Nolan 2014) and rich lexicon.

This paper presents a critical evaluation framework for LING-CSA (Panesar 2017). The objectives of this paper are: (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) and a summary of the findings, (5) conclusions and recommendations.
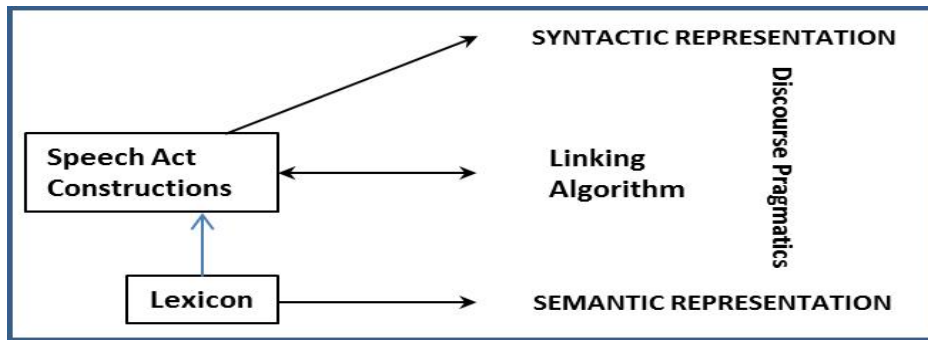
FIGURE 1. RE-ORGANISATION OF RRG FOR THE CSA.

2. CONCEPTUAL ARCHITECTURE OF THE CONVERSATIONAL SOFTWARE AGENT

LING-CSA constitutes three phase models as illustrated in Figure 2 which 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.
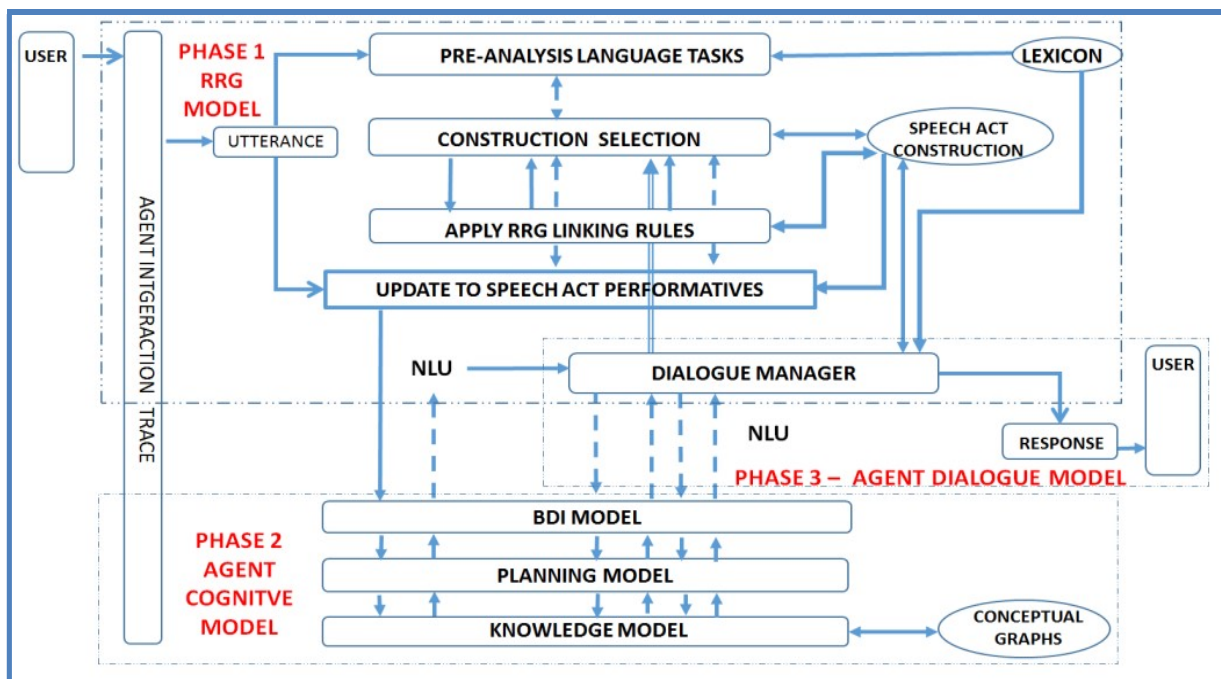
FIGURE 2. CONCEPTUAL ARCHITECTURE OF THE LING-CSA (PANESAR, 2017).

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.

Our LING-CSA is limited to the manipulation of the linking system for simple sentences (active or passive) accepting transitive, intransitive and ditransitive verbs/auxiliary verbs with variable word order flexibility following SVO (subject-verb-object), 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 via computational processes such as invoking the lexicon (Table 1). Here the logical structures have been partially expanded for readability in this paper and DNA refers to 'Does not apply'.

Lexical entry 1: ate								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
Verb	PST	DEF +/-	3	SG	M/F	DNA	ANIM	HUM
LOGICAL STRUCTURE (LS) : <tns:pst <do'(x, [eat'(x, y)]) & BECOME consumed' (y) >>								

Lexical entry 2: eat								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
VERB	PRS/ FUT	DEF+/-	3	SG	M/F	DNA	ANIM	HUM
LOGICAL STRUCTURE (LS) : <tns:prs <do'(x, [eat'(x,y)]) & BECOME consumed'(y) >> <tns:fut <do'(x, [eat'(x,y)]) & BECOME consumed'(y)>>								

Lexical entry 3: eating								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
VERBAL NOUN	PROG	DEF+/-	3	SG	M/F	DNA	ANIM	HUM
LOGICAL STRUCTURE (LS) : <tns:prs <asp:prog <do'(x, [eat'(x, y)]) & BECOME consumed' (y) >>								

Lexical entry 4: is								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
VERB BE	DNA	DEF+	DNA	DNA	DNA	DNA	DNA	DNA
LOGICAL STRUCTURE (LS) : be'(x,[pred'])								

Lexical entry 5: hungry								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
ADJECTIVE	DNA	DNA	DNA	DNA	M/F	DNA	ANIM	HUM
LOGICAL STRUCTURE (LS) : DNA								

Lexical entry 6: restaurant								
POS TYPE	VERB TENSE - ASPECT	DEF	PERSON TYPE	NUMBER	GENDER	CASE	ANIM	HUM
NOUN	DNA	DEF+/-	DNA	SG/PL	DNA	DNA	DNA	DNA
LOGICAL STRUCTURE (LS) : DNA								

TABLE 1. SNAPSHOT OF THE LEXICON - LEXICAL ENTRIES AND LEXEMES (DATA SOURCE 1).

Further Table 2, provides an overview of the Phase 1 – RRG model steps but will also support all three LING-CSA phases, and internal BDI manipulations (Panesar, 2017).

Step	PHASE		(TRACKING) DISPLAY
PHASE 1 - RRG LANGUAGE MODEL			
1	PHASE 1A - PRE-ANALYSIS	Parsing, Sentence and word tokenisation, check against the lexicon.	Display the utterance and various internal representations.
2	PHASE 1B – SAC SELECTION	Construction selection computations	Display construction selected
3	PHASE 1C – SYNTAX-TO-SEMANTICS	• Verb selection and voice • Retrieve the LS and verb from the lexicon • Assignment of arguments to the LS arguments • Semantic representation • Update the LS • Map the operator projection and focus projections • Actor and undergoer assignment • Select the morphosyntactic properties of the argument	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
4	PHASE 1E - SEMANTICS-TO-SYNTAX	• Verb selection and voice • Retrieve the LS and verb from the lexicon • Assignment of arguments to the LS arguments	Display the verb selection and voice topics Display the LS and verb from the lexicon Display the argument values
5	PHASE 1F – DIALOGUE MANAGER	• Ascertain whether there is any missing information • Refer to the stored discourse referents – if any, and process with the earlier Phase 1 steps	Display the specific elements that missing Display appropriately any saved discourse referents
6	PHASE 1G – CREATE SAP	• Attach the performative information to the Speech Act Construction	Display the performative type information

TABLE 2. OVERVIEW OF THE RRG STEPS (PANESAR, 2017).

The analysis of utterance/responses will be constrained to assertive and interrogative (WH-word) SAs. For example, two empty SACs are shown in Figure 3: (1) 'assertive: ATE', and 'interrogative: IS' are taken from a sample collection of 34, originally created by the researcher, after inspection of a range of common corpuses. The selected SAC, is updated with all the grammatical information. These include: (a) voice opposition, (b) macrorole, (c) pragmatic, (d) semantic, (e) lexical, (f) the PSA (privileged syntactic argument) - similar to the 'subject' of SVO (subject-verb-object), (g) the matching signature (matching POS tags of the utterance/response), (h) syntactic morphology (rules), (i) focus, and (j) semantic information such as tense and aspect.

ASSERTIVE:ATE RRG [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] RRG NONE RRG UTTINPUT RRG WKSPACE RRG DEFAULT ASSUMPTION (1ST NP = 'ACTOR') RRG NO PARTICULAR SPEC RRG NONE RRG CONTAINS A NOUN PHRASE BEFORE AND AFTER THE VERB RRG DEFAULT RRG TRUE/FALSE RRG ASSERTIVE RRG NARROW FOCUS ON THE ELEMENT RRG LOG STRUCTURE TO ADD

INTERROGATIVE:IS RRG [WH1 N VBE POSS], [WH3 VBE POSS N], [WH7 VBE DET N ADJ], [WH7 VBE DET N ADJ ADV], [WH3 N VBE ADJ] [WH3 N VBE PRP PN], [WH3 N VBE POSS] [WH1 VBE POSS ADJ N] [WH4 N VBE DEM] [WH1 VBE PRP N] [WH7 VBE N ADJ] [WH8 VBE DET N] RRG NONE RRG UTTINPUT RRG WKSPACE RRG DEFAULT ASSUMPTION (1ST NP = 'ACTOR') RRG NO PARTICULAR SPEC RRG NONE RRG CONTAINS A NOUN PHRASE BEFORE AND AFTER THE VERB RRG DEFAULT RRG TRUE/FALSE RRG INTERROGATIVE RRG NARROW FOCUS ON THE ELEMENT RRG LOG STRUCTURE TO ADD

FIGURE 3. EMPTY SPEECH ACT CONSTRUCTIONS (EXAMPLES).

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 facilitate word order for English (SVO), and to unpack the agreement features between elements of the sentence into a semantic representation (the logical structure (LS)) 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.

2.2. 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 this will 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 4. The behaviour of the agent is identified by the following basic loop where the 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.
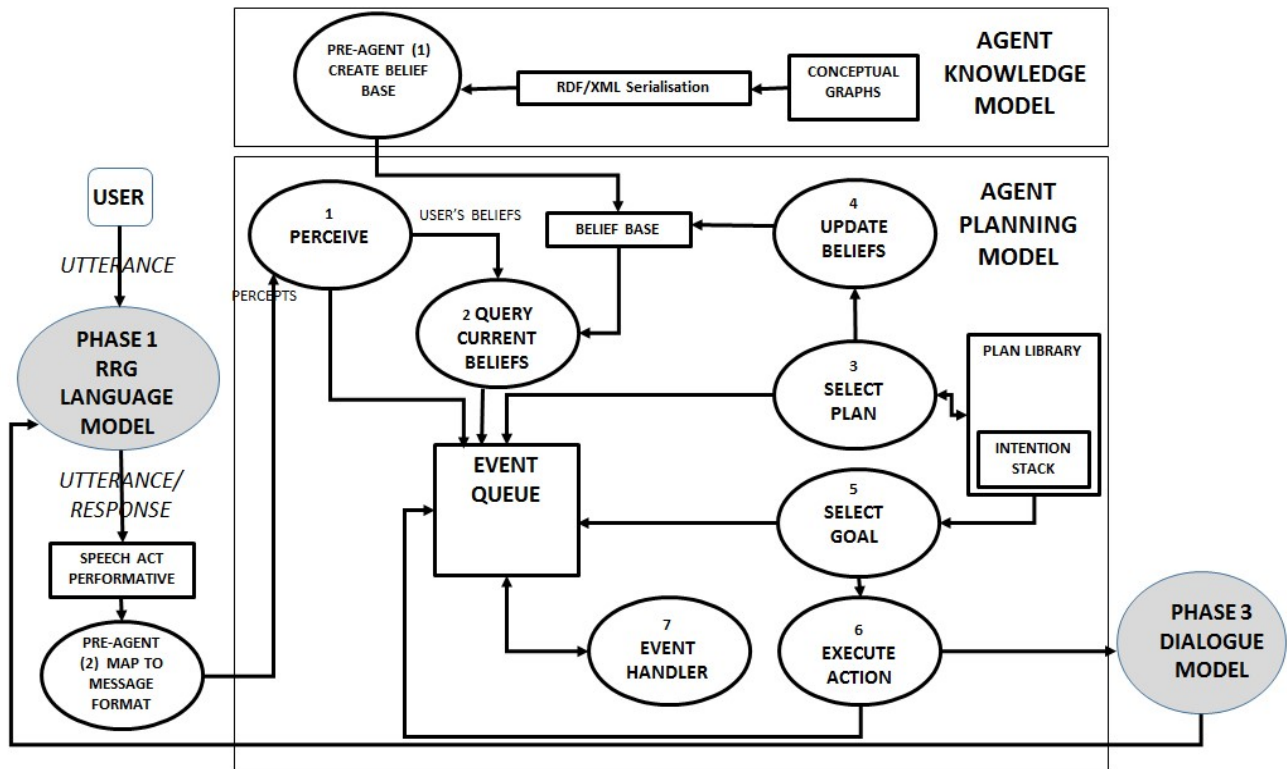

FIGURE 4. DESIGN FRAMEWORK - AGENT COGNITIVE MODEL (PANESAR, 2017).

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 (SAP) message back to the Agent;
g) Step (7) Agent Event Handler – handles different events such as perception (of messages), querying (belief base), updating (belief base), selecting (a plan, and further goal), to execute 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.

Bratman (1987) identifies the workings of how people make decisions and take action in the form of mental attitudes of beliefs, desires and intentions - goals (BDI) and in relation to rational interaction.

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, Van Genabith, and Reyle 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 SAC model, to formulate a grammatically correct response.

This single, goal-orientated conceptual CSA investigates the integration, intersection and interface of the language, knowledge, and speech act constructions (SAC), and the belief desires and intention (BDI) model and dialogue management (DM) with a central theme of natural language processing (NLP) as proof of concept.

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. This prototype constitutes three phase models: (1) a linguistic model based on a functional linguistic theory – RRG, (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 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 CSAs using keywords such as 'quality assurance', ' evaluation', and 'assessment' from dates 2016-2017. They 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, behaviour and accessibility. One interesting attribute was in question: whether or not a CSA should pass the Turing Test (Turing 1950).[1]

According to most researchers, and following the earliest conversational interfaces, responding

---

[1] Historically, CSAs link to the concept of intelligent machines and the question 'can machines think' (Turing 1950). Turing worked on a proposal of how to test this using the imitation game (Turing Test) and used dialogue testing, as means to work out whether or not the computer program was intelligent.

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.[2]

However, it is critical that the goal of the CSA should not aim to act human noted by 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.

The 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) KR logics (first order logics and graph theory) and important performance indicators for KRR as: i) representational adequacy, and 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) and a range of RRG tests applied at various internal representations to address the goals of RRG - explanatory, descriptive, cognitive and computational adequacies (Panesar, 2017). A set of evaluation criteria are identified in (2):

(2)
   a) Criteria 1 - Could the system present a mapping of the syntactic representation to a semantic representation, for a simple utterance?
   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 (SAP))
   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?

---

[2] This question has inspired exciting competitions (such as the yearly Loebner prize) to demonstrate human-like conversation with four times winner 'Mitsuku' Chatbot (Worswick 2018). Further investigating cognitive science - 'can machines think' and strong AI challenge, Searle (1980) introduces the Chinese room argument (experiment). It demonstrates that the computer program may very well look like it understands Chinese, but it only simulates that knowledge which is not a form of intelligence. Subsequently, there is a need to address the differences between a human brain and computer program in terms of having a mind, and being able to think and understand. This "understanding" implies both the possession of mental (intentional) states and the truth (validity and success) of these states (Searle 1980).

5. DETAILED EVALUATION OF THE CSA FRAMEWORK

This section evaluates the LING-CSA framework via the three phase models and their specific criteria and assessed with some supporting tests, 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. Here criteria (1-4) are individually assessed.

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. It highlights a range of utterances, addressing 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 SAC and their respective signature processed by the LING-CSA.

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

TABLE 3. SET OF UTTERANCES FOR TESTING RRG LINKING SYSTEM.

Figures 5 and 6 illustrate the utterance example 'they are hungry'. Firstly, the anaphoric resolutions. Next, tense, aspect, privileged syntactic argument (PSA), actor-undergoer hierarchy (AUH) and case marking are assigned, and LSC and LS representations generated.

```
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 argumnents
*******************************************
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 5. UTTERANCE - 'THEY ARE HUNGRY' – LOGICAL STRUCTURE AND MARKING.

```
Console ☒                          ■ ✖ ✕ | ▣ ▤ ▦ ▧ ▨ | ▸ ▢ ▾ ▢ ▾ ▭ ▯
<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 2
 ***************************************
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 6. UTTERANCE - 'THEY ARE HUNGRY' - LSC AND LS REPRESENTATIONS.

In reference to Table 3 and entry 8 (partially working) this is analsyed further in (3):

(3)

The utterance - 'Gareth ate the pizza in the restaurant'

The syntactic representation was generated correctly as:

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

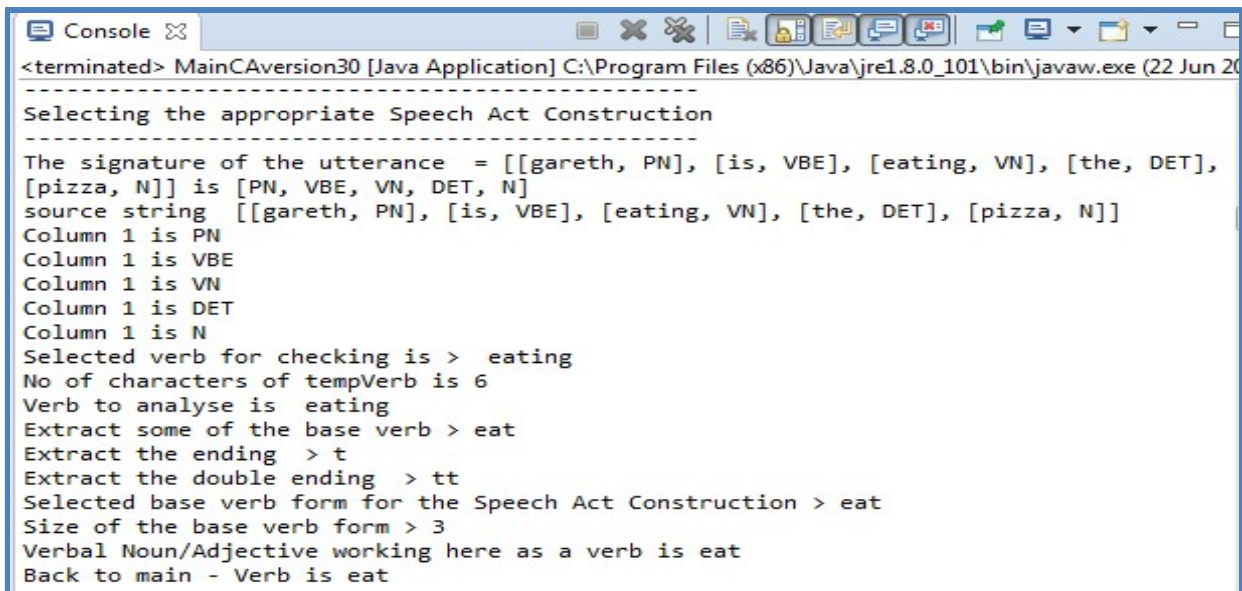The semantic representation was partially correct:

<IF>ASS<TNS>PST do'(gareth,[eat'(gareth,restaurant)]) & INGR

consumed'(restaurant)<IF>ASS<TNS>PST be-in'(gareth,restaurant)

Further discussion of the example utterance analysis is that the propositional assignment rules have been applied, and further refinement of logic rules is required, 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 customised stemming algorithm developed. 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 7 identifies the base verb.


```
Console ⌗                              ▣ ✖ ✖ | ▤ ▦ ▦ ▦ ▦ | ⬚ ▢ ▾ ▢ ▾ ▭ ▭
<terminated> MainCAversion30 [Java Application] C:\Program Files (x86)\Java\jre1.8.0_101\bin\javaw.exe (22 Jun 2(
--------------------------------------------------
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
```
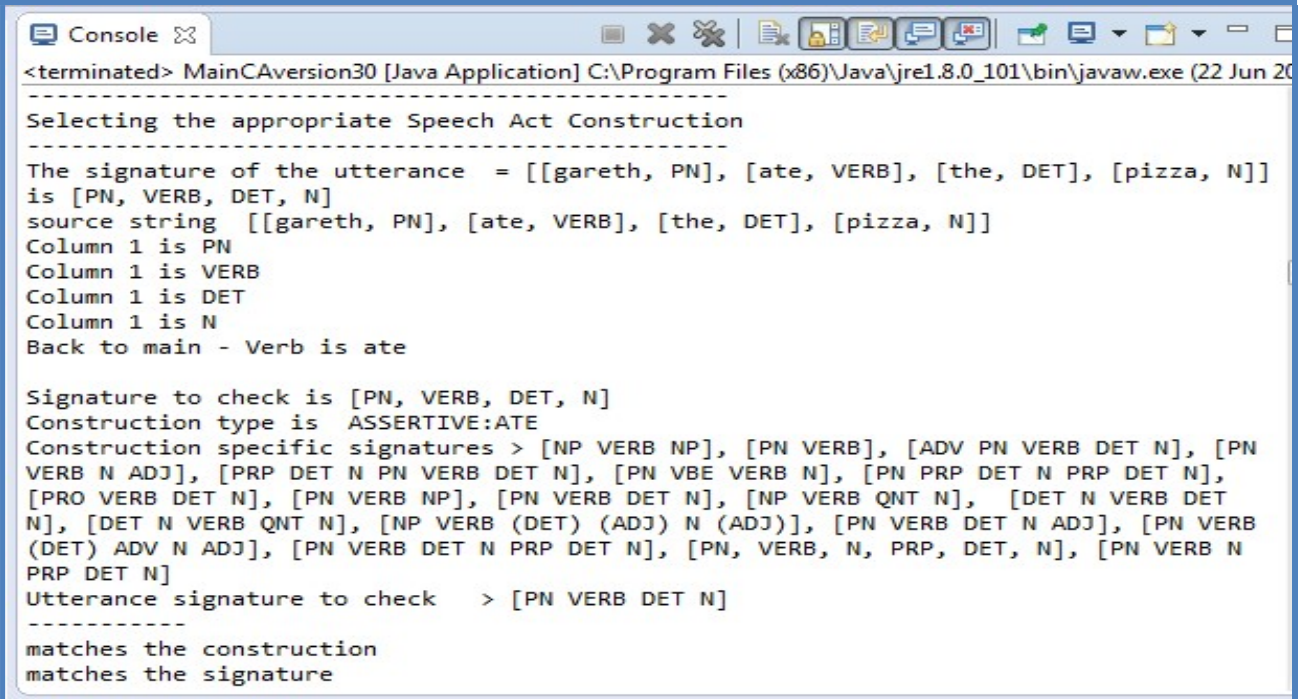
FIGURE 7. STEMMING FEATURE TO EXTRACT THE BASE VERB.

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 (SAT) 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 (IA) – what is to be done – which takes precedence. Here the IA 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 8. 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.


```
Console ✕                                                    ■ ✖ ✖ | ⟑ ▦ ▣ ▣ ▣ | ▱ ▱ ▾ ▱ ▾ ⊟
<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 8. SELECTION OF THE SPEECH ACT CONSTRUCTION.

5.1.4. 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 10 Dialogue Preparation (1) and (2). Here, it makes use of the current utterance and further extracts and saves the nominals for future utterances. The nominals are saved in an output file 'DiscourceReferents.txt' which is saved locally in the resource folder of the CSA program. This file contains a token with its associated POS tag. This information will be used by another utterance, via another execution of the CSA program.
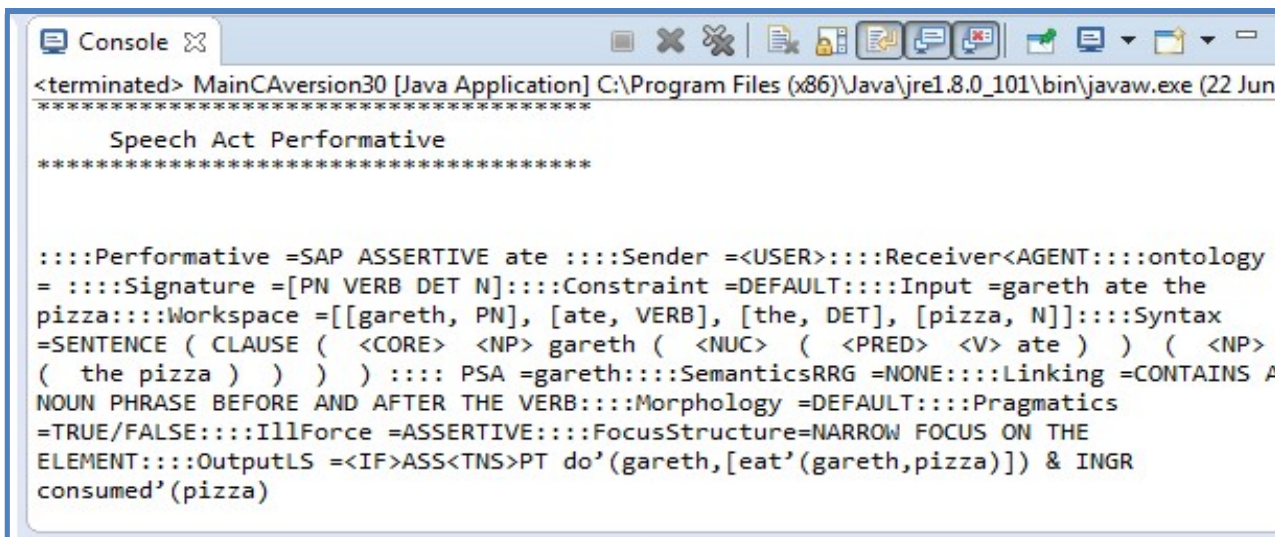
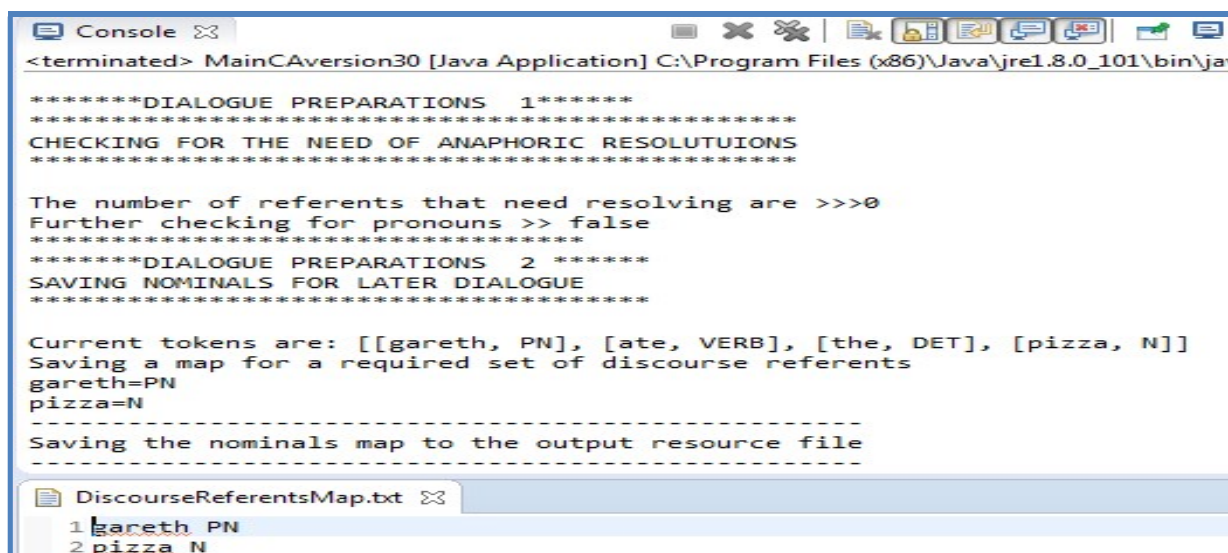**FIGURE 9. SPEECH ACT PERFORMATIVE – MESSAGE FORMAT > AGENT COGNITIVE MODEL.**

FIGURE 10. DIALOGUE PREPARATIONS (1) AND (2).

Taking the example, 'he is in the restaurant', Figure 11 illustrates the re-loading of these nominals into the micro-lexicon. The new utterance has a 'he' which is a pronoun that needs resolving, and its own discourse lexicon is generated as well.

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 11 and Figure 12 respectively. The new utterance generated is 'Gareth is in the restaurant' and thus continues 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.

FIGURE 11. DIALOGUE MANAGEMENT (3).

FIGURE 12. DIALOGUE MANAGEMENT (4).

5.1.5. Assess the CSA in terms of the goals of the linguistic theory

In the previous section we demonstrated that the criteria (1-4)) 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 four goals of a linguistic theory, and critique

them, with the main theoretical characteristics of RRG in relation to grammatical relations.

Firstly, 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). Secondly, 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. Thirdly, 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, computational adequacy – successful working of the RRG linking system and embedding an extension of speech act constructions (SAC)s. Summatively, the implemented RRG Language Model is fit for purpose as a linguistic engine (Panesar 2017).

5.2. Agent Cognitive Model evaluations

LING-CSA's second phase of the Agent Cognitive Model (ACM) has two inner models: the Agent Knowledge Model (AKM) and the Agent Planning Model (APM). Figure 13 presents the class diagram for the ACM, linking with the RRG language model.
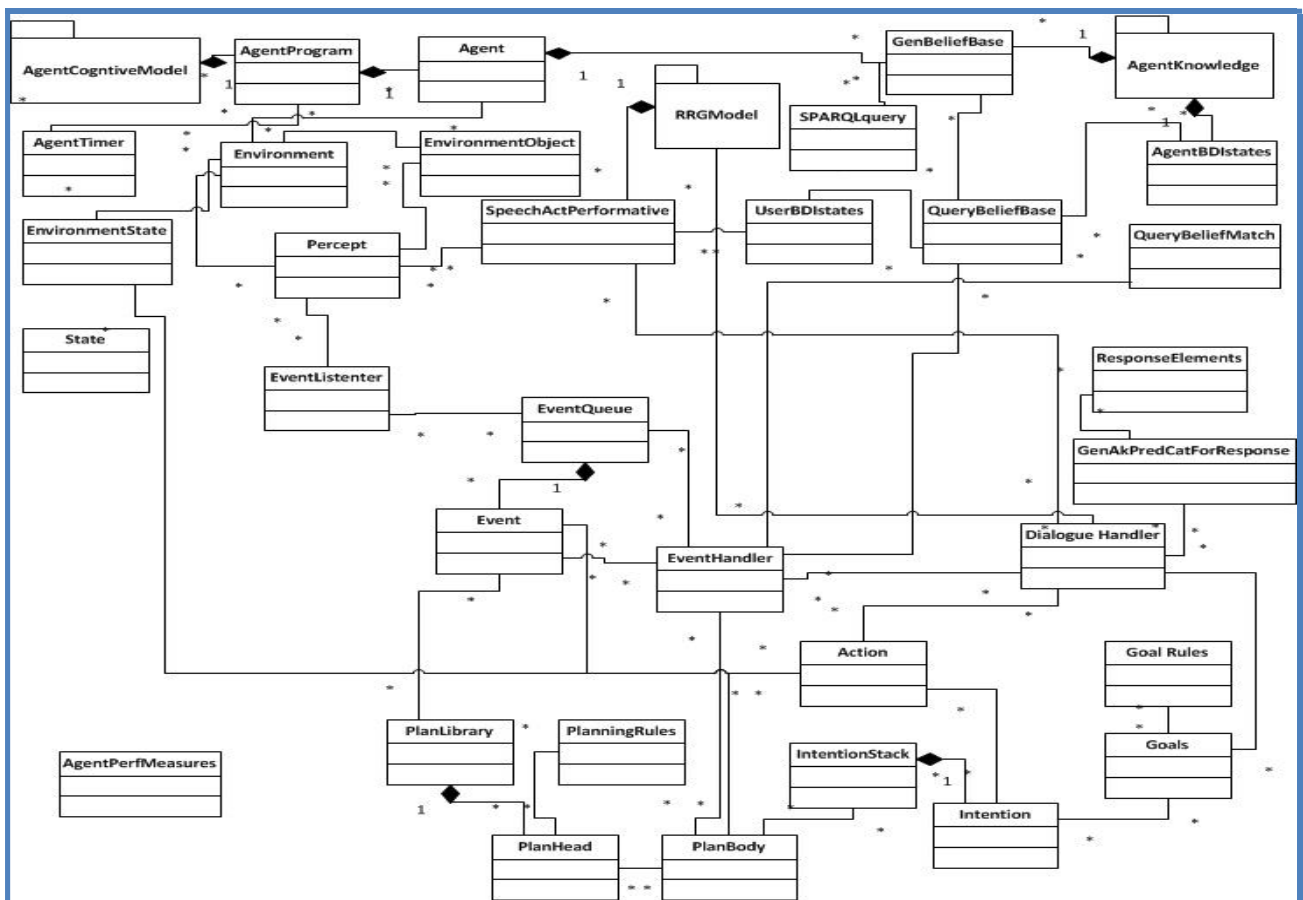

FIGURE 13. CLASS DIAGRAM OF THE AGENT COGNITIVE MODEL.

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 sent 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.

5.2.1. Knowledge Model evaluations

Conceptual Graphs (CGs) were 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. 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 Etzkorn 2006), and its evolution OWL (Horrocks et al. 2005) applied to the Semantic Web space. This is achieved via COGUI (Conceptual Graphs User Interface) 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 considered. Criteria 5: could the system demonstrate the agent BDI and knowledge representation? Criteria 6: could the system represent the user's BDI states?

5.2.2.1. 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 14 and Table 4 demonstrate these two stages.
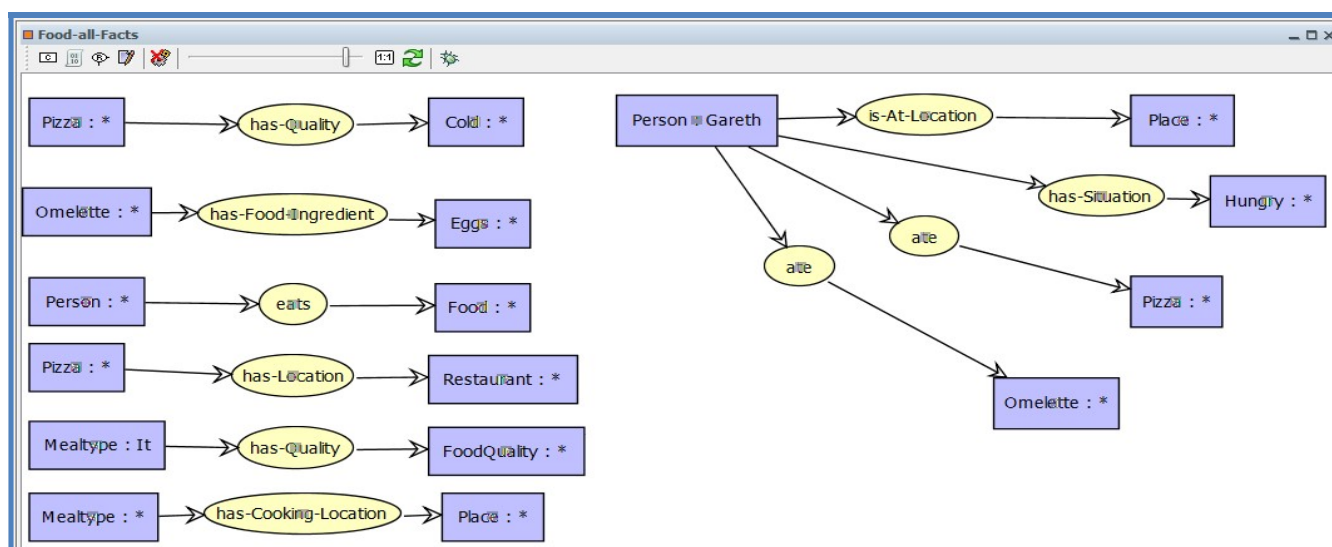

FIGURE 14. COGUI - A SET OF FACTS FOR THE FOOD AND COOKING DOMAIN.

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.

No	Subject	Predicate	Object
1	http://www.lirmm.fr/cogui#ct_ad452f18 -e654-4ae6-b3a1-b7320616283b	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
2	http://www.lirmm.fr/cogui#ct_fdc6d7d 0-1314-4fb7-8428-51e122953250	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2000/01/rdf-schema#Class
3	http://www.lirmm.fr/cogui#ct_ad452f18 -e654-4ae6-b3a1-b7320616283b	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.lirmm.fr/cogui#ct_fdc 6d7d0-1314-4fb7-8428-51e122953250
4	http://www.lirmm.fr/cogui#ct_ad452f18 -e654-4ae6-b3a1-b7320616283b	http://www.w3.org/2000/01/rdf-schema#label	"Breakfast"@en

TABLE 4. AN EXTRACT OF THE RDF TRIPLES -FORMING THE AGENT'S KB.

5.2.2.2. 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 algorithms to formulate the original SAC are available for reuse, and the SAP can be unpacked again. 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 5.

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'

TABLE 5. AN EXAMPLE OF THE USER'S BDI STATES.

5.2.2.3. 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 manipulation of the user's BDI states, and querying of NL text against the RDF triples. This issue has had an impact on the further development of LING-CSA (Panesar 2017), and is 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.

However, it has no ontological grammar rules to create new 'multi-word construct' which will take the 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 15, 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 it difficult to connect different SVO triples to form compound sentences or form a discourse context.

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 15.
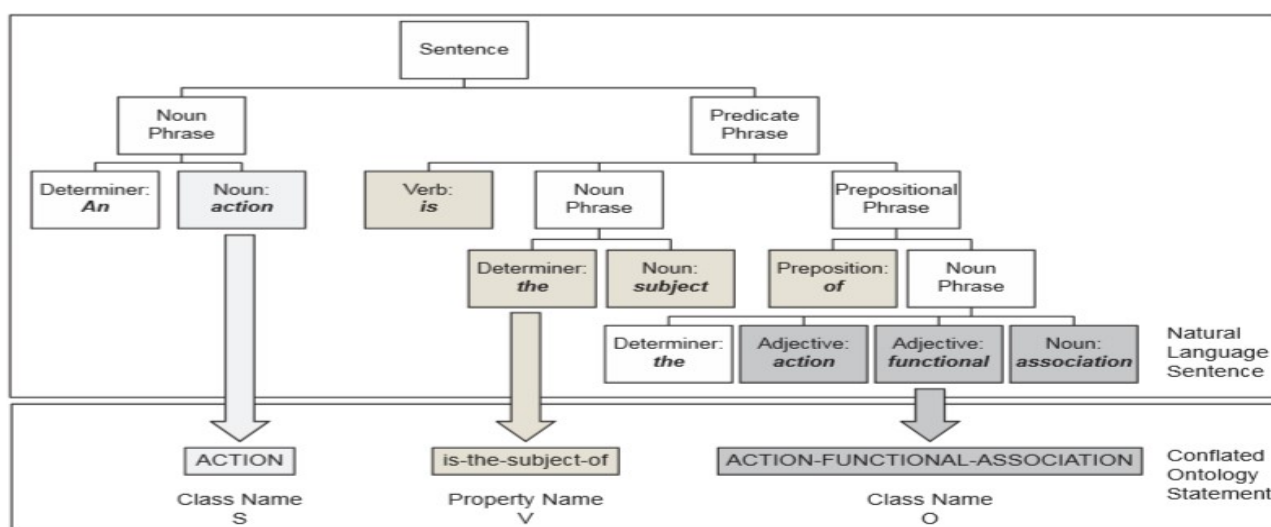
FIGURE 15. COMPLEX NL SENTENCE TO A CONFLATED ONTOLOGY STATEMENT.

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, Hull, and Nieten 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 16 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. LB comprises:

(4)

a) A RDF parser to parse the RDF Triples into NL words, against the common RRG Lexicon (or to generate a RDF lexicon).
b) To reuse an existing NL parser (RRG Parser) to algorithmically parse NL text into words, parts of speech from Phase 1 – RRG model using the RRG lexicon
c) An ontology-to-NL mapping-algorithm – comparing RDF triples with the NL words.
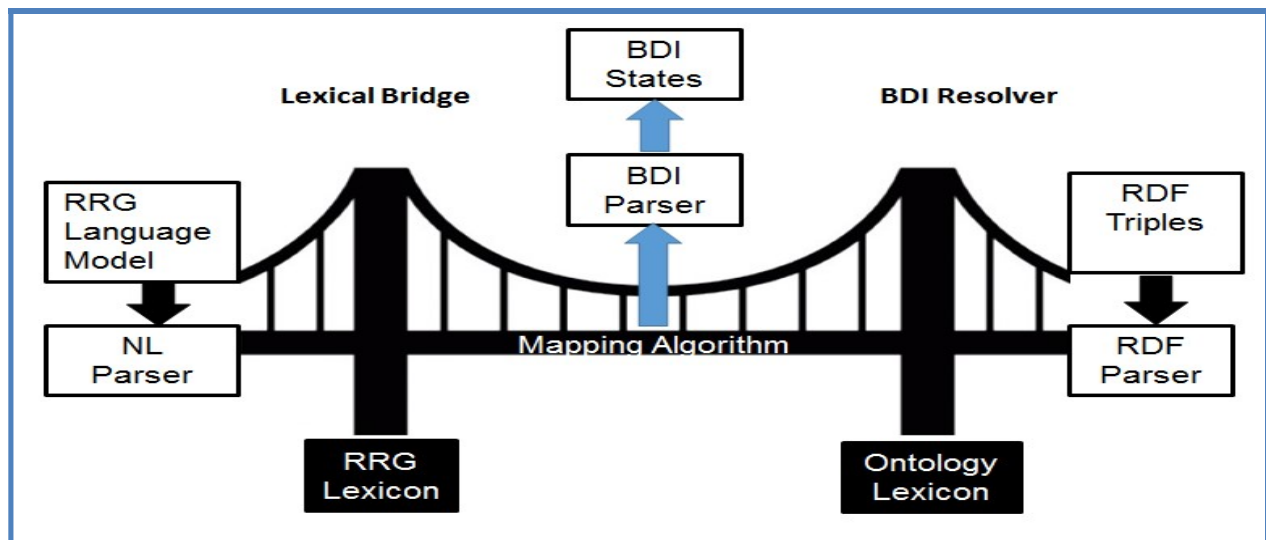
FIGURE 16. LEXICAL BRIDGE FOR THE CSA'S BELIEF BASE + BDI PARSER TO RESOLVE THE AGENT'S BDI STATES.

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 algorithmnic 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 6, and an example RDF triple in Table 7.

Lexical Bridge (LB) Algorithm
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. PSEUDO CODING STEPS ARE IDENTIFIED FOR THE LEXICAL BRIDGE.

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

TABLE 7. RDF OBJECT - SEMANTIC EQUIVALENT NL WORDS IN THE LEXICAL BRIDGE – STEP 1.

The conceptualisarion of the lexical bridge will require continued parsing of all the RDF object property literals, to complete the lexicalisation process, and invoking the RRG Lexicon as identified in Table 8. Step 3 from Table 6 - requires an algorithm to compare the lexical semantics in the RDF form to the NL 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 8. 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).

RDF concept	Word #	Word	POS tag	POS	Base verb
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

TABLE 8. PARSE WITH CORRESPONDING NATURAL LANGUAGE TEXTS – STEP 2.

This LB could potentially provide the following significant benefits: (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.

5.4. 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 (DM) is invoked, which further invoke the RRG Language model. The DM has a dual role in the analysis of the utterance and in the generation of the response. In this section

we address criteria (7-9):

- 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 17 illustrates how the RRG Language Model and Knowledge Model interface with the Agent Cognitive Model (ACM). Further it demonstrates the dialogue handling process to support the generation of a response. Conceptually the DM 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 be refined to a grammatically correct response. Figure 17 presents a class diagram of Java based classes in Eclipse IDE platform.
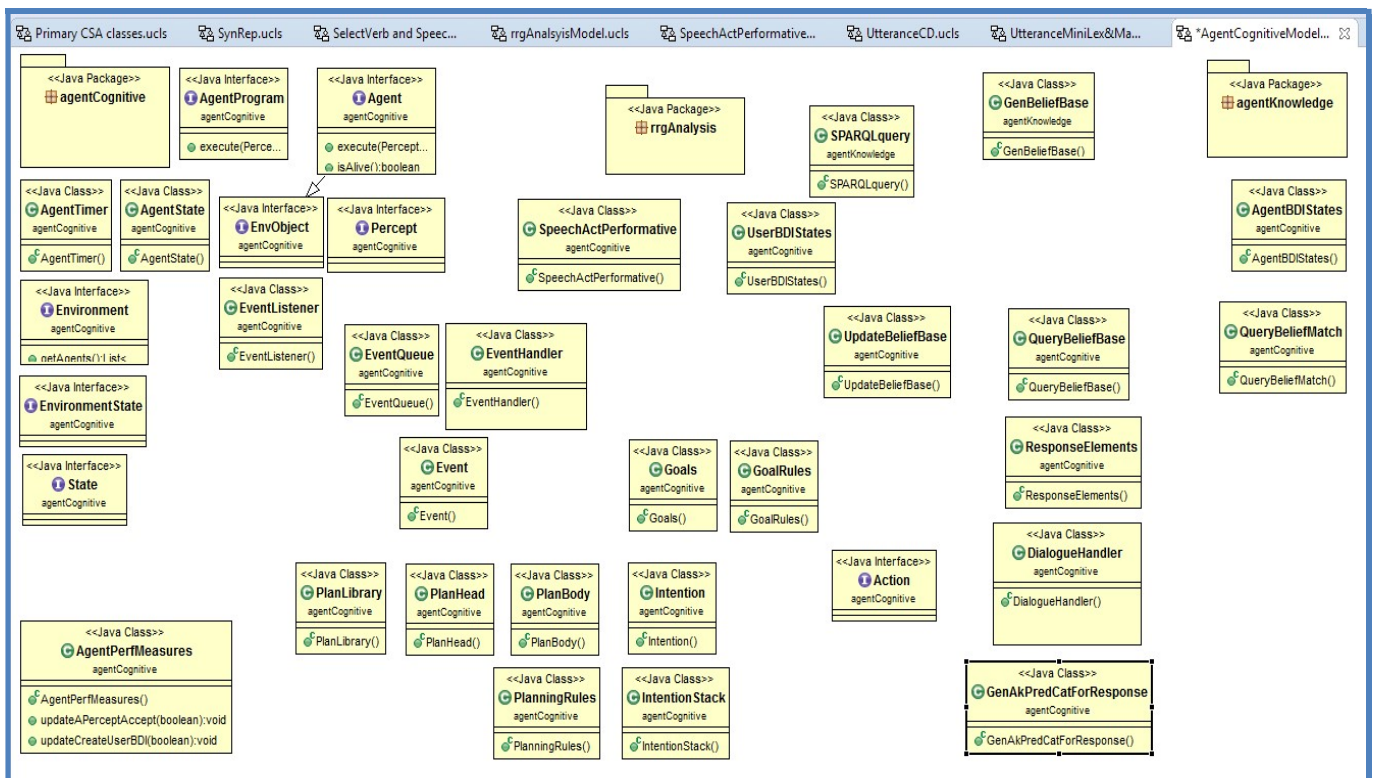

FIGURE 17 - CLASS DIAGRAM - AGENT COGNITIVE MODEL - CLASSES IN ECLIPSE.

Figure 17 evidences groundwork for the : (a) workings of the agent environment, (b) perceiving a 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) 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.

Criteria (7) – 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

ACM. 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 thus present a semantic gap. This technical issue posed a massive challenge similarly identified by (Bimson and Hull 2016). This has been explored and an abstract conceptual framework of a lexical bridge (LB) utilizing the RRG Lexicon with a mapping algorithm has been proposed to solve this problem, and further facilitates the querying of the KB.

Criteria (8) relating to the representation the agent's BDI states was also an issue, and was explored by extending the LB with a BDI parser.

Criteria (9) will be achieved by adding a next layer of classes to reflect the technical solution of the LB, mapping algorithm and the BDI parser. The DM 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 LB solution.

6. CONCLUSIONS AND FUTURE WORK

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. 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, agent environment and the functions of a CSA A set of evaluation criteria was devised to exhaustively assess where the requirements of LING- 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 of these is presented.

Example 1 - the workings of two logical structures (LSs) and three macroroles, - an obstacle. This issue lead to problems with irregular verbs and exceptional transitivity (number of macroroles is greater than two).

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.

6.2. 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 SAs to formulate the dialogue response, and thus will require a more refined evaluation methodology and framework.

REFERENCES

Abran, A., A. Khelifi, W. Suryn, and A. Seffah. 2003. "Consolidating the ISO usability models." In Proceedings of 11th International Software Quality Management Conference 2003: 23-25.

Bimson, Kent D., and Richard D. Hull. 2016. "Unnatural Language Processing: Characterizing the Challenges in Translating Natural Language Semantics into Ontology Semantics." In

Semantic Web: Implications for Technologies and Business Practices, edited by Michael Workman, 119-135. Cham: Springer International Publishing.

Bimson, Kent D., Richard D. Hull, and Daniel Nieten. 2016. "The Lexical Bridge: A Methodology for Bridging the Semantic Gaps between a Natural Language and an Ontology." In *Semantic Web: Implications for Technologies and Business Practices*, edited by Michael Workman, 137-151. Cham: Springer International Publishing.

Bratman, Michael. 1987. *Intention, plans, and practical reason*. Center for the Study of Language and Information: Stanford University.

Cohen, Philip R, and Hector J Levesque. 1988. Rational interaction as the basis for communication. DTIC Document.

Cohen, Philip R, and Hector J Levesque. 1990. "Intention is choice with commitment." *Artificial Intelligence* 42 (2): 213-261.

Dijkstra, Edsger W. 1959. "A note on two problems in connexion with graphs." *Numerische Mathematik* 1 (1): 269-271.

Francois, Jacques. 2014. "Functionalist views on the genesis and development of grammar." In 25th European Systemic Functional Linguistics Conference Paris, July 11th 2014.

GraphIK. 2012. "CoGui (Conceptual Graphs Graphical User Interface) User Guide." accessed 12 March 2014. http://www.lirmm.fr/cogui/newdoc/cogui-index.html.

Horrocks, Ian, Peter F Patel-Schneider, Sean Bechhofer, and Dmitry Tsarkov. 2005. "OWL rules: A proposal and prototype implementation." *Web Semantics: Science, Services and Agents on the World Wide Web* 3 (1): 23-40.

Kamp, Hans, Josef Van Genabith, and Uwe Reyle. 2011. "Discourse representation theory." In *Handbook of Philosophical Logic*, 125-394. Springer.

Lester, James, Karl Branting, and Bradford Mott. 2004. "Conversational agents." *The Practical Handbook of Internet Computing, CRC Press, UK.*

Mao, Xuetao, Jean-Paul Sansonnet, and Lin Li. 2012. "Textual Conversation Agent for Enhancing Attraction in E-Learning." In Proceedings of International Conference on Computer Science and Information Technology.

Nolan, Brian. 2014. "Constructions as grammatical objects : A case study of prepositional ditransitive construction in Modern Irish." In *Linking Constructions into Functional Linguistics: The Role of Constructions in Grammar*, edited by Brian Nolan and Elke Diedrichsen, 143-178. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Panesar, Kulvinder. 2017. "A linguistically centred text-based conversational software agent." Unpublished PhD thesis, School of Computing, Creative Technologies and Engineering, Leeds Beckett University.

Perez-Marin, Diana, and Ismael Pascual-Nieto. 2011. *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*. Pennsylvania: Information Science Reference.

Radziwill, Nicole M, and Morgan C Benton. 2017. "Evaluating Quality of Chatbots and Intelligent Conversational Agents." *arXiv preprint arXiv:1704.04579*.

Russell, S., and P. Norvig. 2016. *Artifical Intelligence: A Modern Approach - Global Edition*. Sixth ed: Pearson.

Searle, John R. 1969. *Speech Acts: An Essay in the Philosophy of Language*. Vol. 626. Cambridge: Cambridge University Press.

Searle, John R. 1980. "Minds, brains, and programs." *Behavioral and Brain Sciences* 3 (3): 417-424.

Searle, John R. 1983. *Intentionality: An Essay in the Philosophy of Mind*. Cambridge: Cambridge University Press.

Stalnaker, Robert. 2002. "Common ground." *Linguistics and Philosophy* 25 (5): 701-721.

Turing, Alan M. 1950. "Computing machinery and intelligence." *Mind, Volume LIX, Issue 236*: 433-460.

Van Valin Jr, Robert D. 2005. *Exploring the Syntax-Semantics Interface*. Cambridge: Cambridge University Press.

Weizenbaum, Joseph. 1966. "ELIZA—a computer program for the study of natural language communication between man and machine." *Communications of the ACM* 9 (1): 36-45.

Wooldridge, Michael. 2013. "Intelligent Agents." In *Multiagent Systems*, edited by Gerhard Weiss, 3-50. USA: Massachusetts Institute of Technology.

Worswick, Steve. 2018. "Mitsuku - Loebner Prize 2018." https://www.pandorabots.com/mitsuku/, accessed 12th September 2018. http://www.mitsuku.com/.

Yao, Haining, and Letha Etzkorn. 2006. "Automated conversion between different knowledge representation formats." *Knowledge-Based Systems* 19 (6): 404-412.