Знания-Онтологии-Теории (ЗОНТ-15)

Agent Language for communication between humans and computer agents

Anton Kolonin[1]

[1]*Aigents Group, Pravdy 6, Novosibirsk, 630090, Russian Federation*

akolonin@gmail.com

Annotation. *In this paper we describe development of the language for cross communications between humans and computer agents. The problem is described from the perspective of the "Internet of Things" and "Semantic Web". The paper describes overall requirements for such a language and considers two different cases of interactions of the kind. After all, the proposed language is described with examples, and practical considerations for its implementation and application are given.*

Keywords: agent, belief, interlingua, natural language, ontology, peer-to-peer, robot, semantic web

1 Introduction

There are multiple studies in the field, targeting development of an "interlingua" usable for communication between software agents as well as between humans and these agents [1,2,3,4,5]. The subject becomes much demanded with the emerging conjunction of the "Internet of Things" (i.e. smart consumer electronic devices) with "Semantic Web" (which in turn refers to "things" instead of "strings" in the World Wide Web space). Within convergence of the two paradigms, the two phenomena overlap. On the one hand, various artifacts of modern technology turn into "smart things", intelligent enough to communicate with each other and their masters and operators. On the other, the medium of such communication turns into universe structured with semantic entities or "things" and their relationships. In the world shaped in such a way, the "smart things" should be able to exchange information between themselves and surrounding humans expressing and comprehending structured information about states of any "things" in the world, including them. Hence, here is the obvious need for an interlingua capable to convey compact structured messages easily comprehensible by any agent of such a cross-intelligence conversation between "humanoids" and "robots".

2 Requirements

Currently, the long-term goal of many companies and projects is enabling computer systems to speak real human languages. However, we assume it will not only need substantial time for further research and development, but it will also require enormous computational resources which are typically not available in case of consumer devices. Respectively, we endeavor to come up with a human-computer "interlingua" language trying to follow a more practical short-term approach. Below is a brief list of requirements for such language.

- The language has to possess **communication-wise symmetry** so that any peer agent of the conversation can transmit any kind of statements (declarative, imperative, interrogative, etc.) referring to the context of earlier statements made by any participant. *It would be very different from the* existing asymmetric client-server languages such as SQL, SPARQL, XML [5] or JSON [3] (unless bilateral implementations of such technologies are used).
- The language has to be **semantically expressive** being able to convey fundamental concepts of structured information such as class, object, attribute, value, name, set or array. It would complicate attempts to use synthetic human languages, for instance, simplified English or Lojban [1].
- The language should be **semantically open and ontologically transparent**, so that description and extension of the schema and object model of the subject area for any domain of communication could be done in the same linguistic structure using the same syntax as declarations, directives and interrogations about the data referring to that schema or model. This requirement cannot be met with

XML [5] and JSON [3] since the schema is not expressed in the same medium and to some extent can be met only with RDF [5].
- The language should be **compact and easily parse-able** so large pieces of information can be transmitted and processed with less computational overhead which is especially important for wireless interactions with low bandwidth and embedded devices with low memory. Here is where XML-based encodings hardly have a promising future.
- The language should have the **structure congruent to one of prevailing human languages** (for instance, English) so the cost of learning it for humans can be kept to a minimum. Ideally, it should be comprehensible by humans with no special training at all, so that any intelligent consumer device could establish conversation with its owner out-of-the box. There are no good examples of such kind now, except Lojban+ [1] which still requires humans to learn a new natural language.
- The language should be capable of expressing any information contained in "Semantic Web" so that any RDF or Turtle [2] syntax can be transparently translated into it without any information loss. In addition, it should be possible to **express higher-order semantics or hyper-graphs** with triples connected not only to the nodes (vertices) of a graph, but to the links (edges or arcs) of the graph as well.

In the following sections we will consider how the requirements described above can be satisfied with the Agent Language that we are suggesting.

3 Real-world cases

For the sake of practical use, let us consider two application examples from the "world of things talking about things". Both cases would consider development of computer agents capable of a simplistic form of artificial general intelligence, not even close to human intelligence in terms of erudition and knowledge about the real world of humans. However, the cases would assume agents capable to learn modest behavioral schemata in the course of experiential learning through interaction with humans and possibly other peer agents. For each of the cases we need to construct an agent's inner ontology including belief of the world environment of an agent in terms of its sensory environment and action capabilities. Then, we would come up with possible communication scenarios expressed in sort of a natural human language so the communications can be later reduced to the formal language that we propose.

Case 1. Intelligent home thermostat

Let us consider an agent embodying thermostat device for home, which is capable to sense temperature, humidity and CO2 level inside and outside the house and perform actions such as opening or closing ventilation lids, setting fan speed and target temperature (or managing heating and air conditioning, in a more complex model which we are not considering here). For an agent, there would also be a sensory capability provided, so it could perceive the environment as well as feel its own actions adjusted by a human operator. Finally, there would be a feedback channel which could supply operator's reward to an agent effecting in agent's "being good" feeling. Then, the following "inner world" of an agent can be drawn.

- Self
  - Sensing
    - T_outside (temperature outside the house)
    - T_inside (temperature inside the house)
    - HUM_outside (humidity outside the house)
    - HUM_inside (humidity inside the house)
    - CO2_outside (CO2 level outside the house)
    - CO2_inside (CO2 level inside the house)
    - T_target (value adjusting operations of heating and air conditioning)
    - Ventilation with State of ventilation lids (such as Open or Closed
    - Fan with Speed (such as Off, Medium or High)
    - Being either "good" or "not good" (depending on the master's feedback)
  - Acting
    - Setting T_Target
    - Setting Ventilation State
    - Setting Fan Speed

The natural verbal interactions that can be imagined between such a simple artificial pet animal and a homeowner, can be expressed in the following natural language statements.

H: What is your HUM_outside?
A: My HUM_outside is 95%.
H: What are your CO2_inside, CO2_outside?
A: My CO2_inside, CO2_outside are 401, 399.
H: What is your Ventilation State?
A: My Ventilation State is Closed.
H: Your Being is not Good.
H: Set your Ventilation State as Opened.
H: Set your Fan Speed as High if Fan Speed is Off or Medium.
H: Your Being is Good.
...
H: What are your CO2_inside, Ventilation State, Being?
A: My CO2_inside, Ventilation State, Being are 400, High, Good.

It can be seen that communication may involve declarative statements (affirmative and negative) stating particular values and relationships between things and states of the things in Agent's belief as well as interrogative statements asking for these and imperative statements directing the changes. Temporal patterns of joint dynamics of values, changes of the values and feedback can be assumed sufficient for an agent to derive simplistic forms of adaptive behavior maximizing the value of Being Good (the details are not further discussed in this paper). It should be noted, that we anticipate the communication could potentially take symmetric form the thermostat agent could ask a human for its experience of CO2 level (if, for instance, an agent can't sense it) or being good (so that an agent could try to please the master proactively) or even ask the master to open or close windows.

Case 2. Intelligent web weather monitor

In the second case we consider another kind of an agent, which can provide supplementary information about outside temperature (T_outside) to the agent of the first case. However, the latter agent would not sense the temperature directly from the physical environment but rather should be able to read the predicted value of it from external web sources, providing the weather conditions on a geographical basis (such as http://www.weather.com for instance).
Such an agent would need to be capable of knowing the list of sites or web pages to watch for the target data, as well as the list of topics to be tracked. The patterns or templates to be used for this can be learned by an agent in the course of experiential learning or pre-set by a human operator. The agent would also need to have a list of its peer contacts that have to be updated with the collected news.

The sensing of such an agent would get represented by an internal time sensor plus variables keeping the context specific to processing a particular site (URL of the site plus text of HTML page downloaded) and the target variable for evaluated temperature. As in the previous case, there could be also a variable for reward/punishment supplement, so that templates as well as behavioral patterns can be learned by an agent by the trial-and-fail method with a feedback from the master.

The acting capabilities of this agent would get more complex than in the previous case: it may include downloading HTML pages from the list of sites, matching templates in the pages and extracting specific values of interest from the findings. In this case, we consider the following “inner world” of the agent.

- Self
  - Sensing
    - Sites (e.g. http://www.weather.com, http://www.accuweather.com)
    - Templates (e.g. “<T_outside> °C Novosibirsk Russia”)
    - Partners (e.g. cell phone numbers, URIs or emails)
    - Time (internal clock value)
    - Site (URL of the current site being read)
    - Text (HTML text being read)
    - T_outside (extracted from the site text with help of the templates)
    - Being (either good or not good, depending on the master's feedback)
  - Acting
    - Reading from <Sites> as <Site> into <Text>

- Matching target <Thing> in <Templates> from <Text>
- Messaging interesting <Thing> to peer <Partner>

Having the agent implementing the above, the following communication scenario can be imagined, if put in terms of a simple natural language referring to the agent's belief system.

H: Your Sites are http://www.weather.com, http://www.accuweather.com .
H: Your T_outside has Templates as "<T_outside> °C Novosibirsk Russia".
H: Your Peers are tel:+79992225555, http://192.168.1.9:88/thermostat, mailto:owner@localhost.
A: Ok.
H: Do Reading from Sites as Site into Text, then Matching target T_outside in Templates from Text, then Messaging found Time, T_outside, Site to peer Partners if T_outside is less than -25 at every 3 hours.
A: Ok.
...
A: My Time, T_outside, Site are 15:00, -23, *http://www.weather.com* .
H: Your Being at 3:00 is not good.
...
A: My Time, T_outside, Site are 18:00, -26, *http://www.weather.com* .
H: Your Being at 6:00 is good.

As in the previous case, that level of human – agent communication will not only enable remote control of an agent monitoring the web, but will also let the agent possibly learn new text patterns and semantic associations and evolve behavioral schemata consisting of elementary actions instead of having it explicitly specified by an operator. Again, the possibility of symmetric communication is assumed so that agent can ask human to check other possibly relevant sites as well as confirm the real outside temperature to verify validity of a given site data or even proactively ask the user for a feedback upon a supplied update.

4 Language description

Starting from the cases and examples above and taking many other cases and examples into account, the following language called "Agent Language" (AL) can be developed. Below is the language grammar specification briefly expressed in EBNF [6] form (without going into details such as defining <number>, <date>, <time> and <string>).

<message> := (<statement> | <acknowledgement>)*
<acknowledgement> := ('ok' | ('true' | 'yes' | <number>) | ('no' | 'false' | 0)) '.'
<statement> := <interrogation> | <confirmation> | <declaration> | <direction>
<interrogation> := 'what' ? <expression> '?' (* "open" incomplete graph *)
<confirmation> := 'if' ? <expression-set> '?' (* "closed" complete graph *)
<declaration> := (<expression-set>) '.' (* "closed" complete graph *)
<direction> := 'do' ? <expression-set> '!' (* "closed" complete graph *)
<expression> := <term> (' ' <term>)* (* separated by spaces *)
<expression-set> := <all-set> | <any-set> | <seq-set> (* different kinds of sets *)
<term> := <negation>? (<anonymous>? | <self> | <peer> | <id> | <name> | <value> | <qualifier>)
<qualifier> := <expression> | <expression-set>
<any-set> := <or-list> | ('{' <or-list> '}')
<all-set> := <and-list> | ('(' <and-list> ')')
<seq-set> := <then-list> | ('[' <then-list> ']')
<or-list> := <expression> ((',' | 'or') <expression>)*
<and-list> := <expression> ((',' | 'and') <expression>)*
<then-list> := <expression> ((',' | 'next') <expression>)*
<negation> := 'not' | 'no' | '~'
<anonymous> := ('there' ('is'|'are')) | 'any' | 'anything' ?
<self> := 'my'|'i'|'we'|'our'
<peer> := 'your'|'you'
<value> := <number> | <date> | <time> | <string>

Here is a brief description of the language, under an assumption that the purpose of the language is to express the full set of operations on a semantic hyper-graph consisting of terminal nodes and typed links between the nodes, links and sets of nodes and/or links.

- **Message (Document)** – a series of semantically consistent statements and optional acknowledgments to the former statements, bounded physically by means of external communication protocol or storage medium.
  - **Acknowledgement** – used to provide response to the statements other than interrogation. For declarations and directions “ok” is returned. In case of “if”-style confirmations “false”, “no” or zero can be returned for failed assertions while “true”, “yes” or number of evidences (count of the applicable subgraphs) satisfying the assertion (operating like COUNT(*) function in SQL) – for successful assertions.
  - **Statement (Sentence)** – an association of semantically connected expressions, separated by delimiters such as commas, exclamation and interrogation symbols ('.', '!', '?') or by external formatting markup (e.g. HTML blocks, list items, table cells, etc.). There can be four kinds of statements, varying in the degree of certainty in respect to the contained expressions. That is, **interrogation** means a speaker has no idea about the complete matter of an expression being asked about, **confirmation** means the matter is guessed and there is a need to have it confirmed, **declaration** expresses the matter with some confidence while **direction** is intended to force the listener to accept the matter communicated by the speaker.
    - **Interrogation** statement is denoted with the question mark at the end and can be used to perform a query against the specified incomplete (or “open”) subgraph to retrieve specific parts of it, effectively representing complementary subgraphs needed to make the expression in the statement complete (similarly to SQL or SPARQL query). It can be preceded with “**what**” (i.e. “which”) keyword as a clue for text parser.
    - **Confirmation** statement is also denoted with the question mark at the end and can be used to check the existence of certain complete (or “closed”) subgraph or a set of subgraphs or truth value of an assertive expression or expressions encompassing them. It can be preceded with “**if**” (i.e. “whether”) keyword as a clue for text processor, so it could be distinguished from interrogation without semantic analysis of the query.
    - **Declaration** statement is denoted with the period mark at the end and is used in conversation purely for the declaration purposes, so the receiver of the message can handle it at its discretion. It contains a single expression or a set of expressions encompassing complete assertions or “closed” subgraphs. This is what is returned upon execution of “what”-style interrogations or can be used to load knowledge into an agent's belief system. It is the default kind of statement to be expected by a language processor, so it is not supplied with a clue keyword.
    - **Direction** statement is denoted with the exclamation mark. Beyond just declaring the essence of the matter like above mentioned declaration does, it is used to force the partner receiving the message to accept the assertive expression or expressions, so they are incorporated in their belief graph. It can be preceded with “**do**” keyword as a clue for the text processor.
  - **Expression (Phrase or Proposition)** – an elementary constituent of a sentence. It is an ordered association of **terms** separated by spaces and bound to the same semantic entity, or a subgraph encompassing the entity in the following way.
    - The subgraph represented by expression can be **“closed” or complete**, so there is no ambiguity in its expression and no hidden variables implied. The subgraph can also be **“open” or incomplete**, encompassing unresolved variables or “hanging links” in the graph, so it can be used to create “what”-style interrogations being asked to resolve these variables.
    - The subgraph can be defined like a locally restricted **path in a hyper-graph**, most likely a set of mutually interconnected RDF or Turtle expressions.
    - Unlike RDF or Turtle expressions, the length of the single expression path is not restricted, so besides subject, predicate and object arguments, an expression can represent subject-predicate-predicate-...-predicate-... **inference chains** of arbitrary length (such as “*my favorite bank customer mother maiden name”*).
    - Unlike Turtle, which can have multiple predicate-object clauses as well as multiple object arguments, an expression can have **multiple subjects and predicate verbs** as well, which may be grammatically supported with using implicit or explicit parentheses ('(' and ')'), braces ('{' and '}') and brackets ('[' and '}') instead of using Turtle's colons and semi-colons (making the “*you and me will work and live together forever*” possible).

    - Unlike Turtle, potentially any **terms can be complete expressions**, rather than static resources, with some restrictions to be implied by particular implementations of the language (so an English phrase "*anyone taller than 1.8 meters is tall*" is a valid AL expression "*(taller 1.8) is tall*").
    - Expressions can be composed together into an **expression set**, qualifying composite semantic entities, or **composite qualifiers** - as is described below.
- **Argument (Term)** – an atomic constituent of an expression, denoting either a node or a link in a graph or a whole set of nodes or links representing a similar semantic entity. Any term can be preceded with unary **negation** operator (expressed as 'no', 'not' or '~') inverting the subgraph scoping from inclusion to exclusion. The following terms can be used.
    - **Anonymous** term denoting an unlabeled entity not unassociated with any prior experiences, to be identified solely by further arguments of an expression.
    - **Self-reference** grammatically denoted as "My" or "I" (or "We" or "Our"), identifying the first-person of an agent.
    - **Peer-reference** identified by "Your" or "You", identifying the partner of the communication.
    - **Reference by Id or URI** which may be used to refer to any subjects and objects in the course of internal non-human communications between non-human agents.
    - **Reference by Name** is intended to identify any named entities including persons, classes, properties of objects, verbs and operator symbols such as '+', '-', '=', '>', '<'.
    - **Value** can be encompassing any semantic terminals such as finite numbers, literal strings, characters, times and dates.
    - **Qualifier** is intended to refer to a complete semantic entity or a set of entities applying a hierarchy of expressions that restrict the graph down to the target set of relationships in the graph. It could be either a singular expression (like "*big tree*") or a **composite qualifier** represented by an **expression set** (like "*big green tree on the other side of the street next to the parking lot*").
- There can be three kinds of **expression sets** or **composite qualifiers** where each of them can associate a list of qualifying expressions recursively – with different logical and sequential operations implied to associate the expressions in the set, as follows.
    - **Disjunctive** qualifier representing "**OR**"-style logical association where **any** expression in the list can be used to qualify the entire term.
    - **Conjunctive** qualifier representing "**AND**"-style logical association where **all** expressions in the list have to be resolved in order to qualify the term.
    - **Successive (Ordered)** qualifier representing "**NEXT**"-style logical and sequential association where all expressions in the list have to be resolved strictly in the specified order.

Trying to use the grammar of AL language to generate some kind of a human-friendly speech might get little weird for a native English speaker. This is because the roles of terms in the expressions in English are driven by the grammatical order, prepositions and forms of the verb "to be". However, it might be easier to accept for Portuguese and Russian speakers where the same grammatical statement can have a different mood (interrogative, declarative, imperative) depending on the tonal modulation, as shown in the following table.

English	**AL (no clues)**	**Russian (with tonal modulation)**
What is your feeling?	*Your feeling?*	*Твое ощущение? (tone up)*
If your feeling is good?	*Your feeling good?*	*Твое ощущение хорошее? (tone up)*
Your feeling is good.	*Your feeling good.*	*Твое ощущение хорошее. (tone neutral)*
Have your feeling good!	*Your feeling good!*	*Твое ощущение хорошее! (tone down)*

Use of composite qualifiers can be presented with the following mapping where expressions on the left are proper AL expressions built with use of braces, brackets and parentheses (which might be easier for computer text parser) while the expressions on the right are glued with conventional prepositions (which might be more convenient for human ear). However, it should be noted that the right-side expressions present the ambiguity of the logical term grouping that cannot be resolved by a simple parser, incapable to detect semantic roles of predicate and object terms.

I (can (eat, sleep), want (dance, sing)). *<=> I can eat and sleep and want dance and sing.*
I {can (eat, sleep), want (dance, sing)}. *<=> I can eat and sleep or want dance and sing.*
I (can {eat, sleep}, want {dance, sing}). *<=> I can eat or sleep and want dance or sing.*
You [eat {rice, meat}, drink {juice, water}]! *<=> You eat rice and meat next drink juice and water!*

From the perspective of using the language to represent semantic expressions, the following example demonstrates how the AL can be used to express statements in conventional term logic or Turtle syntax. In the last two examples, no conversion to Turtle is applicable at all.

AL	**Term logic**		**Turtle**
A C (D,E).	*<=> A C D. A C E.*	*<=>*	*A C D,E.*
A (C D, F G).	*<=> A C D. A F G.*	*<=>*	*A C D; F G.*
A (C (D,E), F (G,H)).	*<=> A C D. A C E. A F G. A F H.*	*<=>*	*A C D,E; F G,H.*
(A,B) C D.	*<=> A C D. B C D.*		
(A,B) (C (D,E), F (G,H)).	*<=> A C D. A C E. B C D. B C E. A F G. A F H. B F G. B F H.*		

5 Testing cases

To justify the constructed language grammar, let us try to encompass the communications for the two real-world cases described two sections above. In the examples below, we translate the required dialogues into suggested AL language. The capitalization is made solely to refer the reader of the paper to the "Sensing" and "Acting" mentioned for both sample agents earlier. That is, we anticipate implementation of the language to be case-insensitive, enabling to simplify speech-to-text and text-to-speech interfaces.

Case 1. Intelligent home thermostat

H: What your HUM_outside?
A: My HUM_outside 95%.
H: What your CO2_inside, CO2_outside?
A: My CO2_inside 401, CO2_outside, 399.
H: What your Ventilation State?
A: My Ventilation State Closed.
H: Your Being not Good!
H: Your Ventilation State Opened!
H: If your Fan Speed {Off, Medium} then Your Fan Speed High!
H: Your Being Good!
A: Ok.
…
H: What your CO2_inside, Ventilation State, Being?
A: My CO2_inside 400, Ventilation State High, Being Good.
H: Ok.

Case 2. Intelligent web weather monitor

H: Your Sites http://www.weather.com, http://www.accuweather.com .
H: Your T_outside has Templates as "<T_outside> °C Novosibirsk Russia".
H: Your Peers tel:+79992225555, http://192.168.1.9:88/thermostat, mailto:owner@localhost.
H: Do Time {3,6,9,12} then Reading from Sites as Site into Text, Matching target T_outside in Templates from Text, T_outside < -25 then Messaging text Time, T_outside, Site peer Partners!
A: Ok.
…
A: My Time, T_outside, Site are 15:00, -23, http://www.weather.com .
H: Your Being (Time 15:00) not Good.
…
A: My Time, T_outside, Site are 18:00, -26, http://www.weather.com .
H: Your Being (Time 21:00) Good.

Being fully comprehensible, the structure of conversations maintains the speaking style of people with little knowledge of an alien language but with expressed cognitive and functional abilities in the restricted domain being the subject of a conversation. A typical example is a conversation of foreign workers or tourists (on the matter of job responsibilities or sightseeing interests) with the locals (from the perspective of those locals). There are some practical considerations worth mentioning, given many such dialogs constructed for different application examples.

6 Practical considerations

To make disambiguation for such kind of conversation working well, without the need to use clue words and explicit symbolic braces and parentheses (so that verbal perception of a language can take a place), congruent ontological and terminological models for both sides of the conversation are necessary. For example, the same classes of real-world objects in beliefs of both participants of the conversation should have the same properties represented by the same literal and verbal terms (i.e. both agents should speak the same language having the same basic ideas regarding the subject of communication). In other words, the communication could be successful only in the strict scope of specific context, shared by all participants of the conversation.

Further, the language intrinsically defines nothing but the ways to express certainty and modality, the structure of statements and the ways to refer to a subject in the first person, in the second person or any subject in the third person referred by name, identifier or qualifier. In order to augment the communication with the features specific to possession, inheritance, time and place, it is necessary to have the minimum set of predicate verbs supported by an ontology employed by an agent speaking the language.

As an example, declarative and directive expressions can be turned into conditional trees in an action graph (representing decision trees or applicable rule sets or executable programs depending on the case) with use of qualifiers and expressions involving predicates such as "**then**" and "**else**" recursively enclosed, as in the following example (note, preceding clue keyword "if" would turn the declaration of the algorithm into confirmation of algorithm existence).

Your CO2_inside > CO2_outside ***then***
T_inside > 19 ***then*** *Your Ventilation State Opened, Fan Speed High*
else *Messaging message text "Alert!" to owner@localhost.home*
else
Your Fan Speed Off,
Your Ventilation State Closed.

In another example, the notion of "**time**" and "**location**" can be expressed in terms of other predicates existing in the ontology of an agent specialized to handle them, as in the following example.

Your time 14:00 being not good, CO2_inside 410, Ventilation State Closed.
My location city Moscow, latitude 55N, longitude 37E weather T_outside -7, HUM_outside 95%.

Finally, to let multiple agents operate and exchange information not only in terms of innately "hardcoded" ontologies, some notions of foundation ontological layer might be supported by a wide range of agents with different practical specializations, including predicate verbs such as "**is**" (being an instance of something), "**has**" (possessing certain properties), "**feels**" (feeling certain senses), "**does**" (be capable of doing specific actions) or "**likes**" (treating someone or something positively). Then, the agents would be able to share their beliefs and world views in conversations like the following.

My is (appliance, computer, agent, thermostat, device).
Device has (shape, color, voltage). Appliance has location.
My (shape rectangular, color blue, voltage 220, location kitchen).
My feels (temperature, humidity, CO2, being). (Temperature, humidity, CO2) is number.
Being is {good, bad}.
My (being good, temperature 20, CO2 312).
My does (messaging, setting).
Messaging has (text, to). To is email. Text is literal.
Setting has target. Target is {ventilation state, fan speed}.

All that said, the AL language does not depend on a human language, as long as a number of clue keywords ("if", "what", "do") can theoretically be avoided and the remaining reference terms such as "i" and "you" can be easily replaced with symbols like "&" and "*". That provides for a few options discussed below.

- First of all, having an ontology implemented for computer agents operating in any practical domain and supplying the ontology with human-friendly labels in some human language (as in the examples above), plain translation of the labels into another language immediately makes agent speaking one more human language about the same domain. Moreover, agents speaking to humans in their own languages would easily understand other agents speaking alien languages as long as label translation mapping table is present.
- Next, many sub-languages can be developed for different practical domains involving intelligent computer agents, so the same communication engine can be re-purposed being overloaded with domain-specific ontologies and vocabularies.
- Lastly, even agents operating in different domains can co-operate if their knowledge relies on the same foundation ontology (for instance, employing basic predicates like then/else, being, possessing, feeling and doing) so their individual intelligence acquired through interactions with humans can be enriched in the course of cross-learning from peer agents.

7 Conclusion

Originally, this work started as an attempt to create a simple and compact API and protocol for sample implementation of one particular kind of intelligent agents. At the very beginning it seemed that simple re-purpose of the ORL [4] would be the best choice. Later, with more practical scenarios considered, we came up with the idea of extending Turtle [2] with the notion of composite qualifiers. However, at the very end we made a hard decision to drop Turtle way of grouping predicates and subjects (using colons and semi-colons) for the sake of being able to group subjects and verb parts of predicate clause alone, and build long predicate chains at the same time.

The language seems compact enough for transmission and visual comprehension, easy to read and write for an average human (not possessing special computer knowledge) and easy to parse into semantic graph operations for computer programs.

In the written form, the ambiguity can be easily resolved with use of clue keywords and braces and parentheses. In the spoken form, however, should one be implemented, or while being typed in by a human, there may be a need for ontology-based disambiguation techniques so only expressions valid in terms of the current ontology are accepted by the parsing process using the underlying ontology while building the parse tree. The is being verified now in our current work on Aigents project (http://aigents.com/) creating personal software agents for intelligent autonomous search of semantic information on the Internet.

References

[1] B. Goertzel: Lojban++: An Interlingua for Communication between Humans and AGIs. Proceedings of 6th International Conference, AGI 2013, Beijing, China, August 2013, Available online: http://goertzel.org/agi-13/Lojban_paper_v1.pdf

[2] World Wide Web Consortium: Turtle – Terse RDF Triple Language". November 2012. Available online: http://www.w3.org/TR/2012/WD-turtle-20120710/

[3] JSON-RPC Working Group, JSON-RPC 2.0 Specification, March 2010, Available online: http://www.jsonrpc.org/specification

[4] A. Kolonin: Object-Relational Language and modern tradeoffs in software technology. International Andrei Ershov Memorial Conference, PSI 2006, Program understanding workshop, Russia, Novosibirsk, June 2006, Available online: http://www.webstructor.net/papers/2006ObjectRelationalLanguageAndModernTradeoffsInSoftwareTechnology.pdf

[5] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks: The Semantic Web: The roles of XML and RDF. IEEE Internet Computing. 15 (3), pp. 63-74. October 2000, Available online: http://www.cs.vu.nl/~frankh/postscript/IEEE-IC00.pdf

[6] R. Scowen: Extended BNF — A generic base standard. Software Engineering Standards Symposium 1993. Available online: http://www.cl.cam.ac.uk/~mgk25/iso-14977-paper.pdf